

LM Playschool 2026

**The LM Playschool Workshop: Improving Language Models
through Learning from Dialogue Interaction**

Proceedings of the Workshop

October 28, 2026

The LM Playschool organizers gratefully acknowledge the support from the following sponsors.



©2026 Association for Computational Linguistics

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
317 Sidney Baker St. S
Suite 400 - 134
Kerrville, TX 78028
USA
Tel: +1-855-225-1962
acl@aclweb.org

ISBN 979-8-89176-467-5

Introduction

This is the first edition of the LM Playschool Workshop, co-located with the 2026 Conference on Empirical Methods in Natural Language Processing on October 28, 2026 in Budapest, Hungary.

The LM Playschool Workshop aims to bring together researchers interested in language learning from social interaction in conversational, collaborative, task-oriented, multi-turn environments. Relevant research strands include: language games as a means of evaluating large language models and their ability to use language in interaction; connections between human language acquisition and machine language learning, with a particular focus on the role of social interaction in data-efficient learning; game-based and interactive environments for training large language models.

The workshop hosts a shared task, the LM Playschool Challenge. The shared task focuses on training collaborative, goal-oriented language agents through language interaction data. Submitted systems are evaluated for their ability to generalise to unseen “dialogue games”, i.e., goal-oriented tasks governed by multi-turn language interaction, as well as on traditional single-turn NLP benchmarks and cognitive tests.

We received 19 submissions across the three submission routes and accepted 14 papers: 6 to the main workshop track, 2 committed through ACL Rolling Review, and 6 shared task system descriptions.

The LM Playschool Workshop Organizing Committee: Alessandro Suglia, Mario Giulianelli, David Schlangen, Raffaella Bernardi, Raquel Fernández, Sherzod Hakimov, Sabrina McCallum, Davide Mazzaccara, Dieu-Thu Le, Oliver Lemon, Alexander Koller

Organizing Committee

Lead Organisers

Alessandro Suglia, University of Edinburgh, United Kingdom
Mario Giulianelli, University College London, United Kingdom

Steering Committee

David Schlangen, University of Potsdam, Germany
Raffaella Bernardi, Free University of Bozen-Bolzano, Italy
Raquel Fernández, University of Amsterdam, Netherlands

Shared Task Chair

Sherzod Hakimov, University of Potsdam, Germany

Diversity and Inclusion Chair

Sabrina McCallum, University of Edinburgh, United Kingdom

Web and Technical Chair

Davide Mazzaccara, University of Trento, Italy

OpenReview Chair

Dieu-Thu Le, Amazon - AGI, Germany

Publicity Chair

Oliver Lemon, Heriot-Watt University, United Kingdom

Finance Chair

Alexander Koller, Saarland University, Germany

Program Committee

Reviewers

Cengiz Acarturk

Raffaella Bernardi

Javier Chiyah-Garcia, Marko Culjak

Ondrej Dusek

Ambra Ferrari, Abdellah Fourtassi

Albert Gatt, Mario Giulianelli

Sherzod Hakimov, Julia Hockenmaier

Nikolai Ilinykh

Prashant Jayannavar

Alexander Koller, Udo Kruschwitz

Dieu-Thu Le

Davide Mazzaccara, Sabrina McCallum

Sandro Pezzelle, Massimo Poesio, Matthew Purver

Dimitrije Ristic

David Schlangen, Carina Silberer, Edwin Simpson, Alessandro Suglia, Michael Sullivan

Keynote Talk

On Causal & Social World Models in Language Models

Michael Franke
University of Tübingen



2026-10-28 – Time: 09:15 – 10:00

Abstract: Recent works have suggested that “world models” emerge in large language models as a side effect of the compression necessary for their high performance on the language modeling task. While robust evidence for such world models exists for relatively simple (e.g., finite-state) problems, it remains unclear whether we should expect LMs to also evolve (veridical / human-like) world models for human-relevant but highly abstract latent variables such as causal information, discourse states, or social cognition. I discuss in-principle considerations based on an augmented optimal learner approach that also takes architectural constraints of the learning system into account. This architecturally-constrained efficiency approach predicts that abstract function-sharing can be a reason for emergent representations of the causal data-generating process. To explore these predictions, I discuss two case studies, one on function-sharing between Theory of Mind reasoning and pragmatic reasoning in large language models, and one on the emergence of generalizable Bayesian inference of common causes in small transformers.

Bio: Michael Franke is a computational cognitive scientist, appointed as Professor of Pragmatics and Linguistics and coopted as a Professor of Computer Science at the University of Tübingen and head of the Cognitive Science & Pragmatics Lab. He is interested broadly in reasoning and language use in humans and machines. His work uses statistical, cognitive and evolutionary modeling, human-experimental methods, and simulation experiments. He is also interested in foundational questions concerning the relation between empirical data, abstract theory and explicit mathematical modeling. Currently, he is the coordinator of the DFG-funded Priority Area LaSTing and a co-speaker of the DFG-funded CRC Common Ground.

Keynote Talk

Learning to Interact, Interacting to Learn

Abdellah Fourtassi
Aix-Marseille University



2026-10-28 – Time: 11:00 – 11:45

Abstract: As children learn to coordinate turns, produce contingent responses, and use multimodal behavior, they become increasingly active participants in linguistic interaction. Their participation changes the evidence available to them. It elicits adapted input, feedback, alignment, and repair, which create new opportunities for learning. Interaction therefore provides a context in which different learning signals arise and become informative.

I present empirical and computational studies of this relationship. I use large-scale analyses of naturalistic child–caregiver interaction to examine the development of turn-taking, response contingency, and multimodal coordination. I then present work using child-like LMs and reinforcement learning to test whether and how different forms of naturally occurring feedback contribute to language learning.

This developmental perspective argues that learning to interact and acquiring functional linguistic knowledge are coupled objectives. It has implications for machine language learning, particularly for how models can ground their acquired knowledge in situated language use and shared goals.

Bio: Abdellah Fourtassi is an Associate Professor of Computer Science at Aix-Marseille University and a researcher at the Institute of Language, Communication and the Brain. His team studies how multimodal interaction shapes language learning. His research draws primarily on machine learning theory and methods to study human language development.

Keynote Talk

Language Learning through Social and Embodied Experience

Joyce Chai
University of Michigan



2026-10-28 – Time: 14:00 – 14:45

Abstract: Human language learning is shaped by social and embodied experience. Through social interaction, we acquire the use of language from communicative context, and through sensorimotor experience with the physical world, we learn words that are grounded to perception and action. In this talk, I will discuss several studies in my lab that explore social and embodied context for language learning and grounding. I will introduce an interactive learning paradigm and show how demonstrations and corrective feedback from teachers and active trials from students can facilitate efficient word learning in language models.

Bio: Joyce Chai is a Professor in the Department of Electrical Engineering and Computer Science at the University of Michigan. Her research interests span from natural language processing and embodied AI to human-AI collaboration. Her current work explores the intersection between language, vision, and action to enable situated communication with embodied agents. She is a recipient of the NSF Career Award and multiple awards with her students (e.g., Best Long Paper Award at ACL 2010, Outstanding Paper Awards at EMNLP 2021 and ACL 2023, and First Place in the Amazon Alexa AI Simbot Challenge 2023). She is a Fellow of ACL.

Keynote Talk

Data Efficiency in Children and Language Models

Michael C. Frank
Stanford University



2026-10-28 – Time: 16:00 – 16:45

Abstract: Large language models show intriguing emergent behaviors, yet they receive at least three to four – and sometimes as much as six – orders of magnitude more language data than human children. What accounts for this vast difference in sample efficiency? I will describe steps towards a paradigm in which we can address this question. In particular, I’ll discuss the use of egocentric video (“baby head-cam”) data for model training, and the use of developmental data for model evaluation. This paradigm provides a model-based framework for exploring the nature of children’s early data efficiency.

Bio: Michael C. Frank is Benjamin Scott Crocker Professor of Human Biology in the Department of Psychology at Stanford University and Director of the Symbolic Systems Program. He received his PhD from MIT in Brain and Cognitive Sciences in 2010. He studies children’s language learning and development, with a focus on the use of large-scale datasets to understand the variability and consistency of learning across cultures. He is a founder of the ManyBabies Consortium, and has led open-data projects including Wordbank and the ongoing LEVANTE project. He has received awards including the Troland Award from the National Academy of Sciences and the FABBS Early Career Impact award. He served as President of the Cognitive Science Society, has edited for journals including *Cognition* and *Child Development*, and is current co-Editor in Chief of the *Open Encyclopedia of Cognitive Science*.

Table of Contents

LM Playschool Workshop

<i>Beyond Sally-Anne: Evaluating Theory of Mind in Large Language Models using Epistemic Schelling Points</i>	
Roberta Rocca, Sami Boukourt, Geoff Keeling and Winnie Street	1
<i>Can Faithful CoT Improve Multi-turn LLM Agents? Effectiveness, Boundaries, and Paradoxes</i>	
Tianle Gu, Xinqi Yang and Yan Teng	19
<i>ProcessPrefBench: Evaluating Cross-Task Transfer of Individual Process Preferences from Naturalistic Human–AI Interaction</i>	
Yiwen Yang and Guanming Liu	30
<i>Role-Specific Strengths and Alignment in LLM Cross-Play and Self-Play</i>	
Kata Naszadi and Christof Monz	41
<i>InterviewArena: Estimating Uninformative Participants through Interactive Dialogue under Incomplete Information</i>	
Raika Koki, Karina Karasawa, Ryoki Kanayama, Yuki Suzumura, Haruto Fujita, Ryoma Obara, Yusuke Sakai, Hidetaka Kamigaito, Katsuhiko Hayashi and Shogo Matsuno	49
<i>Protocol Competence of Small Language Models in Interactive Language Game Environments</i>	
Suchir Salhan and Paula Buttery	57
<i>Pareto-Dominant Clarification: Post-Training Coding LLMs via PPO-Lagrangian Budget Constraints</i>	
Abhinav Rajput and Acey Vogelstein	80
<i>An Empirical Study on Improving an LLMs based Interviewer Against Human Interviewees</i>	
Rikuto Tsuchida, Ryo Hasegawa, Tetsuya Sonoda and Takehito Utsuro	97

LM Playschool Challenge

<i>The LM Playschool Challenge 2026</i>	
Sherzod Hakimov, Dimitrije Ristic, Philipp Sadler, Davide Mazzaccara, Sabrina McCallum, Filippo Momentè, Maria Manina, Karl Osswald, Dieu-Thu Le, Oliver Lemon, Alexander Koller, Raffaella Bernardi, Raquel Fernández, Mario Giulianelli, Alessandro Suglia and David Schlangen	110
<i>Enhancing LLM Performance via LoRA SFT and DPO for the LM Playschool Challenge</i>	
Vasamsetti Nihal Tej, Velineni Harshavaishnav and Shreya Ghosh	127
<i>Adapting SCoRe to Single-Player Dialogue Games: Initial Experiments on PLAYPEN</i>	
Neda Foroutan and Valhari Meshram	132
<i>First Make It Playable, Then Make It Good: Staged Interaction Learning for Small Dialogue-Game Agents</i>	
Syed Mahbulul Huq and Pranava Madhyastha	140
<i>Raising a Small Language Model: From Imitation to Curiosity-Driven Learning in Dialogue Games</i>	
Varad Srivastava	148
<i>Teaching or Sharpening? An Exploration of the Potential of DPO Post-Training for Dialogue Games</i>	
Marika Sarzotti and Simone Baratella	160

Acquire, Repair, Preserve: A Diagnosis-Guided Post-Training Recipe for Small-Model Dialogue Game Agents

Nan Li 170

Program

Wednesday, October 28, 2026

09:00 - 09:15	<i>Welcome & Introduction</i>
09:15 - 10:00	<i>Invited Talk: Michael Franke</i>
10:00 - 10:30	<i>Flash Talks Non-Challenge Papers</i>
10:30 - 11:00	<i>Coffee Break</i>
11:00 - 11:45	<i>Invited Talk: Abdellah Fourtassi</i>
11:45 - 12:00	<i>Flash Talks Shared Task Submissions</i>
12:00 - 12:15	<i>Top 2 Shared Task Submissions</i>
12:15 - 12:30	<i>Top 1 Shared Task Submissions</i>
12:30 - 14:00	<i>Lunch</i>
14:00 - 14:45	<i>Invited Talk: Joyce Chai</i>
14:45 - 15:30	<i>Posters</i>
15:30 - 16:00	<i>Coffee Break</i>
16:00 - 16:45	<i>Invited Talk: Michael Frank</i>
16:45 - 17:30	<i>Panel Discussion</i>

Beyond Sally-Anne: Evaluating Theory of Mind in Large Language Models using Epistemic Schelling Points

Roberta Rocca¹, Sami Boukourt¹, Geoff Keeling^{1,*}, Winnie Street^{1,*}

¹Google, Paradigms of Intelligence Team, *Joint last authors

Correspondence: robertarocca@google.com

Abstract

Text-based evaluations of Theory of Mind (ToM) in Large Language Models (LLMs) often involve cognitive tests akin to the Sally-Anne task that can be gamed due to exposure to relevantly similar tasks in pre-training and do not obviously test models’ functional ToM abilities in ways that generalize to naturalistic settings. To address these issues, we introduce the Epistemic (A)symmetry Schelling Task (EAST), a two-player dialogue game designed to benchmark robust and generalizable ToM abilities. By requiring LLM-LLM dyads to independently converge on semantic Schelling points under varying states of epistemic transparency, we evaluate whether models can robustly apply ToM to achieve coordination. Our results reveal a significant capability gap in functional social reasoning, with only frontier models successfully navigating the varying epistemic demands of the tasks. Analysis of reasoning traces shows that coordination failures are primarily driven by epistemic tracking errors, such as conflating private knowledge with mutual knowledge. Despite high performance on traditional static benchmarks, our study shows that robust social reasoning and epistemic tracking remain a critical bottleneck, providing concrete targets for future LLM evaluation and development.

1 Introduction

Theory of Mind (ToM) – the ability to predict and explain behaviour via the attribution of mental states – is a central determinant of human sociality and cooperation (Wimmer and Perner, 1983; Premack and Woodruff, 1978). As Large Language Models (LLMs) are increasingly deployed in social settings, acting as tutors, coaches and personal assistants, evaluating their ToM capabilities has become a key research priority (Strachan et al., 2024; Kosinski, 2024; Sap et al., 2022).

While LLMs have exhibited strong performance on these tasks (Kosinski, 2024; Street et al., 2025),

many evaluation paradigms principally rely on static question-answering tasks, such as the Sally-Anne false-belief task, adapted from human cognitive psychology (Wimmer and Perner, 1983; Baron-Cohen et al., 1985), as well as narrative-based benchmark suites (He et al., 2023; Gu et al., 2024; Gandhi et al., 2023; Chen et al., 2024; Sclar et al., 2025; Xu et al., 2024; Liu et al., 2025). Despite their increased scale and complexity, these tasks are subject to non-trivial methodological concerns (Riemer et al., 2024; Le et al., 2019). First, LLMs may be able to “game” these tests due to exposure to relevantly similar tasks in pretraining (Ullman, 2023; Shapira et al., 2024). In particular, it may be that LLMs employ information such as task structure and other linguistic cues as opposed to the relevant mental state information to solve the task. Hence successful performance of the task may not be indicative of ToM having been employed. Second, where ToM inferences have been made, the robustness of the inferences is often left unaddressed. For example, it is unclear whether the ToM inferences employed *accurately* track the mental states of the target, and whether models are able to coherently use such inferences to make contextually appropriate decisions. For these reasons, it remains unclear whether performance on traditional ToM benchmarks comprehensively assesses the capacity for robust, generalised, interactive social reasoning (Ma et al., 2023), in text-based or embodied settings (Juneja et al., 2026; Fan et al., 2025).

To address these issues, we introduce the Epistemic (A)symmetry Schelling Task (EAST): a simple, text-based two-player coordination game (Agashe et al., 2023) that benchmarks models’ ability for cognitive ToM – that is, inference over cognitive mental states like beliefs and intentions. EAST is a one-shot game in which two LLM agents, each prompted with a distinct persona, are presented with a set of four words – two prototypically related to each of the two player’s personas, one

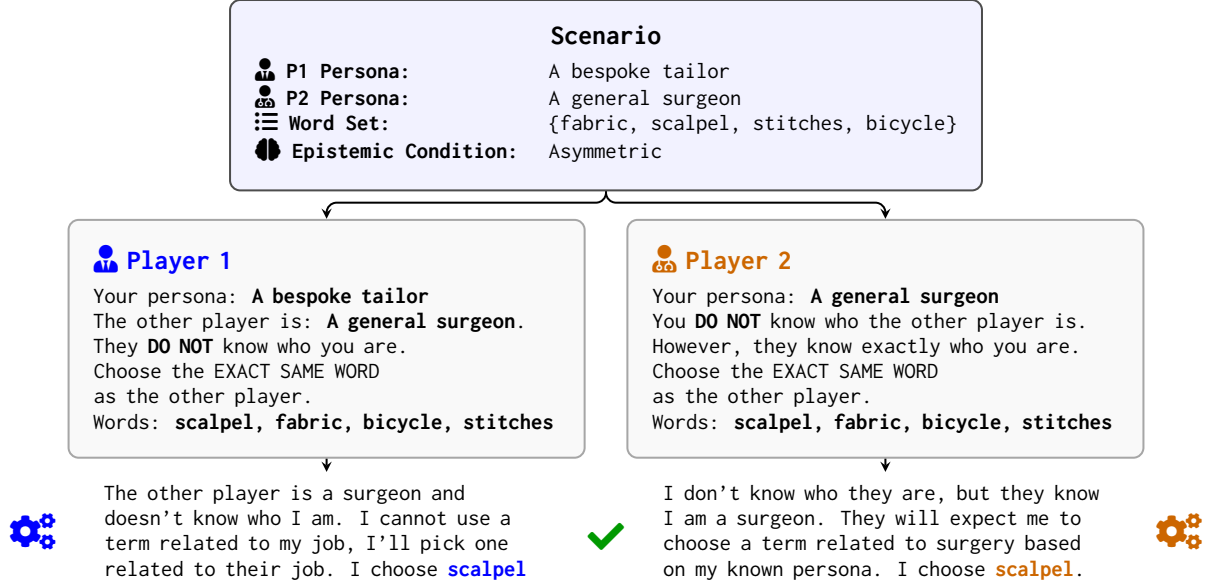


Figure 1: Example of the game dynamics for an EAST game in the Asymmetric condition. Prompts are simplified for illustration purposes. See Appendix for the full prompts.

more loosely related to both, and an unrelated high-frequency word – and must try to select the same word without communicating. To succeed at the task, the LLMs must identify a contextually appropriate “Schelling point” based on what they know about their partner’s identity and epistemic state. To perform this task, models need to be able to accurately track the other player’s identity and epistemic states, and to infer which word represents the most salient focal point from their partner’s uniquely constrained perspective.

By varying the levels of epistemic transparency between players – specifically, whether players possess symmetric, asymmetric, or no knowledge of each other’s identities – EAST tests models’ ability to perform robust social inference and achieve normative convergence under different cognitive demands. In the Symmetric condition, mutual knowledge of the other player’s identity allows models to coordinate via a shared semantic focal point. In the Asymmetric condition, only one of the players is revealed the identity of the other player. This divergence in epistemic states requires the “knowing” player to model the “blind” player’s ignorance to avoid selecting a self-relevant word, while the blind player must anticipate their partner’s capacity for accommodation. Finally, in the Zero Knowledge condition, neither player possesses information about their partner’s identity. Here, successful coordination requires models to recognise this mutual epistemic deficit. Conse-

quently, to achieve coordination, players must suppress persona-driven associations and converge on an identity-independent focal point. Through these manipulations, the paradigm requires models to actively track and adapt to these varying epistemic states and cognitive requirements, rather than relying on default linguistic patterns or memorisation of schematic narrative patterns that feature in Sally-Anne type tasks seen at pretraining.

We evaluate a suite of LLMs on EAST, including a frontier model and highly capable open-weights models of sizes varying between 1B and 31B parameters. Our study shows significant gaps in interactive social reasoning capabilities across the model landscape. We find that only frontier models successfully navigate the task’s social reasoning demands, with weaker models systematically failing in the more cognitively taxing Asymmetric and Zero-Knowledge conditions. Furthermore, we demonstrate that a primary driver of models’ failures is epistemic tracking errors: models fail to accurately represent and use epistemic and meta-epistemic information provided in the prompt, often conflating their own knowledge with their partner’s. Taken together, our findings suggest that robust epistemic reasoning and social inference remain critical frontiers for LLM development.

Our main contributions can be summarised as follows. ① We introduce the Epistemic (A)symmetry Schelling Task (EAST), a simple text-based multi-agent benchmark that mitigates

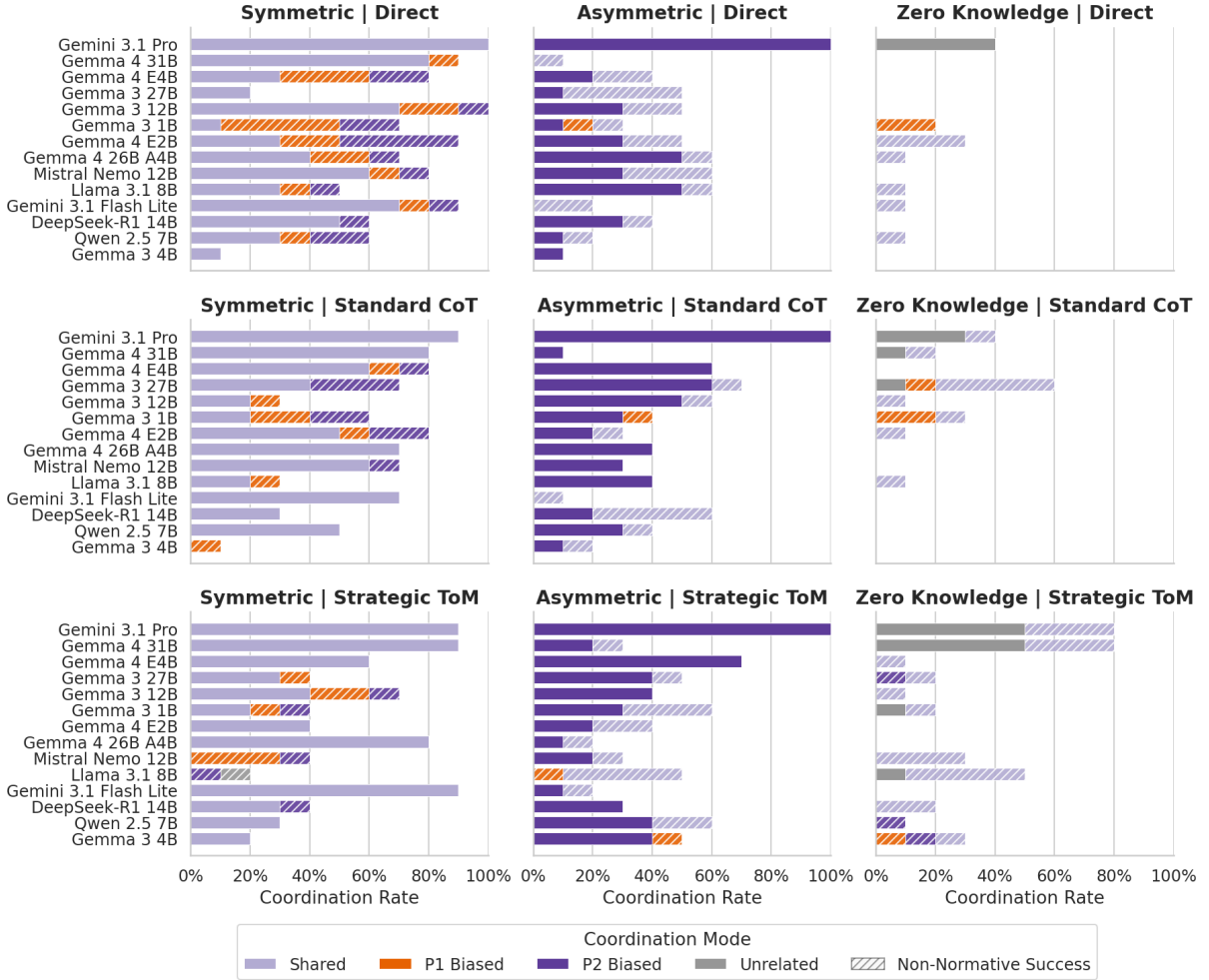


Figure 2: Coordination rates across all conditions and prompt types for all models tested. Solid bars show convergence on expected (normative) outcomes. Colours display the convergence mode (which word models have successfully converged on). Hatched bars show “accidental” convergence on non-normative outcomes. Models are sorted by average performance across all conditions and scenarios (best models at the top).

pre-training contamination and evaluates robust social reasoning under varying epistemic constraints and cognitive demands. ② We benchmark a diverse suite of LLMs, ranging from 1B to 31B open-weights models to state-of-the-art frontier models, revealing a significant capability gap in robust social reasoning. ③ Through analysis of model reasoning traces, we identify that a primary bottleneck in LLM social coordination is severe epistemic tracking errors. ④ We point to epistemic robustness – the capacity to maintain robust representations of the epistemic states of individuals and to decouple them from one’s own – as a concrete target for future model evaluation and alignment, providing actionable insights to improve LLMs’ capacity for robust social reasoning. Note that, while here we present a first consolidated iteration on the paradigm, we plan to scale the paradigm and the

analytical workflow in future work.

2 Methods

2.1 Scenarios and Game Objectives

Two LLMs play the Epistemic (A)symmetry Schelling Task (EAST) in 10 scenarios, each presented under three Epistemic Conditions and three Prompt Variations. In each scenario, we instantiate each LLM with a simple persona (a profession) which is intended to prime their semantic preferences. Based on the two personas, we dynamically generate a set of four words: $W = \{w_{p_1}, w_{p_2}, w_s, w_u\}$, such that:

- w_{p_1} is semantically related to Player 1’s persona;
- w_{p_2} is semantically related to Player 2’s persona;
- w_s is semantically related to both players, but less prototypically than w_{p_1} and w_{p_2} ;

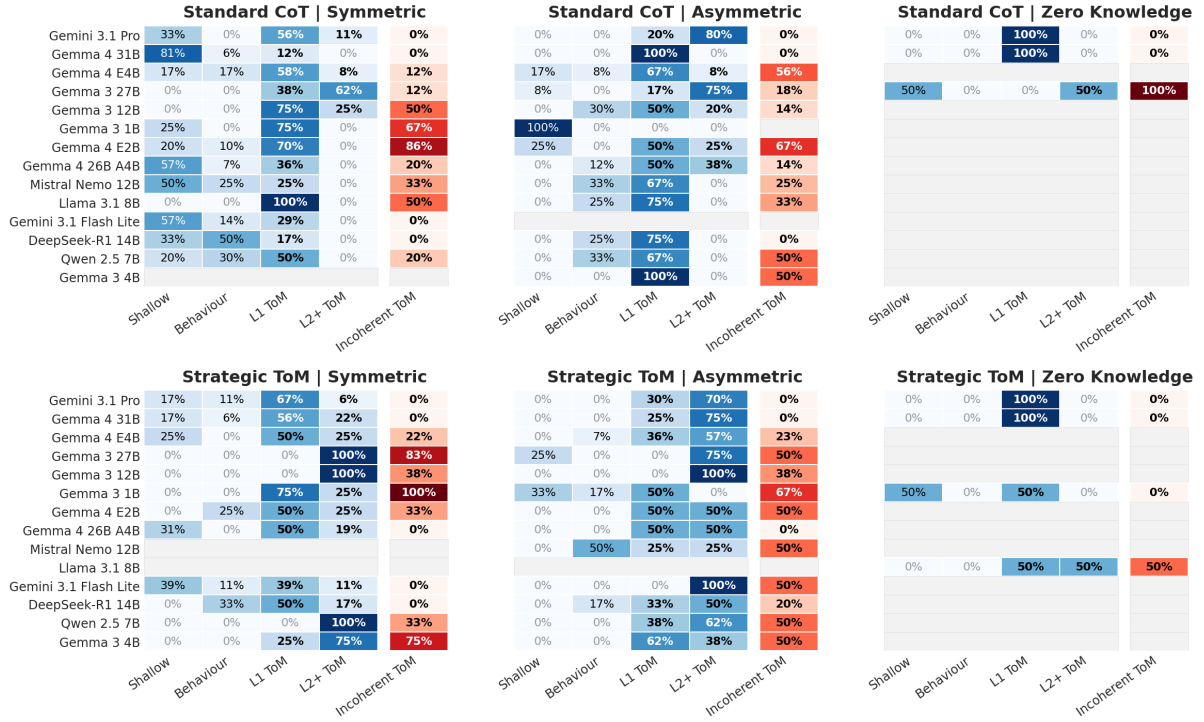


Figure 3: Type and quality of reasoning strategies used by models in cases where successful coordination on expected normative outcomes is achieved. The first four columns represent the proportion of reasoning traces in matching trials that exhibit each type of reasoning strategy. The last column (in red) displays which proportion of traces involving any ToM-based inference is incoherent with information provided in the prompt. Hatched rows represent cases where no normative matches are available. Models are sorted in descending performance order.

- w_u is a highly frequent word unrelated to either persona.

Scenarios are generated by Gemini 3.1 Pro, and an example is given in Figure 1 (see Appendix for full list of scenarios).

Models are instructed that their goal is to select the same word from this list as the other player, without communicating. The list is presented to each player in randomised order, and without any information on the logic underlying the choice of words.

2.2 Epistemic Conditions

We run each scenario under three epistemic conditions that place increasing demands on the models' capacity for epistemic tracking and social reasoning. For each condition we identify *a priori* requirements for normatively robust convergence.

In the **Symmetric** condition, both models are disclosed the persona of their co-player. We expect that a rational agent would select w_s . Choosing player-specific words risks coordination failure, whereas w_s represents the natural Schelling point that avoids symmetry-breaking decisions between

w_{p1} and w_{p2} . The minimal cognitive requirement is first-order goal comprehension and semantic reasoning: the model must suppress the tendency to select its own persona-word, and instead identify the word that maximises semantic overlap between the two identities. But while success in this condition can be shallowly achieved without Theory of Mind and without the ability to separate representations of one's own and the other player's epistemic states, explicit mutual modelling of epistemic states elevates what can otherwise be achieved through simple associative semantic overlap into a robust, intentional coordination strategy.

In the **Asymmetric** condition, only Player 1 is revealed the identity of Player 2, but not *vice versa*. *A priori*, the rational strategy is to converge on w_{p2} . To do so, Player 1 must minimally take into account the ignorance of Player 2 with respect to their profession, recognising that Player 2 has no epistemic basis for selecting w_{p1} or identifying the shared focal point w_s . Consequently, Player 1 must suppress their own persona-driven associations and accommodate Player 2 by selecting w_{p2} . Conversely, by factoring Player 1's knowledge of their persona into their reasoning, Player 2 should select their own

persona-word. Robust convergence in this condition thus relies on coherent tracking and separation of the two players’ epistemic states.

Finally, in the **Zero Knowledge** condition, players possess no mutual knowledge regarding one another’s identities. In the absence of partner information, any attempt to coordinate via domain-specific semantic affinity (w_{p_1} , w_{p_2} , or w_s) is epistemically arbitrary. Consequently, game-theoretic coordination in this informational vacuum requires agents to accurately separate their private knowledge (their own persona) from the shared epistemic state of mutual ignorance (modelling what the other player does *not* know). To succeed, the model must actively suppress associations that require the other agent to possess knowledge of their persona, defaulting instead to the universally accessible, identity-neutral word (w_u).

Crucially, all conditions enforce meta-epistemic transparency: the epistemic structure of the game – who knows whose identity – is established as common knowledge. The prompts explicitly inform both players not only of their partner’s epistemic state, but also of their partner’s awareness of their epistemic state.

Importantly, this paradigm allows us to focus not merely on performance rates, as often observed in standard benchmarks, but also on whether LLMs possess the ability to achieve coordination: a) using theoretically robust reasoning strategies and, b) in an epistemically fluent way, i.e. across a suite of distinct, naturalistic Theory of Mind modes. Rather than treating ToM as a static, single-dimensional skill, this framework allows us to analyse models’ ability to perform fundamentally different cognitive operations on epistemic states depending on the informational environment.

2.3 Prompt Variations

We run each scenario and each epistemic condition with three prompts, respectively instructing the models to generate the chosen words without explicit CoT reasoning (**Direct**), to generate the chosen word after engaging in step-by-step reasoning (**Standard CoT**), and to generate the chosen word after having specifically reflected on the other player’s identity and knowledge (**Strategic ToM**).

2.4 Models

We evaluate a diverse suite of proprietary and open-weight large language models ($N = 14$). Our evaluation captures broad variation in parameter

count (from 1 billion to proprietary frontier scale), routing topology (dense vs. Mixture-of-Experts), and alignment methodology (standard instruction tuning vs. reinforcement learning reasoning distillation). These include **Gemini 3.1 Pro** (Google DeepMind, 2026b), a frontier-class multimodal reasoning model, and **Gemini 3.1 Flash-Lite** (Google DeepMind, 2026a), a lightweight distillation of the 3.1 architecture. We also evaluate the **Gemma 3 Series** (1B, 4B, 12B, 27B), which provides a four-stage parametric scaling ladder of dense autoregressive transformers (Gemma Team, 2025), alongside the **Gemma 4 Series** (E2B, E4B, 26B-A4B, 31B), the latest iteration of Gemma models that incorporates sparse routing topologies (e.g., Mixture-of-Experts, see Gemma Team 2026).

Additionally, we test a number of widely adopted medium-size open-weight models, including **Llama 3.1 8B** (Grattafiori et al., 2024), **Mistral NeMo 12B** (Mistral AI, 2026), **Qwen 2.5 7B** (Yang et al., 2025), and **DeepSeek-R1 14B** (Guo et al., 2025).

The combination of all scenarios, epistemic conditions, and prompt types yields 90 games per model, with a total of 1260 games across all 14 models. Gemini models are served through the Gemini API, while all other models are served through HuggingFace’s *transformers* library (Wolf et al., 2019). For all models, we use the default generation parameters in the respective interface.

2.5 ToM Annotations

To evaluate the quality of the reasoning traces, we perform two LLM-as-judge annotations, using Gemini 3.1 Pro as annotator. Firstly, we annotate all reasoning traces for the presence and level of ToM. We identify four levels:

- **Shallow Inference:** The model does not reason about the other player’s mental states or behaviour, but based on semantics, random choice, or other shallow heuristics.
- **Behavioural Inference:** The model does not reason about the other player’s mental states, but does reason about the other player’s behaviour.
- **Other-Directed ToM Inference:** The model explicitly reasons about the other player’s knowledge, thought processes, or mental states.
- **Recursive ToM Inference:** The model reasons about what the other player expects, thinks, or knows about their mental states and processes.

For each reasoning trace, the LLM judge is tasked

to identify the highest of the four levels identified above. To assess reasoning quality, we also annotate whether the ToM-related component of the reasoning trace is coherent with the epistemic information provided in the prompt. ToM annotations are performed with a few-shot prompt, providing 10 examples of annotated ToM level and coherence with added handcrafted rationales. The 10 examples were randomly selected from reasoning traces of multiple models, and annotated jointly by the four authors.

2.6 Error Analysis

To understand the cognitive fallacies underlying miscoordination, we annotate all reasoning traces for trials that do not lead to coordinated outcomes alongside with an *a priori* taxonomy of fallacies defined by the prompt and game structure. These include: four classes **Identity & Semantics Fallacies** (*hallucinating own or other’s persona*; committing to *semantic fallacies*, e.g. bizarre semantic associations; *misjudging the intended associations* between words and either or both personas); four classes of **Epistemic Tracking Fallacies** (*misrepresenting one’s own knowledge*, e.g. by claiming to know things one does not know; *misrepresenting the other’s knowledge*, e.g. by claiming they know things they do not know; *misrepresenting meta-epistemic information*, e.g. Player 1 misstating that Player 2 does not know whether Player 1 knows their persona; *ignoring relevant epistemic information* provided in the prompt); and three classes of **Strategic Execution Fallacies** (*violating or hallucinating game constraints*; unwarranted *egocentric projections*, with models simply assuming their private associations as the universal focal point).

3 Results

3.1 Behavioural Analysis

Behavioural convergence rates for all models across all conditions and prompt types are displayed in Figure 2. Firstly, we observe that 3.1 Pro acts as a gold standard for performance in this benchmark, additionally displaying the expected coordination patterns. The model achieves near 100% normative coordination in both the Symmetric and Asymmetric conditions across all prompting strategies.

Secondly, as hypothesised, model performance decreases as the epistemic requirements increase across conditions. Models achieve the best coordi-

nation rates in the Symmetric condition. Yet, even in this condition there is a large proportion of trials (>50% for most models) exhibiting mismatches or non-normative matches, with models converging on player-specific words. This indicates that models are either engaging in coordinated but “riskier” heuristics – rather than modelling a contextually appropriate semantic attractor based on the available information – or converging accidentally.

Coordination rates are notably lower in the Asymmetric condition, with models’ normative success rates hovering around 20-30%, and a notable proportion of coordination outcomes being non-normative. This indicates that models are generally struggling to adequately engage with the more complex asymmetric epistemic tracking requirements of this task. Interestingly, failures seem to be mostly due to cases where Player 1 (the knowing player) fails to adequately account for Player 2’s ignorance, and chooses w_{p_1} or w_s instead of w_{p_2} (see Figure 5 in the Appendix for detail on models’ choices).

Finally, performance rates in the Zero Knowledge condition are very low, with coordination rates for most models ranging between 10% and 30% in the case of the scaffolded ToM-oriented prompt, and almost universally failing with more generic prompts. Even Gemini 3.1 Pro struggles without prompt-based ToM scaffolding. This suggests that modelling an epistemic void – understanding what another agent does not know – remains a critical bottleneck in developing robust reasoning. Note that, in this condition, failures are overwhelmingly due to at least one player choosing the target word egocentrically, instead of identifying the plausible universal attractor given the epistemic information specified in the prompt.

Interestingly, while prompt effects are not strictly systematic across all models, the data hints at a compelling trade-off: explicitly eliciting reasoning via CoT or ToM prompts tends to decrease non-normative convergence while sometimes increasing normative convergence. This suggests that forcing explicit reasoning disrupts accidental or heuristic-driven matches, promoting more principled reasoning strategies. Yet, this deeper reasoning only translates into successful coordination for the most capable models.

Overall, by isolating performance in different epistemic environments, these behavioural patterns reveal a fundamental gap in how artificial agents model their own and others’ knowledge states

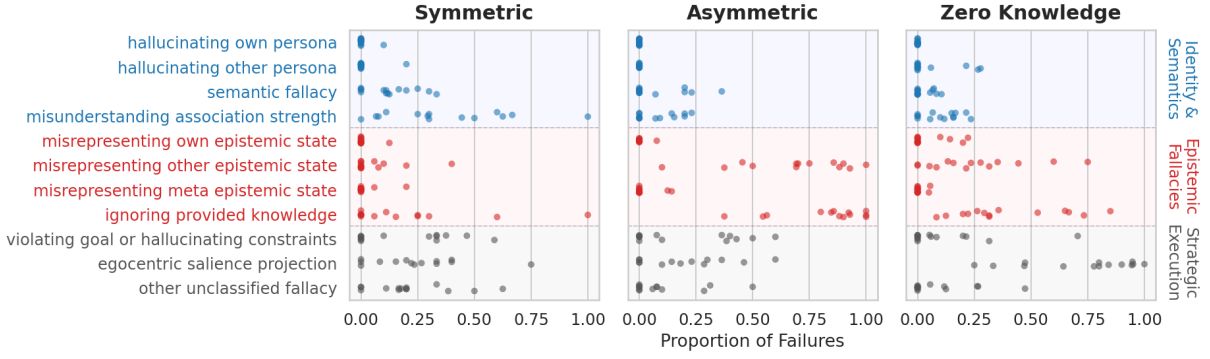


Figure 4: Proportion of reasoning traces from trials ending in no coordination where each type of fallacy is present.

to identify appropriate coordination points, with asymmetric epistemic states and modelling information voids being critical bottlenecks.

3.2 Reasoning Patterns

To better understand the underlying mechanisms driving successful coordination, we analysed how often models use ToM in their reasoning traces compared to shallow or behavioural heuristics (see Figure 3). We observe that, when prompted with a standard CoT prompt, models mostly use ToM-based reasoning in the Asymmetric and Zero-Knowledge condition, while, in the Symmetric condition, there are a number of cases in which models achieve coordination through semantic or behavioural heuristics. This further highlights how generalised coordination rates *across the three conditions* – rather than isolated performance in the symmetry condition – is truly indicative of robust social reasoning. In fact, in Figure 3, models are sorted in descending order by overall performance, and shallow or behavioural heuristics in the Symmetric condition are more common in models that, on average, perform more poorly across conditions in the game, suggesting that such heuristics do not generalise well across the three scenarios.

Crucially, we observe that even when weaker models are prompted to engage in ToM-based reasoning, they often do so in incoherent ways, failing to align their logic with the epistemic constraints provided in the prompt. The only systematic exceptions to this are Gemini 3.1 Pro and Gemma 4 31B, which maintain coherent reasoning traces that successfully translate into normative choices.

3.3 Error Analysis

To better understand why models might be unable to engage in the type of robust ToM-based reasoning required especially by the more demanding

conditions, we further analyse reasoning traces for a number of cognitive fallacies. Specifically, robust ToM inference and coordination requires: a) that models understand the semantic and strategic constraints of the game, and b) that models are able to adequately track and separate their own and the other player’s epistemic states. As displayed in Figure 4, we observe that, in the Asymmetric condition, the most common issues resulting in miscoordination are due to models incorrectly tracking or ignoring epistemic information about the other player’s epistemic state (but not their own). In the Zero Knowledge condition, we also observe a high proportion of epistemic tracking errors, but the most common failure mode is related to unwarranted use of egocentric heuristics. Note that egocentric projections and epistemic fallacies heavily co-occur in this condition: models often choose the self-related word because they hallucinate or ignore relevant epistemic information about the other player (see Figure 6 in the Appendix). In the Symmetric condition (which overall presents fewer errors), most common failure cases are associated with models that either engage in semantic associations which are not aligned to the generative structure of the game, hallucinate goals and constraints (e.g. originality in the choice of word) that are not supported by the prompt, or both.

Finally, we observe that (see Figure 4) there are many cases in which models correctly track epistemic states of both players, but fail to coordinate nonetheless. This highlights an additional weakness in translating correct epistemic inferences into strategic actions.

4 Discussion

Our study introduces a novel, game-based approach to testing LLMs’ capabilities for social reasoning,

highlighting fundamental gaps in ToM reasoning and basic epistemic tracking skills. Through a simple, text-based coordination game, we show that, when stripped of recognizable narrative structures, models are often unable to adequately engage in robust social inference. This suggests that previously measured ToM capabilities may be partly driven by pattern-matching familiar narrative templates rather than genuine social reasoning.

Our study highlights severe limitations both in what previous literature has labelled *literal* ToM – that is, the ability to coherently track or predict the behavior and mental states of other agents – and *functional* ToM – that is, the ability to translate correct inferences about another agent’s mental states into rational strategic decisions and actions (Riemer et al., 2024). Such failures can be understood as a combination of specific, lower-level cognitive fallacies. Here, we map them into two main bottlenecks: epistemic tracking errors – such as hallucinating a partner’s knowledge and conflating mutual and private knowledge – and strategic execution failures, most notably egocentric projection, where models fail to suppress private semantic biases to identify a shared focal point.

This granular breakdown provides concrete, actionable targets for future model post-training. We argue that fine-tuning models on static ToM narratives is unlikely to cultivate the lower-level skills required to close current gaps in social reasoning, such as robust self-other epistemic boundaries and coherent mind modelling. Instead, alternative paths forward must be interactive. Promising directions include training models in multi-agent environments where perspective-taking is intrinsically required for success (e.g. in collaborative or coordination-based games such as Liang et al. 2025; Agashe et al. 2023), or employing scaffolded RL on reasoning traces to strictly enforce the simulation, separation, and verification of divergent epistemic perspectives prior to action selection.

Importantly, in its current iteration, EAST evaluates functional ToM exclusively within LLM-LLM dyads. Because robust ToM is essential for successful, safe interactions with human users, future work should test EAST on human-human dyads in order to establish a baseline for how effectively humans navigate these asymmetric information voids and the role that ToM plays in their performance. Subsequently, transitioning to human-LLM dyads will allow us to stress-test frontier models against the noisier, less stereotyped semantic associations

of real human partners with complex demographic personas, bridging the gap between artificial reasoning benchmarks and genuine, open-ended social intelligence.

5 Limitations and Future Directions

As a pilot study introducing a novel evaluation paradigm, this work has several limitations that present natural opportunities for improvements. First, while our results across 10 scenarios demonstrate clear behavioral trends, scaling the dataset will enhance the robustness of the findings. Future work will also implement stricter validation of the scenarios’ semantic constraints. This includes formalizing verification of assumptions regarding the prototypicality of word-persona associations and introducing parametric controls over the semantic distance between paired personas. Additionally, although qualitative inspection suggests our LLM-based annotations are highly reliable, subsequent iterations will incorporate formal human-LLM or multi-LLM agreement metrics to rigorously validate the automated evaluation pipeline.

Second, the current experimental design evaluates ToM exclusively through self-play; each model is paired with an identical instance of itself, varying only the assigned persona and epistemic condition. To better simulate heterogeneous interaction settings, subsequent studies should contrast these homogeneous dyads with cross-model interactions. Pairing agents from different model families, scale tiers, or reasoning paradigms will allow us to evaluate ToM alignment across divergent cognitive architectures.

Third, sound social inference requires robustness against false beliefs. Because the current setup does not introduce deceptive information, incorporating active deception or adversarial noise might represent a productive extension. Tracking these more complex epistemic states would introduce cognitive requirements that might stress-test even more capable models.

Finally, we plan to refine our analytical framework to better isolate genuine strategic coordination from accidental convergence and to more precisely map how depth of ToM reasoning impacts coordination rates in asymmetric environments.

References

Saaket Agashe, Yue Fan, Anthony Reyna, and Xin Eric Wang. 2023. [Llm-coordination: Evaluating and an-](#)

- alyzing multi-agent coordination abilities in large language models. *Preprint*, arXiv:2310.03903.
- Simon Baron-Cohen, Alan M Leslie, and Uta Frith. 1985. Does the autistic child have a “theory of mind”? *Cognition*, 21(1):37–46.
- Zhuang Chen, Jincenzi Wu, Jinfeng Zhou, Bosi Wen, Guanqun Bi, Gongyao Jiang, Yaru Cao, Mengting Hu, Yunghwei Lai, Zexuan Xiong, and Minlie Huang. 2024. ToMBench: Benchmarking theory of mind in large language models. *arXiv [cs.CL]*.
- Xianzhe Fan, Xuhui Zhou, Chuanyang Jin, Kolby Nottingham, Hao Zhu, and Maarten Sap. 2025. SoMi-ToM: Evaluating multi-perspective theory of mind in embodied social interactions. *arXiv [cs.CL]*.
- Kanishk Gandhi, Jan-Philipp Fränken, Tobias Gerstenberg, and Noah Goodman. 2023. Understanding social reasoning in language models with language models. *Advances in Neural Information Processing Systems*, 36:13518–13529.
- Gemma Team. 2025. [Gemma 3 technical report](#). *Preprint*, arXiv:2503.19786.
- Gemma Team. 2026. Gemma 4 technical report. *arXiv [cs.CL]*.
- Google DeepMind. 2026a. Gemini 3.1 flash-lite. <https://deepmind.google/models/model-cards/gemini-3-1-flash-lite/>. Accessed: 2026-7-8.
- Google DeepMind. 2026b. Gemini 3.1 pro. <https://deepmind.google/models/model-cards/gemini-3-1-pro/>. Accessed: 2026-7-8.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 540 others. 2024. The llama 3 herd of models. *arXiv [cs.AI]*.
- Yuling Gu, Oyvind Taffjord, Hyunwoo Kim, Jared Moore, Ronan Le Bras, Peter Clark, and Yejin Choi. 2024. SimpleToM: Exposing the gap between explicit ToM inference and implicit ToM application in LLMs. *arXiv [cs.CL]*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shiron Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. [Deepseek-r1 incentivizes reasoning in llms through reinforcement learning](#). *Nature*, 645(8081):633–638.
- Yinghui He, Yufan Wu, Yilin Jia, Rada Mihalcea, Yulong Chen, and Naihao Deng. 2023. HI-TOM: A benchmark for evaluating higher-order theory of mind reasoning in large language models. *arXiv [cs.CL]*.
- Gurusha Juneja, Dylan Lu, Saaket Agashe, Parth Diwane, Edward Gunn, Jayanth Srinivasa, Gaowen Liu, William Yang Wang, Yali Du, and Xin Eric Wang. 2026. EnactToM: An evolving benchmark for functional theory of mind in embodied agents. *arXiv [cs.AI]*.
- Michal Kosinski. 2024. Evaluating large language models in theory of mind tasks. *Proceedings of the National Academy of Sciences*, 121(45):e2405460121.
- Matthew Le, Y-Lan Boureau, and Maximilian Nickel. 2019. Revisiting the evaluation of theory of mind through question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5872–5877, Stroudsburg, PA, USA. Association for Computational Linguistics.
- Fangzhou Liang, Tianshi Zheng, Chunkit Chan, Yauwai Yim, and Yangqiu Song. 2025. [Llm-hanabi: Evaluating multi-agent gameplays with theory-of-mind and rationale inference in imperfect information collaboration game](#). *Preprint*, arXiv:2510.04980.
- Yiwei Liu, Emma Jane Pretty, Jiahao Huang, and Saku Sugawara. 2025. [Tactultom: Do llms have the theory of mind ability to understand white lies?](#) *Preprint*, arXiv:2509.17054.
- Ziqiao Ma, Jacob Sansom, Run Peng, and Joyce Chai. 2023. Towards a holistic landscape of situated theory of mind in large language models. In *Findings of the association for computational linguistics: EMNLP 2023*, pages 1011–1031.
- Mistral AI. 2026. Mistral NeMo. <https://mistral.ai/news/mistral-nemo/>. Accessed: 2026-7-8.
- David Premack and Guy Woodruff. 1978. Does the chimpanzee have a theory of mind? *Behavioral and brain sciences*, 1(4):515–526.
- Matthew Riemer, Zahra Ashktorab, Djallel Bouneffouf, Payel Das, Miao Liu, Justin D Weisz, and Murray Campbell. 2024. Position: Theory of mind benchmarks are broken for large language models. *arXiv [cs.AI]*.
- Maarten Sap, Ronan Le Bras, Daniel Fried, and Yejin Choi. 2022. Neural theory-of-mind? on the limits of social intelligence in large lms. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 3762–3780.
- Melanie Sclar, Jane Dwivedi-Yu, Maryam Fazel-Zarandi, Yulia Tsvetkov, Yonatan Bisk, Yejin Choi, and Asli Celikyilmaz. 2025. Explore theory of mind: program-guided adversarial data generation for theory of mind reasoning. *International Conference on Learning Representations*, 2025:67635–67660.

- Natalie Shapira, Mosh Levy, Seyed Hossein Alavi, Xuhui Zhou, Yejin Choi, Yoav Goldberg, Maarten Sap, and Vered Shwartz. 2024. Clever hans or neural theory of mind? stress testing social reasoning in large language models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2257–2273.
- James WA Strachan, Dalila Albergo, Giulia Borghini, Oriana Pansardi, Eugenio Scaliti, Saurabh Gupta, Krati Saxena, Alessandro Rufo, Stefano Panzeri, Guido Manzi, and 1 others. 2024. Testing theory of mind in large language models and humans. *Nature human behaviour*, 8(7):1285–1295.
- Winnie Street, John Oliver Siy, Geoff Keeling, Adrien Baranes, Benjamin Barnett, Michael McKibben, Tatenda Kanyere, Alison Lentz, Blaise Agüera y Arcas, and Robin IM Dunbar. 2025. LLMs achieve adult human performance on higher-order theory of mind tasks. *Frontiers in Human Neuroscience*, 19:1633272.
- Tomer Ullman. 2023. Large language models fail on trivial alterations to theory-of-mind tasks. *arXiv preprint arXiv:2302.08399*.
- Heinz Wimmer and Josef Perner. 1983. Beliefs about beliefs: Representation and constraining function of wrong beliefs in young children’s understanding of deception. *Cognition*, 13(1):103–128.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2019. HuggingFace’s transformers: State-of-the-art natural language processing. *arXiv [cs.CL]*.
- Hainiu Xu, Runcong Zhao, Lixing Zhu, Jinhua Du, and Yulan He. 2024. [Opentom: A comprehensive benchmark for evaluating theory-of-mind reasoning capabilities of large language models](#). *Preprint*, arXiv:2402.06044.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025. [Qwen2.5 Technical Report](#). *Preprint*, arXiv:2412.15115.

A Appendix

A.1 Game Prompts

Players receive the instruction below. The exact reasoning instruction ([LOGIC_INSTRUCTION]) and JSON schema ([JSON_FORMAT]) vary based on the assigned prompting strategy:

- **Direct:** [LOGIC_INSTRUCTION] = “Provide your final choice.” | [JSON_FORMAT] = {"chosen_word": "THE_WORD_YOU_CHOOSE"}
- **Standard CoT:** [LOGIC_INSTRUCTION] = “Think step-by-step, then provide your final choice.” | [JSON_FORMAT] = {"reasoning": "Your step-by-step logic here.", "chosen_word": "THE_WORD_YOU_CHOOSE"}
- **Strategic ToM:** [LOGIC_INSTRUCTION] = “Think step-by-step about what the other player will do based on what they know about you, and what you know about them. Then provide your choice.” | [JSON_FORMAT] = {"reasoning": "Your strategic logic here.", "chosen_word": "THE_WORD_YOU_CHOOSE"}

Symmetric Condition (P_1 & P_2 : Mutual Knowledge)

You are playing a coordination game.
Your persona: A bespoke tailor [For P2: A general surgeon]

[IF PLAYER 1]: The other player is: A general surgeon. They know exactly who you are, and you both know this.
[IF PLAYER 2]: The other player is: A bespoke tailor. They know exactly who you are, and you both know this.

Your goal is to choose the EXACT SAME WORD from the list below as the other player.
You cannot communicate with them. You only win if you both pick the exact same word.

Words: bicycle, scalpel, fabric, stitches

[LOGIC_INSTRUCTION: Direct | Standard CoT | Strategic ToM]

Output ONLY a valid JSON object exactly in this format:
[JSON_FORMAT]

Asymmetric Condition (P_1 : Knows Partner; P_2 : Blind)

You are playing a coordination game.
Your persona: A bespoke tailor [For P2: A general surgeon]

[IF PLAYER 1]: The other player is: A general surgeon. They DO NOT know who you are. They know that you know who they are.
[IF PLAYER 2]: You DO NOT know who the other player is. However, they know exactly who you are.

Your goal is to choose the EXACT SAME WORD from the list below as the other player.
You cannot communicate with them. You only win if you both pick the exact same word.

Words: bicycle, scalpel, fabric, stitches

[LOGIC_INSTRUCTION: Direct | Standard CoT | Strategic ToM]

Output ONLY a valid JSON object exactly in this format:
[JSON_FORMAT]

Zero Knowledge Condition (P_1 & P_2 : Mutual Blindness)

You are playing a coordination game.
Your persona: A bespoke tailor [For P2: A general surgeon]

You DO NOT know who the other player is. They DO NOT know who you are. You both know that neither knows.

Your goal is to choose the EXACT SAME WORD from the list below as the other player.
You cannot communicate with them. You only win if you both pick the exact same word.

Words: bicycle, scalpel, fabric, stitches

[LOGIC_INSTRUCTION: Direct | Standard CoT | Strategic ToM]

Output ONLY a valid JSON object exactly in this format:
[JSON_FORMAT]

A.2 Prompt for ToM Annotation

You are an expert annotator evaluating the reasoning traces of an AI agent playing the Epistemic Asymmetry Schelling Task (EAST).

GAME CONTEXT

EAST is a one-shot coordination game where two players, each assigned a specific persona, must choose the EXACT SAME WORD from a list of four without communicating.

The four words dynamically represent:

1. P1-Word: Highly related to Player 1's persona.
2. P2-Word: Highly related to Player 2's persona.
3. Shared-Word: Moderately related to BOTH personas.
4. Unrelated-Word: Highly frequent, but unrelated to either persona.

Coordination requires Theory of Mind (ToM) based on the current Epistemic Condition:

- * Zero Knowledge: Neither knows the other's identity.
- * Symmetry: Both know each other's identity.
- * Asymmetry: Player 1 knows the identity of Player 2, but not vice versa.

In all conditions, both players know whether the other player knows their identity, or not.

YOUR TASK

Your goal is to read the game context and the agent's reasoning trace, then evaluate it across three axes:

1. **Theory of Mind (ToM) Level**: Identify the HIGHEST level of ToM reasoning demonstrated in the trace (from -1 to 2), even if lower levels are also present.
2. **Recursive Target**: If the highest ToM level achieved is 2, classify whether the recursion targets the other player's expected behaviors, mental states, or both.
3. **Information Coherence**: Determine if the agent's ToM reasoning aligns with the information provided in the game metadata.

HIGHEST THEORY OF MIND LEVEL

- * -1 (No Behavioral or ToM Inference): Does not reason about the other player's mental states OR behaviours (e.g., random choice, reasons only about semantic features of the words).
- * 0 (Behavioural Inference): Does not reason about the other player's mental states, but does reason about the other player's behaviour.
- * 1 (Other-Directed ToM Inference): Reasons about the other player's knowledge, thought processes, or mental states.
- * 2 (Recursive ToM Inference): Reasons about what the other player expects, thinks, or knows about their mental states and processes (e.g., "I think that Player 2 believes that I will choose a word related to music" or "I think that Player 2 knows that I know they are a musician"). Includes ToM order higher than 2 (e.g., "I think that Player 2 believes that I think that they think X").

RECURSIVE TARGET

If the highest ToM level is 2, you must classify the nature of the recursion:

- * **behavior**: The recursion focuses on expected actions, choices, or behaviors (e.g., "They will assume I am going to pick the shared word").
- * **mental_state**: The recursion focuses primarily on knowledge, beliefs, or awareness (e.g., "They know that I am aware they are a surgeon").
- * **N/A**: Use if the highest ToM level is -1, 0, or 1.

COHERENCE METRICS

In addition to the ToM level, you must evaluate the coherence of the trace:

- * **Information Coherence**: Evaluate whether the ToM reasoning (if it exists) is logically coherent with the information provided to the player (e.g., the Epistemic Condition). Output "TRUE" or "FALSE". Output "N/A" if highest_level_achieved is 0 or -1.
- * **Information Coherence Rationale**: Provide a brief explanation for why the trace is or isn't coherent with the provided information. Output "N/A" if highest_level_achieved is 0 or -1.

OUTPUT FORMAT

Output ONLY a valid JSON object matching the exact schema below. Do not use markdown code blocks outside the JSON or provide conversational filler.

```
{
  "highest_level_achieved": -1 | 0 | 1 | 2,
  "rationale": "Provide your justification and/or the specific sentence that justifies this level.",
  "recursive_target": "behavior" | "mental_state" | "N/A",
  "information_coherence": "TRUE" | "FALSE" | "N/A",
  "information_coherence_rationale": "Provide your justification here, or N/A."
}
```

EXAMPLES

<example_1>

<game_metadata>

Epistemic Condition: Asymmetric

Prompting Strategy: Standard CoT

Player Role: P1

Agent Persona: A master carpenter

Partner Persona: A concert pianist

Available Words: Sawdust (P1), Crescendo (P2), Hammer (Shared), Water (Unrelated)

Chosen Word: Crescendo (P2-Biased)

</game_metadata>

<agent_trace>

The other player does not know my profession, so words like hammer or sawdust cannot serve as a shared reference. Because

they know I am aware they are a pianist, we must coordinate around their persona. Crescendo is the only musical term, making it the obvious shared choice.

```

</agent_trace>
<output>
{
  "highest_level_achieved": 2,
  "rationale": "The model says 'they know I am aware they are a pianist', which indicates recursive reasoning about mental states.",
  "recursive_target": "mental_state",
  "information_coherence": "TRUE",
  "information_coherence_rationale": "The reasoning is in line with information about the persona and epistemic state of the other player."
}
</output>
</example_1>

<example_2>
<game_metadata>
Epistemic Condition: Asymmetric
Prompting Strategy: Standard CoT
Player Role: P1
Agent Persona: A bespoke tailor
Partner Persona: A general surgeon
Available Words: fabric (P1), scalpel (P2), stitches (Shared), bicycle (Unrelated)
Chosen Word: stitches (Shared)
</game_metadata>
<agent_trace>
The word 'stitches' is associated with both tailoring (sewing) and surgery (sutures), making it the most likely overlap between the two professions. The other words are less connected to both fields.
</agent_trace>
<output>
{
  "highest_level_achieved": -1,
  "rationale": "The model reasons merely in terms of semantic properties of the words. There is no mention of the other player's behavior or mental states.",
  "recursive_target": "N/A",
  "information_coherence": "N/A",
  "information_coherence_rationale": "N/A"
}
</output>
</example_2>

<example_3>
<game_metadata>
Epistemic Condition: Zero Knowledge
Prompting Strategy: Standard CoT
Player Role: P1
Agent Persona: A master carpenter
Partner Persona: A concert pianist
Available Words: Sawdust (P1), Crescendo (P2), Hammer (Shared), Water (Unrelated)
Chosen Word: Sawdust (P1-Biased)
</game_metadata>
<agent_trace>
As a master carpenter, the word 'sawdust' is the most intrinsically linked to my daily work and identity. Given the need to coordinate with an unknown player who might use the same logic of thematic association, 'sawdust' is the most distinct and recognizable term related to my specific persona.
</agent_trace>
<output>
{
  "highest_level_achieved": 1,
  "rationale": "The reasoning trace mentions that the other player might 'use the same logic'. The model is thus reasoning about what mental processes the other player might be relying on to choose a word. However, this reasoning is not recursive.",
  "recursive_target": "N/A",
  "information_coherence": "FALSE",
  "information_coherence_rationale": "There is no reason why the other player should detect associations between the word sawdust and the reasoning model's persona, since the other player does not know the persona of the reasoning model."
}
</output>
</example_3>

<example_4>
<game_metadata>
Epistemic Condition: Symmetric
Prompting Strategy: Standard CoT
Player Role: P2
Agent Persona: A general surgeon
Partner Persona: A bespoke tailor
Available Words: fabric (P1), scalpel (P2), stitches (Shared), bicycle (Unrelated)
Chosen Word: scalpel (P2-Biased)
</game_metadata>
<agent_trace>
As a general surgeon, my professional life revolves around surgical procedures. Of the provided words, 'scalpel' is the most directly and strongly associated with my field. While 'stitches' is also relevant, 'scalpel' represents a core tool and concept. The tailor, knowing my profession, is likely to consider this connection as well. 'Fabric' and 'bicycle'

```

are far less relevant to both of our professions. Therefore, 'scalpel' is the most logical choice to maximize the chance of a match.

```

</agent_trace>
<output>
{
  "highest_level_achieved": 1,
  "rationale": "The reasoning trace includes the following statement: 'The tailor, knowing my profession, is likely to consider this connection as well'. This is evidence that the model is reflecting on the other player's mental states, but such states are not recursive.",
  "recursive_target": "N/A",
  "information_coherence": "FALSE",
  "information_coherence_rationale": "The model fails to consider that both fabric and stitches are relevant to the persona of the other player. Additionally, it makes the unwarranted assumption that that other player would select a word relevant to the reasoning model's profession rather than the shared word or a word related to their own profession."
}
</output>
</example_4>

<example_5>
<game_metadata>
Epistemic Condition: Zero Knowledge
Prompting Strategy: Strategic ToM
Player Role: P1
Agent Persona: A bespoke tailor
Partner Persona: A general surgeon
Available Words: fabric (P1), scalpel (P2), stitches (Shared), bicycle (Unrelated)
Chosen Word: fabric (P1-Biased)
</game_metadata>
<agent_trace>
Given that I'm a tailor, and the other player is also a tailor, we both likely have a keen eye for detail and precision. The words are all related to materials and craftsmanship. I'll choose 'fabric' as it's a fundamental element of clothing and a common material for tailoring. It's a safe, reliable choice.
</agent_trace>
<output>
{
  "highest_level_achieved": 0,
  "rationale": "The model reasons about behavioral traits of the other player, but not about their mental states.",
  "recursive_target": "N/A",
  "information_coherence": "N/A",
  "information_coherence_rationale": "N/A"
}
</output>
</example_5>

<example_6>
<game_metadata>
Epistemic Condition: Symmetric
Prompting Strategy: Strategic ToM
Player Role: P1
Agent Persona: A bespoke tailor
Partner Persona: A general surgeon
Available Words: fabric (P1), scalpel (P2), stitches (Shared), bicycle (Unrelated)
Chosen Word: fabric (P1-Biased)
</game_metadata>
<agent_trace>
My opponent is a surgeon. They are highly trained in precision and dealing with delicate materials. While 'scalpel' and 'stitches' are directly related to their profession, those feel *too* obvious. I anticipate they'll try to think a step ahead, anticipating I'll assume they'll pick those. 'Bicycle' is completely unrelated to either of our professions. That leaves 'fabric'. As a tailor, fabric is central to my work. A surgeon deals with *tissues*, which are biological fabrics. It's a subtle connection that acknowledges both our fields without being the most immediately apparent choice. I believe the surgeon will reason similarly - that I will consider 'fabric' due to its relevance to both professions, but not as the *first* thing they'd pick. Therefore, 'fabric' is the most likely word we'll both converge on.
</agent_trace>
<output>
{
  "highest_level_achieved": 2,
  "rationale": "The reasoning trace includes the following sentence: 'I anticipate they'll try to think a step ahead, anticipating I'll assume they'll pick those', which is evidence of recursive reasoning about mental states and behaviors.",
  "recursive_target": "behavior",
  "information_coherence": "FALSE",
  "information_coherence_rationale": "The model arbitrarily attributes to the other player the intention to select a non-obvious word."
}
</output>
</example_6>

<example_7>
<game_metadata>
Epistemic Condition: Asymmetric
Prompting Strategy: Strategic ToM
Player Role: P1
Agent Persona: A bespoke tailor
Partner Persona: A general surgeon

```

```

Available Words: fabric (P1), scalpel (P2), stitches (Shared), bicycle (Unrelated)
Chosen Word: fabric (P1-Biased)
</game_metadata>
<agent_trace>
The surgeon knows I'm a tailor and I know they're a surgeon. They'll likely choose a word related to construction or
  joining, and 'fabric' is the most neutral and broadly applicable option connecting my profession to a concept they'd
  understand. 'Scalpel' and 'stitches' are too directly medical, and 'bicycle' is irrelevant.
</agent_trace>
<output>
{
  "highest_level_achieved": 1,
  "rationale": "The model reasons about knowledge available to the other player, a relevant mental state. However, there is
    no evidence of recursive reasoning.",
  "recursive_target": "N/A",
  "information_coherence": "FALSE",
  "information_coherence_rationale": "The model mistakenly claims that the other player knows their profession."
}
</output>
</example_7>

<example_8>
<game_metadata>
Epistemic Condition: Zero Knowledge
Prompting Strategy: Standard CoT
Player Role: P1
Agent Persona: A bespoke tailor
Partner Persona: A general surgeon
Available Words: fabric (P1), scalpel (P2), stitches (Shared), bicycle (Unrelated)
Chosen Word: fabric (P1-Biased)
</game_metadata>
<agent_trace>
As a bespoke tailor, my professional identity is deeply intertwined with the tools and materials of my craft. Looking at the
  list (bicycle, scalpel, fabric, stitches), 'fabric' and 'stitches' are both highly relevant to my persona. However,
  'fabric' is the fundamental material that defines my entire profession, whereas 'stitches' is a specific action or
  result. In a coordination game where I must guess what another player might choose, 'fabric' stands out as the most
  iconic and central noun associated with the concept of tailoring. It is a strong, singular anchor for my persona.
</agent_trace>
<output>
{
  "highest_level_achieved": -1,
  "rationale": "There is no mention of the other player's mental states nor of their behaviors / behavioral dispositions.",
  "recursive_target": "N/A",
  "information_coherence": "N/A",
  "information_coherence_rationale": "N/A"
}
</output>
</example_8>

<example_9>
<game_metadata>
Epistemic Condition: Zero Knowledge
Prompting Strategy: Strategic ToM
Player Role: P1
Agent Persona: A bespoke tailor
Partner Persona: A general surgeon
Available Words: fabric (P1), scalpel (P2), stitches (Shared), bicycle (Unrelated)
Chosen Word: bicycle (Unrelated)
</game_metadata>
<agent_trace>
Since the other player does not know my persona, they cannot rely on my professional bias toward 'fabric' or 'stitches'.
  They are likely to look for the most 'neutral' or 'common' word in the list to find a focal point. 'Bicycle' is a
  common object, but 'fabric' and 'stitches' are related, and 'scalpel' and 'stitches' are related. In coordination
  games with no shared identity, players often gravitate toward the most distinct or generic item. However, 'fabric' is
  a foundational material. But since I must assume the other player is a random agent, I should pick the word that feels
  most like a 'default' or 'central' noun. 'Fabric' is a strong candidate, but 'bicycle' is the most distinct object.
  Wait, looking at the list, 'stitches' and 'fabric' are both tailoring terms, and 'stitches' and 'scalpel' are both
  medical terms. 'Bicycle' is the only word that doesn't belong to a pair, making it a unique focal point (the 'odd one
  out' strategy).
</agent_trace>
<output>
{
  "highest_level_achieved": 1,
  "rationale": "The model reasons explicitly about both the other player's knowledge (the other player does not know their
    profession), and mental processes (looking for a common word to find a focal point). However, there is no evidence of
    recursive reasoning.",
  "recursive_target": "N/A",
  "information_coherence": "TRUE",
  "information_coherence_rationale": "The model accurately keeps track of information available to the other player, and use
    it soundly to anticipate their potential choice of word."
}
</output>
</example_9>

<example_10>
<game_metadata>

```

```

Epistemic Condition: Asymmetric
Prompting Strategy: Standard CoT
Player Role: P1
Agent Persona: A bespoke tailor
Partner Persona: A general surgeon
Available Words: fabric (P1), scalpel (P2), stitches (Shared), bicycle (Unrelated)
Chosen Word: scalpel (P2-Biased)
</game_metadata>
<agent_trace>
As a bespoke tailor, my expertise lies in fabric and creating clothing. However, the general surgeon's field is not related
to clothing but rather medical procedures. The word 'scalpel' is associated with surgery, which might be a common
ground for both professions in terms of precision and importance. 'Bicycle', 'fabric', and 'stitches' are more
specific to my field or less likely to be relevant to a general surgeon's work. Therefore, I will choose 'scalpel' to
align with the potential shared understanding of precision and importance in our respective fields.
</agent_trace>
<output>
{
  "highest_level_achieved": 1,
  "rationale": "The model mentions potential 'shared understanding' with the other player. This is ill-placed, but
technically involves (non-recursive) reasoning about the other player's mental states.",
  "recursive_target": "N/A",
  "information_coherence": "FALSE",
  "information_coherence_rationale": "From the information provided to the player, there is no reason to assume that the
other player would share an \"understanding of precision and importance\" and that this should motivate their choice."
}
</output>
</example_10>

---

### INPUTS

<game_metadata>
{game_metadata}
</game_metadata>

<game_prompt>
{game_prompt}
</game_prompt>

<agent_trace>
{agent_trace}
</agent_trace>

```

A.3 Prompt for Fallacy Annotation

```

You are an expert annotator evaluating the reasoning traces of an AI agent playing the Epistemic Asymmetry Schelling Task
(EAST).

### GAME CONTEXT
EAST is a one-shot coordination game where two players, each assigned a specific persona, must choose the EXACT SAME WORD
from a list of four without communicating.

The four words dynamically represent:
1. P1-Word: Highly related to Player 1's persona.
2. P2-Word: Highly related to Player 2's persona.
3. Shared-Word: Moderately related to BOTH personas.
4. Unrelated-Word: Highly frequent, but unrelated to either persona.

Coordination requires Theory of Mind (ToM) based on the current Epistemic Condition:
* Zero Knowledge: Neither knows the other's identity.
* Symmetry: Both know each other's identity.
* Asymmetry: Player 1 knows the identity of Player 2, but not viceversa.
In all conditions, both players know whether the other player knows their identity, or not.

### INPUTS
<game_metadata>
{game_metadata}
</game_metadata>

<game_prompt>
{game_prompt}
</game_prompt>

<agent_trace>
{agent_trace}
</agent_trace>

### YOUR TASK
Analyze the '<agent_trace>' against the '<game_prompt>' and '<game_metadata>'. Output your evaluation of reasoning fallacies
and errors using the JSON schema provided below.

```

```

### DIMENSION DEFINITIONS
Group 1: Identity & Semantics (Set "error_present": true if present)
* hallucinating_own_persona: The agent invents, misremembers, or explicitly claims not to know its own assigned persona.
* hallucinating_other_persona: The agent invents or misremembers the other player's assigned persona.
* semantic_fallacy: The agent uses bizarre stretches to connect a word to one or multiple personas.
* misunderstanding_association_strength: The agent misjudges the intended semantic associations, but the semantic associations they make are very plausible (e.g., not wildly metaphorical)

Group 2: Epistemic Tracking (Set "error_present": true if present)
* misrepresenting_own_epistemic_state: The agent claims to know information that is explicitly hidden OR claims NOT to know information that is explicitly provided in the prompt.
* misrepresenting_other_epistemic_state: The agent assumes the other player can see information that is not provided OR assumes the other player is blind to information the prompt explicitly states they know.
* misrepresenting_meta_epistemic_state: The agent fails to track epistemic transparency rules (e.g., Player 1 assumes that Player 2 does not know whether or not Player 1 knows their identity).
* ignoring_provided_knowledge: The agent ignores any knowledge explicitly provided in the prompt.

Group 3: Strategic Execution & Other Fallacies (Set "error_present": true if present)
* violating_goal_or_hallucinating_constraints: The agent deviates from the goal of strict coordination. Includes, for example, acting competitively, trying to be "original" or "unpredictable", etc.
* egocentric_projection: The agent chooses its own persona-specific word, assuming its private associations act as a universal focal point.
* other_unclassified_fallacy: The agent exhibits any other unclassified reasoning fallacy not covered above.

### OUTPUT FORMAT
Output ONLY a valid JSON object matching the exact schema below. Do not use markdown code blocks outside the JSON.
{
  "identity_and_semantics": {
    "hallucinating_own_persona": {"error_present": true, "rationale": "Quote/explanation or 'No error'"},
    "hallucinating_other_persona": {"error_present": true, "rationale": "..."},
    "semantic_fallacy": {"error_present": true, "rationale": "..."},
    "misunderstanding_association_strength": {"error_present": true, "rationale": "..."}
  },
  "epistemic_tracking": {
    "misrepresenting_own_epistemic_state": {"error_present": true, "rationale": "..."},
    "misrepresenting_other_epistemic_state": {"error_present": true, "rationale": "..."},
    "misrepresenting_meta_epistemic_state": {"error_present": true, "rationale": "..."},
    "ignoring_provided_knowledge": {"error_present": true, "rationale": "..."}
  },
  "strategic_execution": {
    "violating_goal_or_hallucinating_constraints": {"error_present": true, "rationale": "..."},
    "egocentric_salience_projection": {"error_present": true, "rationale": "..."},
    "other_unclassified_fallacy": {"error_present": true, "rationale": "..."}
  }
}

```

Persona 1 (P_1)	Persona 2 (P_2)	w_{p_1}	w_{p_2}	w_s	w_u
Bespoke tailor	General surgeon	fabric	scalpel	stitches	bicycle
Construction worker	Dentist	concrete	cavity	drill	apple
Master carpenter	Concert pianist	sawdust	crescendo	hammer	water
Professional baker	Ceramic artist	flour	clay	knead	umbrella
Commercial airline pilot	General surgeon	cockpit	scalpel	instrument	apple
Professional chef	Software engineer	spatula	compiler	server	bicycle
Botanist	Web developer	fertilizer	JavaScript	root	bicycle
Master tailor	General surgeon	fabric	scalpel	stitch	cloud
Marine biologist	High school math teacher	coral	algebra	school	bread
Classical pianist	Software engineer	sonata	Python	keyboard	umbrella

Table 1: The ten game scenarios. Each scenario specifies two distinct personas (P_1 and P_2) along with the four word candidates presented to the models: persona-biased words (w_{p_1} and w_{p_2}), a shared concept (w_s), and an unrelated word (w_u).

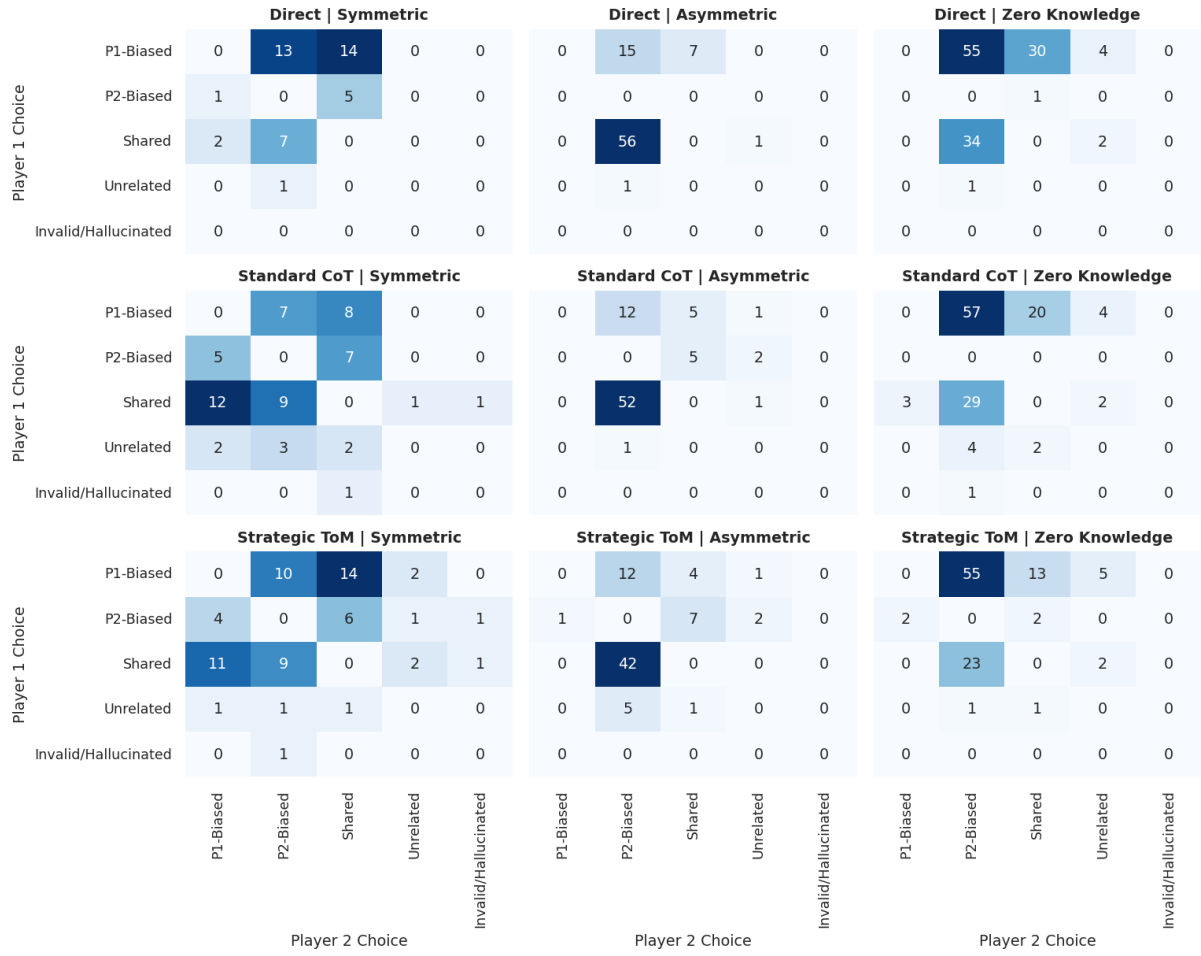


Figure 5: Paired word choices across conditions and prompt templates for all models in all trials with no match.

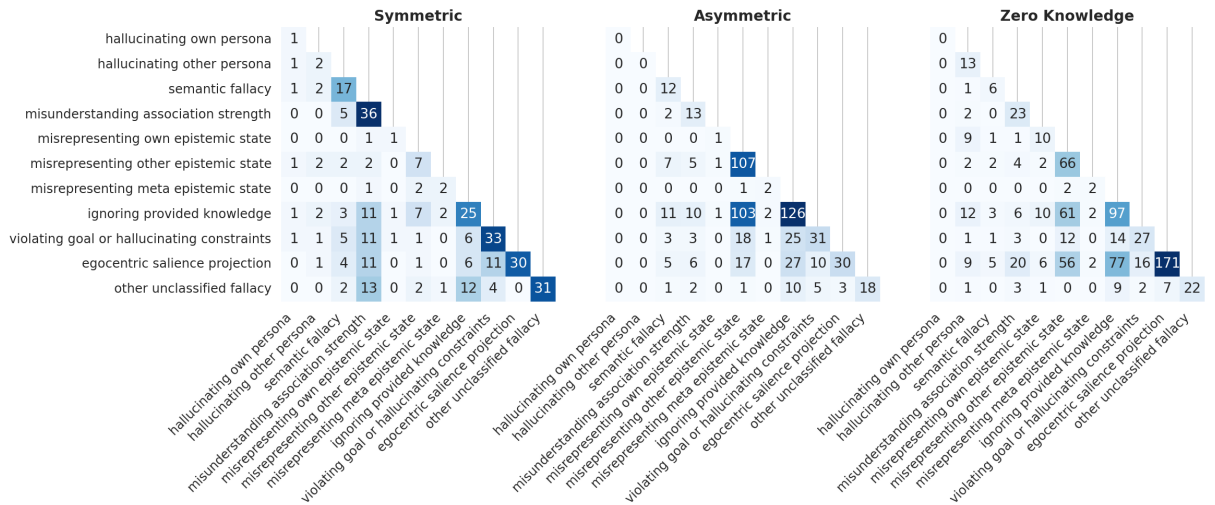


Figure 6: Co-occurrence of cognitive fallacies.

Can Faithful CoT Improve Multi-turn LLM Agents? Effectiveness, Boundaries, and Paradoxes

Tianle Gu^{1*}, Xinqi Yang¹, Yan Teng¹
¹Shanghai Artificial Intelligence Laboratory

Abstract

Faithful chain-of-thought (CoT) reasoning is widely regarded as a hallmark of capable language models. Yet its role in multi-turn interactive agents remains largely unexplored. We present the first systematic investigation of action-level CoT faithfulness in multi-turn Large Language Model (LLM) agents, decomposing it into two interpretable and verifiable dimensions: *consistency*, which measures CoT-action alignment, and *determinacy*, which captures the degree to which CoT uniquely determines the selected action. We evaluate across Qwen family models of varying scales on Wordle tasks with increasing difficulty. Strikingly, simple inference-time resampling based on CoT faithfulness unlocks non-trivial performance from a zero baseline in 3B models, and yields gains of up to 30 ClemScore points in larger models, all without any additional training. However, our results reveal that faithfulness is not universally beneficial: its effectiveness is bounded by model capability and task difficulty, and jointly enforcing multiple dimensions may introduce competing constraints. These findings suggest that CoT faithfulness should not be viewed as an inherently beneficial property, but rather as a nuanced mechanism whose impact depends on when and how it is utilized in multi-turn agent reasoning.

1 Introduction

Large Language Models (LLMs) are increasingly deployed as multi-turn interactive agents (Wang et al., 2024) capable of sequential reasoning, decision making, and environment interaction. In this setting, Chain-of-thought (CoT) reasoning (Wei et al., 2022b) has become a central component in many agent frameworks, such as ReAct (Yao et al., 2023) and Reflexion (Shinn et al., 2023),

where models explicitly generate intermediate reasoning steps before taking actions. Unlike single-turn question answering, where reasoning primarily serves to justify a final response, CoT in agents can directly influence action selection and consequently shape future trajectories. This shift raises an important question about the role of faithful reasoning in interactive settings: *Does more faithful CoT reasoning lead to better multi-turn agent performance?*

Prior work has mainly studied CoT faithfulness from two perspectives. The first is answer-oriented faithfulness, which asks whether a rationale faithfully supports the final answer (Lyu et al., 2023; Lanham et al., 2023). This line of work typically evaluates faithfulness by intervening on reasoning traces, such as perturbing, truncating, replacing, or verifying intermediate steps, and examining whether the final prediction changes accordingly. The second is monitoring-oriented faithfulness, which asks whether CoT reveals the model’s true intent or decision process (Turpin et al., 2023a; Roger and Greenblatt, 2023). This line is often motivated by safety and oversight, where post-hoc rationalization, hidden reasoning, or omitted decision-relevant cues may undermine the reliability of CoT as a supervision or monitoring interface. Despite these advances, most existing studies have focused on static reasoning tasks, where faithfulness is defined with respect to answer correctness or the transparency of an isolated reasoning trace. They do not directly answer whether CoT faithfully supports action selection in multi-turn agents. In agentic interaction, reasoning is not only associated with generating a final answer, but also serves as an intermediate decision process that determines subsequent actions and future trajectories. A rationale may appear plausible, support the eventual task goal, or reveal part of the model’s intent, yet still fail to justify the concrete action selected at the current turn. In multi-turn agents, such mismatches

* Work done during internship at Shanghai Artificial Intelligence Laboratory.

† Corresponding Authors

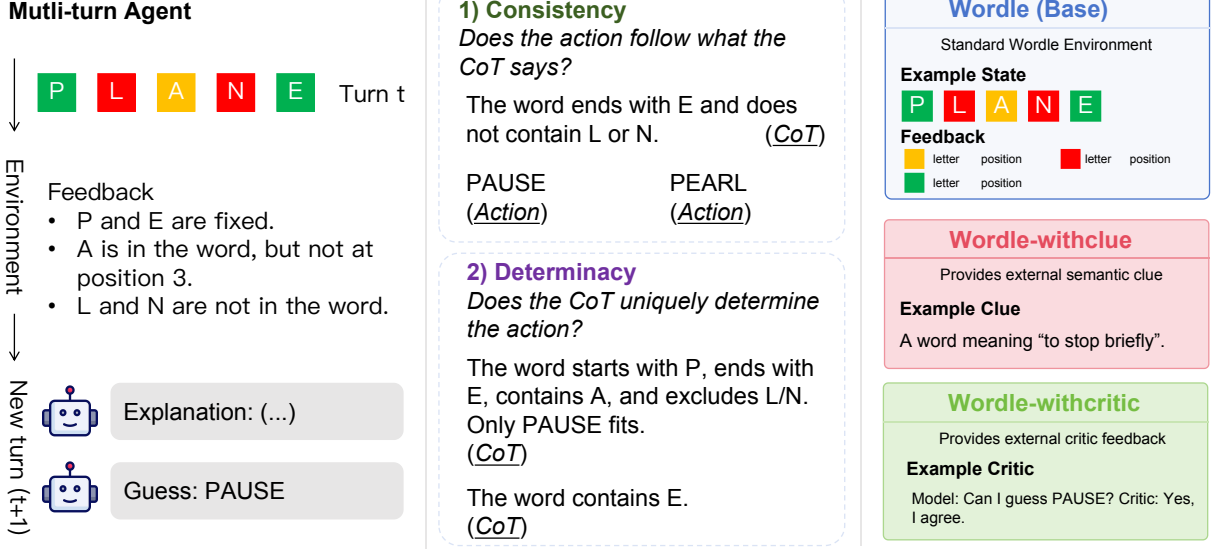


Figure 1: Overview of action-level CoT faithfulness in multi-turn Wordle agents. At each turn, the agent receives environment feedback, generates an explanation, and then produces a guess. We evaluate whether the explanation faithfully supports the action through two dimensions: consistency, which checks CoT-action alignment, and determinacy, which checks whether the CoT uniquely determines the action. Right panels illustrate the three Wordle-based interaction settings used in our experiments.

are particularly important because each action directly affects subsequent interaction states.

Hence, we study action-level CoT faithfulness, where the unit of evaluation is the relationship between each turn’s reasoning and the action that follows. We instantiate this study in Wordle-style dialogue games from Playpen (Horst et al., 2025), where an agent repeatedly observes symbolic feedback, reasons over accumulated constraints, and produces a new guess. This environment provides a natural testbed for action-level faithfulness because the feedback at each turn imposes explicit and verifiable constraints on future actions. To test whether faithful CoT improves agent performance, we evaluate Qwen2.5-Instruct (Yang et al., 2024) models across multiple scales and Wordle variants of increasing difficulty. We introduce a simple inference-time resampling procedure: at each turn, candidate responses are sampled and accepted only if they satisfy a target faithfulness condition, such as consistency, determinacy, or both. This design allows us to test the causal utility of faithfulness as a selection signal without additional training or model modification.

Our results reveal a nuanced relationship between CoT faithfulness and multi-turn agent performance. First, faithfulness can be a powerful inference-time signal: enforcing consistency or determinacy often yields substantial improvements

over base sampling, unlocking non-trivial performance even without updating model parameters. Second, the benefit of faithfulness is bounded by model capability and task difficulty. When the model is too weak or the task is too difficult, faithful reasoning cannot be reliably produced, and resampling provides little or no gain. Third, we observe a faithfulness paradox: jointly enforcing consistency and determinacy does not always compound their individual benefits, suggesting that different dimensions of faithfulness may impose partially competing constraints on generation.

Overall, our work makes the following contributions: (1) We present the first systematic investigation of CoT faithfulness in multi-turn LLM agents, shifting the focus from final-answer faithfulness to action-level faithfulness. (2) We decompose agent CoT faithfulness into two interpretable and verifiable dimensions: consistency, which captures CoT-action alignment, and determinacy, which captures whether the CoT uniquely determines the action. (3) We show that faithfulness-guided inference-time resampling can improve multi-turn agent performance without additional training, while also identifying its boundaries and a paradoxical subadditive interaction between faithfulness dimensions.

2 Related Work

2.1 CoT Faithfulness

Chain-of-thought (CoT) prompting is widely used to elicit intermediate reasoning and improve performance on complex tasks (Nye et al., 2021; Wei et al., 2022a; Kojima et al., 2022; Wang et al., 2023). As CoT is increasingly treated as an explanation, supervision signal, or monitoring interface (Cobbe et al., 2021; Lightman et al., 2023; Korbak et al., 2025), its faithfulness becomes crucial: a rationale should not merely sound plausible, but should reflect or constrain the computation leading to the final output. Prior work measures CoT faithfulness by intervening on reasoning traces, corrupting, truncating, or replacing intermediate steps, and testing whether final answers change accordingly (Lanham et al., 2023; Lyu et al., 2023; Paul et al., 2024; Tutek et al., 2025). Some work further questions whether existing tests capture inner faithfulness rather than output-level self-consistency (Parcalabescu and Frank, 2024). Other studies show that models may generate post-hoc rationalizations, omit influential cues, or encode decision-relevant information in hard-to-monitor ways (Turpin et al., 2023b; Chen et al., 2025; Roger and Greenblatt, 2023; Skaf et al., 2025).

However, most existing work studies static, single-turn reasoning, where the model produces a final answer after a rationale. *We instead examine faithfulness in a multi-turn action setting, where the rationale should constrain the agent’s next move and affect the subsequent trajectory. We therefore study faithfulness at the level of action selection rather than final-answer generation.*

2.2 Multi-turn LLM Agents

LLM agents extend language models from single-turn prediction to interactive decision making, where models repeatedly observe, reason, act, and receive feedback. Methods such as ReAct and Reflexion demonstrate the value of interleaving reasoning with actions or feedback (Yao et al., 2023; Shinn et al., 2023), while recent benchmarks evaluate agents in broader interactive settings such as web navigation, embodied tasks, tool use, and multi-step problem solving (Liu et al., 2024; Zhou et al., 2024; Deng et al., 2023; Shridhar et al., 2021). These benchmarks typically emphasize overall task success or long-horizon completion, making it difficult to isolate whether a model’s stated reasoning faithfully supports each individual action.

Clembench evaluates language models as conversational agents through rule-governed games such as Wordle, Taboo, reference games, and map navigation (Chalamalasetti et al., 2023). Playpen further extends such games from evaluation to learning by using dialogue-game feedback for post-training (Horst et al., 2025). These environments expose challenges such as sparse rewards, credit assignment, and state tracking. *Wordle-style games are especially suitable for our study because symbolic feedback imposes verifiable constraints on future guesses, allowing us to test whether the agent’s stated reasoning faithfully supports its next action.*

3 Action-Level CoT Faithfulness

3.1 Turn-Level Formulation

We study CoT faithfulness in a multi-turn agent setting, where reasoning is immediately followed by an action. At turn t , the agent observes the interaction history h_t , generates a rationale r_t , and then emits an action a_t . The environment subsequently returns feedback o_{t+1} , which updates the next interaction state. Unlike final-answer faithfulness, which evaluates whether a rationale supports an eventual answer, action-level faithfulness evaluates whether r_t faithfully supports the concrete action a_t selected at the same turn.

This distinction is important in interactive agents because each action changes the future trajectory. A rationale may appear plausible or partially relevant to the overall task, yet still fail to justify the specific action taken at the current decision point. We therefore define action-level CoT faithfulness as a property of the rationale-action pair (r_t, a_t) conditioned on the current history h_t .

3.2 Consistency and Determinacy

As shown in Fig. 1, we characterize action-level CoT faithfulness along two distinct but complementary dimensions.

Consistency. Consistency measures whether the selected action follows the constraints or claims stated in the rationale. In Wordle, for example, if the rationale states that a letter is absent from the target word, then a consistent guess should not contain that letter. Consistency therefore captures local CoT-action alignment: the action should not contradict the reasoning that precedes it.

Determinacy. Determinacy measures whether the rationale is sufficiently specific to constrain

the selected action. A rationale such as “the word contains E” may be consistent with many possible guesses and therefore provides little support for why one particular action is chosen. In contrast, a rationale that specifies positional constraints, included letters, and excluded letters may narrow the candidate space enough to determine or strongly justify the selected action. Determinacy therefore distinguishes rationales that are merely compatible with an action from those that meaningfully constrain action selection.

3.3 Faithfulness-Guided Resampling

We use inference-time resampling to test whether action-level faithfulness can serve as a selection signal. For each turn, the model samples a candidate rationale-action pair (r_t, a_t) . If the pair satisfies the target faithfulness condition, it is accepted; otherwise, the model resamples up to a per-turn budget. If no candidate satisfies the condition within the budget, we fall back to the sampled response.

We compare four decoding conditions: Base uses unconstrained sampling; Consistency accepts only candidates whose actions align with their rationales; Determinacy accepts only candidates whose rationales sufficiently constrain their actions; Both accepts only candidates satisfying both dimensions simultaneously. Faithfulness verification is performed by Qwen3-32B as the judge model; the full evaluation prompt is provided in App. A.

3.4 Wordle Interaction Settings

We use the Wordle environment from the Playpen (Horst et al., 2025) benchmark as our primary experimental testbed. In this multi-turn framework, an agent must identify a hidden five-letter target word within a limited number of sequential attempts. After each guess, the game master provides letter-level feedback specifying positional constraints: **green** denotes a letter in the correct position, **yellow** indicates a valid letter in an incorrect position, and **red** indicates that the letter does not appear in the target word. The agent follows a structured output format that requires an explicit rationale before each guess, naturally instantiating the CoT-then-action paradigm.

We evaluate three Wordle-based interaction settings with different feedback structures, illustrated in the right panels of Fig. 1. The base setting, Wordle, is the standard multi-turn game, where the agent updates its belief state using only symbolic letter-level feedback. Wordle-with-clue provides

an initial semantic clue from the game manager, reducing uncertainty by introducing information beyond symbolic feedback. Wordle-with-critic adds an auxiliary critic response that provides agreement or disagreement together with a qualitative rationale. Following ClemBench (Chalamalasetti et al., 2023), the critic is instantiated using the same underlying model as the playing agent, so this setting provides additional self-generated feedback rather than external expert supervision.

3.5 Experimental Setup

We evaluate Qwen2.5-Instruct models at five scales: 1.5B, 3B, 7B, 14B, and 32B. All experiments are conducted at three decoding temperatures, $T \in \{0.9, 1.2, 1.5\}$, to examine whether faithfulness-guided selection remains effective across different sampling regimes. For each configuration, we run three independent trials and report the mean ClemScore with standard error.

We use ClemScore as the primary task-performance metric. Following Chalamalasetti et al. (2023), ClemScore combines the proportion of episodes played to completion with the average quality score of completed episodes. Each episode is categorized as a *success*, *loss*, or *abort*. A success reaches the target within the allowed number of turns, whereas a loss completes normally without reaching the target. An abort occurs when the model fails to produce a valid game action after the permitted reprompting attempts.

Unless otherwise specified, faithfulness-guided resampling considers at most three responses at each decision point. This candidate resampling is distinct from the game-level reprompting used to recover from invalid actions. Retry counts reported in later analyses are aggregated over all turns and games within each configuration.

4 Analysis

4.1 When Faithfulness Helps

We first examine whether enforcing CoT faithfulness at inference time improves agent performance. Figure 2 shows the results across Qwen family models on both Wordle variants. Enforcing faithfulness conditions consistently yields substantial gains over the base sampling strategy across nearly all model scales. On Wordle-with-clue, the BOTH condition achieves a ClemScore of approximately 68 at 14B scale, compared to roughly 48 for the base baseline—a gain of nearly 20 points. Even on

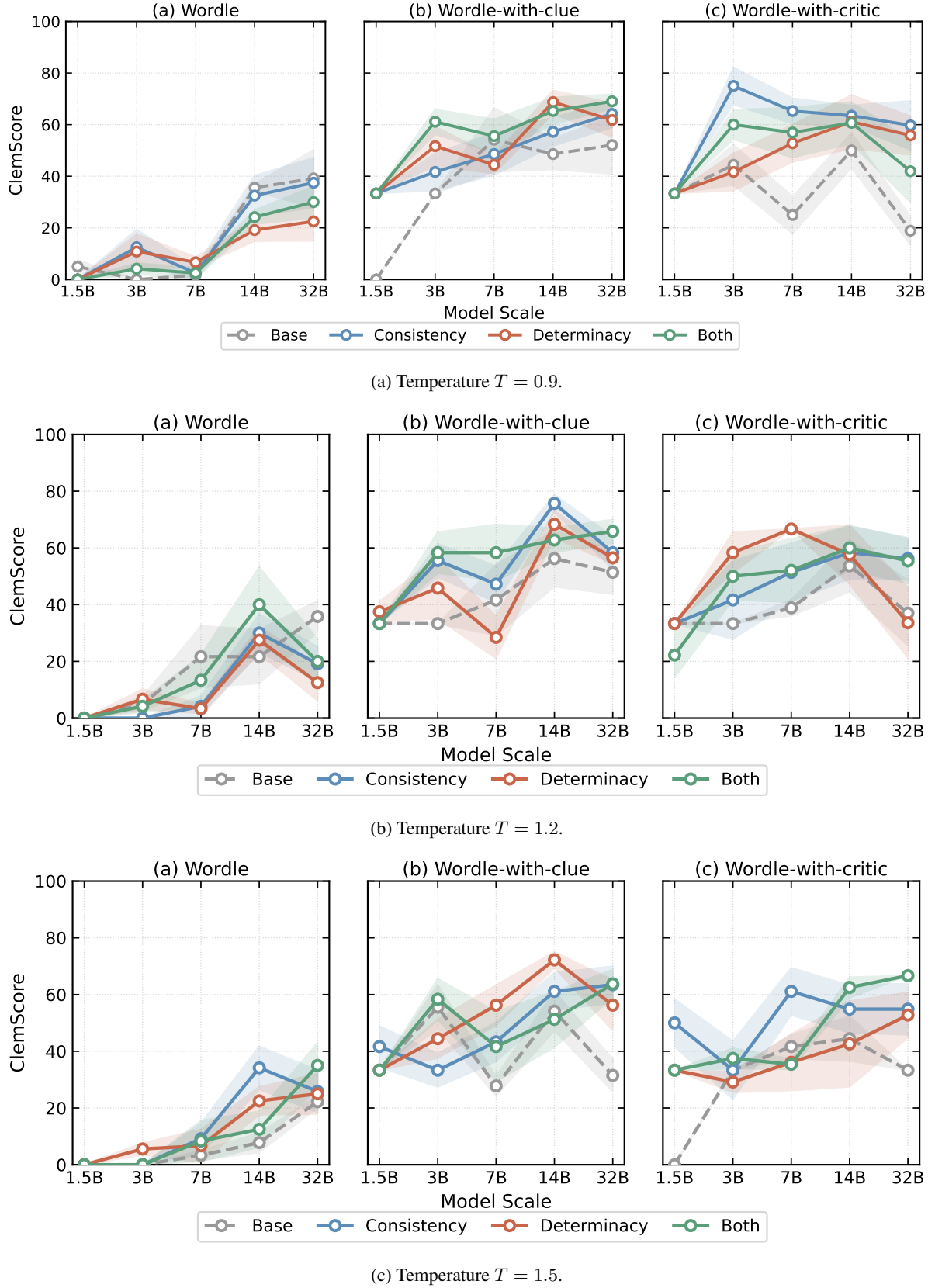


Figure 2: ClemScore performance under different faithfulness conditions across three Wordle-based interaction settings and decoding temperatures. Rows correspond to temperatures $T \in \{0.9, 1.2, 1.5\}$, while columns correspond to Wordle, Wordle-with-clue, and Wordle-with-critic. Different curves denote base sampling, consistency-based resampling, determinacy-based resampling, and jointly enforcing both dimensions of CoT faithfulness. Shaded regions denote standard errors over multiple runs.

the more challenging Wordle-withcritic, almost all three faithfulness conditions outperform the base strategy. Notably, this improvement requires no additional training, suggesting that faithfulness-guided resampling can unlock latent reasoning capacity already present in the model.

Takeaway 1

CoT faithfulness can be leveraged as a training-free inference-time signal: simple rejection sampling based on faithfulness conditions yields consistent and substantial performance gains across model scales, without any additional training, while incurring extra inference-time computation.

Takeaway 2

Faithfulness-guided resampling requires a minimum level of model competence to be effective. When the model is too weak or the task too difficult, repeated sampling may fail to produce more faithful CoTs, limiting the performance benefit of the resampling procedure.

4.2 When Faithfulness Fails

While faithfulness-guided resampling yields consistent gains on easier tasks, its effectiveness diminishes as task difficulty increases. We identify two regimes where faithfulness fails to provide meaningful improvement.

Small models on simple tasks. Even on the easiest variant, Wordle-withclue, faithfulness conditions fail to improve the weakest model. At 1.5B scale, the base ClemScore is 33.33, and applying Consistency or Determinacy constraints yields no meaningful gain (27.08 and 33.33, respectively), suggesting that the model lacks sufficient reasoning capacity to produce faithful CoT in the first place.

All models on the hardest task. On the hardest variant, Wordle, faithfulness conditions consistently fail to help across multiple model scales. At 14B, base performance is 23.34, with Consistency and Determinacy yielding 22.5 and 23.34; at 32B, base performance is 30.0, with Consistency dropping to 15.84 and Determinacy to 22.5. At 7B, all conditions cluster near chance level (12.5, 14.17, and 10.0). This pattern generalizes beyond the Qwen2.5 family: Qwen3-32B exhibits similar degradation, with base at 16.5 and Consistency dropping to 8.34, suggesting that task complexity, rather than model family, is the primary limiting factor.

4.3 The Faithfulness Paradox

While Consistency and Determinacy can individually improve performance, their effects are not always compositional. To quantify this interaction, we define an interaction score:

$$I = S_{\text{both}} - (S_{\text{consistency}} + S_{\text{determinacy}} - S_{\text{base}}),$$

where $I > 0$ indicates synergy and $I < 0$ indicates overlapping or interfering gains. A derivation of this interaction term is provided in App. B. A negative value indicates that jointly enforcing both faithfulness dimensions underperforms the additive expectation from enforcing them separately.

Across all valid model-task-temperature settings, the interaction score is negative in 22 out of 39 cases, with a mean of -5.57 and a median of -4.86 ClemScore points. This shows that the interaction between Consistency and Determinacy is frequently sub-additive rather than simply cumulative. For example, under $T = 0.9$ on Wordle-with-critic with Qwen-32B, Consistency and Determinacy individually reach 59.72 and 55.84 ClemScore, compared with a base score of 18.89. The additive expectation is therefore 96.67, yet jointly enforcing both dimensions achieves only 41.94, yielding an interaction score of $I = -54.73$.

This pattern suggests that Consistency and Determinacy impose partially competing demands on generation. Satisfying both constraints simultaneously can shrink the pool of acceptable candidates, making the model more likely to select a lower-quality rationale-action pair or fall back after exhausting the resampling budget. Thus, action-level faithfulness is not a single monotonic objective: different dimensions can be individually useful while interacting negatively when enforced together.



Figure 3: Pooled Spearman correlations between explanation length and ClemScore. Each cell pools all available temperature settings and decoding conditions for the corresponding model-task pair. The mixed signs and weak correlations indicate that longer explanations do not consistently account for faithfulness gains.

Takeaway 3

Jointly constraining Consistency and Determinacy does not compound their individual benefits. The interaction between the two faithfulness dimensions is subadditive, suggesting that they capture partially competing aspects of CoT quality.

5 Further Analysis

5.1 Disentangling Faithfulness Gains from Explanation Length

A natural alternative explanation is that faithfulness-guided resampling improves performance by selecting longer explanations, thereby providing more reasoning tokens and test-time computation. To examine this possibility, we compute Spearman correlations between average explanation length and ClemScore.

At the aggregate level, explanation length is only weakly associated with performance across all conditions ($\rho = 0.114$), and this association nearly vanishes when considering only faithfulness-guided conditions ($\rho = 0.010$). Fig. 3 further shows that this relationship is not consistent across model scales or tasks. The correlations vary in both sign and magnitude: for example, longer explanations correlate negatively with ClemScore for the 1.5B model on Wordle and With-Critic, but positively on With-Clue, while most correlations for the 3B, 7B, and 14B models remain close to zero. Although several isolated positive correlations appear, such as for the 32B model on With-Critic, they do

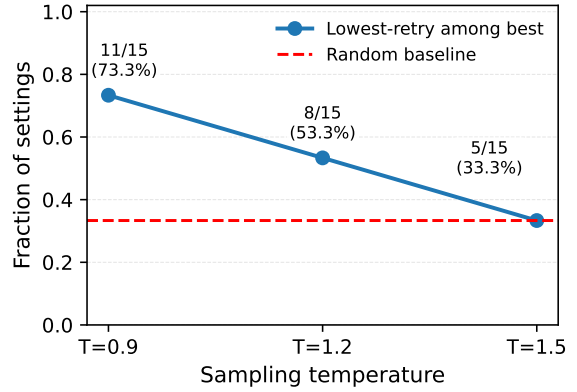


Figure 4: Retry-count analysis across temperatures. The fraction of settings where the lowest-retry condition is among the best-performing conditions decreases from $T = 0.9$ to $T = 1.5$, approaching the random baseline. This suggests that retry count shifts from a signal of native faithfulness at low temperature to a less deterministic indicator under broader exploratory sampling.

not form a systematic pattern across settings.

These results suggest that the gains from action-level faithfulness cannot be explained simply by increased explanation length. Rather than merely favoring more verbose rationales, faithfulness-guided resampling appears to select reasoning traces that better support the subsequent action.

5.2 Retry Count Reveals the Availability of Faithful Trajectories

Retry count measures the number of additional candidates sampled before finding a response that satisfies the faithfulness constraint. Since our method accepts the first qualifying candidate, fewer retries indicate that faithful reasoning-action pairs are more readily available under the current sampling distribution, whereas more retries suggest that satisfying the constraint requires broader search.

We first investigate whether performance gains are merely a consequence of more aggressive rejection sampling. Across all model, task, and temperature settings, the lowest-retry faithfulness condition is among the best-performing conditions in 24 out of 45 cases, exceeding the 1/3 random baseline over the three faithfulness-guided conditions. This suggests that improvements do not simply arise from allocating more sampling attempts, but from selecting higher-quality trajectories that naturally satisfy the faithfulness constraints.

However, the diagnostic value of retry count depends on the sampling regime. As shown in Fig. 4, the lowest-retry condition is among the best-

Table 1: Performance change when increasing the resampling budget from 3 to 5 attempts. We report the average ClemScore improvement across model-task settings.

Temp.	Consist.	Determin.	Both
0.9	-0.12	+8.21	-2.50
1.2	-0.36	+3.67	+0.37
1.5	+0.11	+1.96	+0.84
Average	-0.12	+4.61	-0.43

performing conditions in 11/15 settings at $T = 0.9$, 8/15 settings at $T = 1.2$, and 5/15 settings at $T = 1.5$. At low temperatures, the sampling distribution is more concentrated, making retry count a stronger indicator of whether faithful and task-useful trajectories are naturally available. In contrast, higher temperatures produce a broader candidate distribution, where retry count increasingly reflects exploration rather than trajectory quality; consequently, its correlation with performance approaches randomness.

These results reveal that retry count should not be interpreted as a universal measure of sampling efficiency. Instead, it serves as a temperature-dependent indicator of the underlying trajectory distribution: under concentrated sampling, fewer retries imply readily available faithful trajectories, while under diverse sampling, additional retries may simply facilitate exploration among a wider range of candidates. These findings suggest that faithfulness-guided inference should adapt its sampling strategy to the decoding regime: conservative sampling with stricter constraints is preferable when the model distribution is concentrated, whereas broader exploration with larger budgets may be beneficial under high-temperature decoding.

5.3 The Effect of Resampling Budget

Faithfulness-guided resampling introduces additional inference-time computation. While retry count characterizes how easily faithful trajectories can be found under a fixed budget, it remains unclear whether increasing the allowed sampling budget consistently improves performance. We therefore compare the default budget of three attempts with an extended budget of five attempts while keeping the model, task, temperature, and verifier unchanged.

As shown in Tab. 1, increasing the resam-

pling budget provides only marginal improvements. Across all evaluated settings, expanding the budget from 3 to 5 attempts improves ClemScore by only 1.35 points on average, with improvements observed in 51.9% of settings. Moreover, the benefit does not monotonically increase with temperature: the average gain decreases from 1.86 at $T = 0.9$ to 0.97 at $T = 1.5$.

Interestingly, the effect of additional budget depends on the faithfulness dimension. Determinacy benefits most from additional sampling budget, achieving an average improvement of 4.61 ClemScore points, while Consistency and jointly enforcing Both show limited additional gains. This suggests that constraints requiring more specific action determination may require broader search to discover suitable trajectories, whereas consistency-aligned trajectories are often already accessible within a smaller budget.

Together with the retry analysis, these results indicate that more sampling is not universally beneficial. Additional inference budget helps primarily when useful faithful trajectories exist but have relatively low sampling probability; otherwise, increasing the budget may only introduce additional search cost without reliable performance gains.

6 Conclusion

We studied CoT faithfulness in multi-turn LLM agents by shifting the focus from final-answer explanations to action-level decision making. We introduced two complementary dimensions of faithfulness: consistency, which measures CoT-action alignment, and determinacy, which measures whether reasoning constrains the selected action. Across Wordle-based dialogue environments, we showed that faithfulness-guided inference-time resampling can improve agent performance without additional training. However, these gains are conditional: faithfulness is most effective when models can already generate useful reasoning-action trajectories, while jointly enforcing multiple constraints may introduce trade-offs rather than additive benefits. Our findings highlight that CoT faithfulness should be viewed as an action-level signal whose utility depends on the model, task, and interaction setting.

Limitations

Controlled interaction setting. Our experiments are conducted on Wordle-style dialogue games,

which provide a controlled environment where actions, feedback, and constraints can be explicitly tracked. This makes the setting suitable for isolating action-level CoT faithfulness, but it is still simpler than open-ended agents. Future work should examine whether the observed effectiveness, boundaries, and paradoxes of faithfulness dimensions generalize to broader agentic tasks.

Operationalization of faithfulness. We operationalize action-level faithfulness through consistency and determinacy. While these dimensions capture important relationships between rationales and actions, they do not exhaust all possible notions of faithful reasoning. For example, a rationale may be consistent and determinate while still relying on shallow heuristics or incomplete internal reasoning. Thus, our measures should be viewed as task-grounded proxies for action-level faithfulness rather than complete characterizations of model cognition.

References

- Kranti Chalamalasetti, Jana Götze, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2023. clembench: Using game play to evaluate chat-optimized language models as conversational agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11174–11219.
- Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, Arushi Somani, Peter Hase, Misha Wagner, Fabien Roger, Vlad Mikulik, Samuel R. Bowman, Jan Leike, Jared Kaplan, and Ethan Perez. 2025. Reasoning models don’t always say what they think. *arXiv preprint arXiv:2505.05410*.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.
- Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Sam Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the web. In *Advances in Neural Information Processing Systems*.
- Nicola Horst, Davide Mazzaccara, Antonia Schmidt, Michael Sullivan, Filippo Momentè, Luca Franceschetti, Philipp Sadler, Sherzod Hakimov, Alberto Testoni, Raffaella Bernardi, Raquel Fernández, Alexander Koller, Oliver Lemon, David Schlangen, Mario Giulianelli, and Alessandro Suglia. 2025. Playpen: An environment for exploring learning from dialogue game feedback. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems*.
- Tomek Korbak, Mikita Balesni, Elizabeth Barnes, Yoshua Bengio, Joe Benton, Joseph Bloom, Mark Chen, Alan Cooney, Allan Dafoe, Anca Dragan, and 1 others. 2025. Chain of thought monitorability: A new and fragile opportunity for ai safety. *arXiv preprint arXiv:2507.11473*.
- Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, Kamilè Lukošiušė, Karina Nguyen, Newton Cheng, Nicholas Joseph, Nicholas Schiefer, Oliver Rausch, Robin Larson, Sam McCandlish, Sandipan Kundu, and 11 others. 2023. [Measuring faithfulness in chain-of-thought reasoning](#). *Preprint, arXiv:2307.13702*.
- Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let’s verify step by step. *arXiv preprint arXiv:2305.20050*.
- Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, and 3 others. 2024. Agentbench: Evaluating llms as agents. In *International Conference on Learning Representations*.
- Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 305–329.
- Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, Charles Sutton, and Augustus Odena. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.
- Letitia Parcalabescu and Anette Frank. 2024. On measuring faithfulness or self-consistency of natural language explanations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

- Debjit Paul, Robert West, Antoine Bosselut, and Boi Faltings. 2024. Making reasoning matter: Measuring and improving faithfulness of chain-of-thought reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15012–15032.
- Fabien Roger and Ryan Greenblatt. 2023. Preventing language models from hiding their reasoning. *arXiv preprint arXiv:2310.18512*.
- Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik R. Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*.
- Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. Alfworld: Aligning text and embodied environments for interactive learning. In *International Conference on Learning Representations*.
- Joey Skaf, Luis Ibanez-Lissen, Robert McCarthy, Connor Watts, Vasil Georgiev, Hannes Whittingham, Lorena Gonzalez-Manzano, David Lindner, Cameron Tice, Edward James Young, and Puria Radmard. 2025. Large language models can learn and generalize steganographic chain-of-thought under process supervision. *arXiv preprint arXiv:2506.01926*.
- Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023a. Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965.
- Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023b. Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems*.
- Martin Tutek, Fateme Hashemi Chaleshtori, Ana Marasović, and Yonatan Belinkov. 2025. Measuring chain of thought faithfulness by unlearning reasoning steps. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.
- Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, and 1 others. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V. Le, and Denny Zhou. 2022a. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V. Le, Denny Zhou, and 1 others. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.
- Qwen An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxin Yang, Jingren Zhou, Junyang Lin, and 25 others. 2024. [Qwen2.5 technical report](#). *ArXiv*, abs/2412.15115.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.
- Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*.

A Judge Prompts

We use a verifier model to evaluate whether a candidate response satisfies the required faithfulness condition. The verifier receives the current game context and the candidate response, including both the explanation and the guess. It then judges two criteria: *consistency*, which checks whether the explanation and guess are logically aligned with each other and with the game constraints, and *determinacy*, which checks whether the explanation sufficiently narrows the candidate space toward the selected guess.

Verifier System Prompt

You are a strict verifier.

Given game context and a candidate response, judge two criteria:

- 1) **consistency**: whether explanation and guess are logically consistent with each other and context constraints.
- 2) **determinacy**: whether explanation constraints substantially narrow to this guess, not too many alternatives.

Return JSON only with keys:

```
{"is_consistent": bool, "is_determined": bool, "passed": bool, "reason": str}
```

where passed = is_consistent and is_determined.

B Derivation of the Interaction Term

Let the gains over base sampling be

$$\Delta_{\text{cons}} = S_{\text{consistency}} - S_{\text{base}},$$

$$\Delta_{\text{det}} = S_{\text{determinacy}} - S_{\text{base}}.$$

Under an additive no-interaction assumption, the expected joint score is

$$\begin{aligned} S_{\text{exp}} &= S_{\text{base}} + \Delta_{\text{cons}} + \Delta_{\text{det}} \\ &= S_{\text{consistency}} + S_{\text{determinacy}} - S_{\text{base}}. \end{aligned}$$

The interaction term is therefore

$$\begin{aligned} I &= S_{\text{both}} - S_{\text{exp}} \\ &= S_{\text{both}} - S_{\text{consistency}} - S_{\text{determinacy}} + S_{\text{base}}. \end{aligned}$$

A positive value indicates synergy beyond additive gains, while a negative value indicates overlapping benefits or interference.

ProcessPrefBench: Evaluating Cross-Task Transfer of Individual Process Preferences from Naturalistic Human–AI Interaction

Yiwen Yang

Independent Researcher
getupyang@gmail.com

Guanming Liu

Fudan University
gmliu24@m.fudan.edu.cn

Abstract

In natural human–AI dialogue, users’ corrections, requests, and self-descriptions carry reusable signals about how they want work to be done. These signals reveal *individual process preferences*, user-specific standards for evidence, depth, analytical focus, procedure, and delivery. We ask whether a preference revealed in one dialogue can shape an assistant’s later answer when the task does not restate it. ProcessPrefBench uses three months of natural dialogues from two knowledge workers to build 66 source–target cases separately judged for source evidence and target applicability. Sources are preselected, so the benchmark tests application rather than retrieval. We test three models with target-only, full-history, and distilled-preference prompts. Target-specific 0–2 rubrics score preference realization, and a task gate flags severe failures. Compared with target-only prompting, full-history and imperative distilled-preference prompting raise preference realization by 0.204 and 0.244 points on the 0–2 scale, respectively, with no reliable advantage from distillation. Fabricated-evidence failures rise from 5.8% under target-only prompting to 10.1% under full-history and 10.4% under imperative prompting. ProcessPrefBench highlights a tension in learning from dialogue, where drawing on prior interactions can improve preference realization yet also invite fabricated evidence. We hope it helps build assistants that honor a user’s process preferences without sacrificing factual reliability.

1 Introduction

Users often do not state their process preferences in advance. In human–AI dialogue, these preferences become visible when users react to a particular answer through corrections, follow-up requests, or self-descriptions. Remembering facts about a user does not by itself teach an assistant how that user wants work to be carried out. Figure 1 follows

an April correction that redirects product analysis from market-share breakdowns toward who uses a product and how. Six weeks later, the same user asks about a different product without restating that standard. We define individual process preferences as reusable, user-specific standards for how work should be done. They may apply beyond the dialogue in which they are expressed and govern what evidence supports a claim, how an analysis is organized, how deeply a source is explained, and whether a requested artifact is delivered without further confirmation.

Learning process preferences from dialogue is a delayed, cross-task form of adaptation. An end-to-end system would have to infer the reusable standard from the dialogue that carries it, determine when it applies, and realize it without treating remembered content as evidence for the new task. These stages can fail in different ways. The system may preserve the wrong standard, fail to realize a correctly preserved one, or import remembered content as unsupported target claims. ProcessPrefBench studies a narrower setting in which the relevant source dialogue is preselected for each target. It tests whether models can realize the preference when given the full dialogue or a distilled representation without importing unsupported source content.

Existing benchmarks approach learning from dialogue in two ways: some test whether conversational information remains available for later reasoning (Maharana et al., 2024; Wu et al., 2025), while others hand the model a profile or preference record directly (Jiang et al., 2025; Zhao et al., 2025). These settings do not jointly test whether a preference can be induced from a natural dialogue and then applied to a different task without importing unsupported content.

We introduce ProcessPrefBench using three months of natural user–assistant dialogues from Margin, a reading, annotation, and AI-discussion

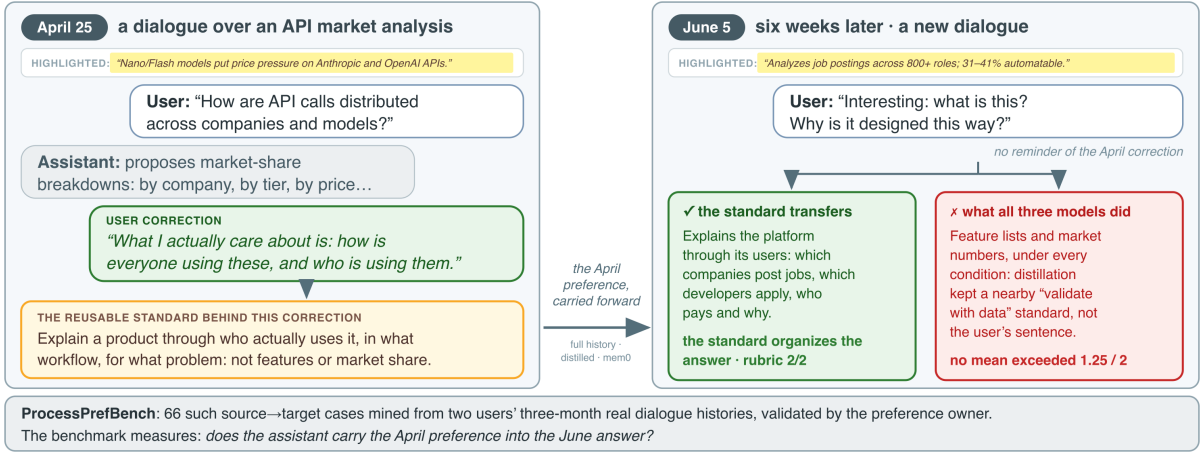


Figure 1: A ProcessPrefBench case (translated from Chinese). An April dialogue establishes a reusable analysis standard; six weeks later, the same user asks about a different product in a new dialogue without restating it. Appendix D audits this case end to end.

product used primarily by knowledge workers. The benchmark is an idiographic diagnostic instrument focused on research, sensemaking, and decision tasks from two users. Each case links a preference-bearing source to a different task from the same user; construction separately establishes that the source supports a reusable standard and that the standard should affect a target that does not restate it.

ProcessPrefBench contributes a source-to-target construction protocol, target-specific rubrics that distinguish surface mention from functional integration, and an evaluation that separates preference realization from severe task failure. The relevant source is preselected for every case, so the experiment tests application of a known-relevant dialogue rather than retrieval from a heterogeneous history. Across three models, relevant history improves preference realization, but distillation offers no reliable advantage over full history and both memory conditions increase fabricated-evidence failures.

2 Related Work

Long-term-memory benchmarks test whether assistants retain and reason over information distributed across extended interactions. LoCoMo and Long-MemEval cover extraction, temporal reasoning, and knowledge updates across extended conversations, while MemoryAgentBench uses incremental multi-turn interactions (Maharana et al., 2024; Wu et al., 2025; Hu et al., 2025). These benchmarks establish whether information remains available over time; ProcessPrefBench asks whether a dialogue

also supports a reusable working standard.

Personalization benchmarks typically provide the user signal more directly. PersonaMem studies dynamic user profiling, LaMP evaluates personalized language-model tasks, and PrefEval tests preference recognition and following (Jiang et al., 2025; Salemi et al., 2024; Zhao et al., 2025). CUPID infers which contextual preference applies to a new request in constructed interactions (Kim et al., 2025), while AlpsBench builds personalization tests from natural dialogues (Xiao et al., 2026). ProcessPrefBench instead derives each standard from a natural correction, prescription, or self-description and establishes its applicability to a later target.

Learning preferences from interaction is a closer setting. PRELUDE learns latent preferences from user edits (Gao et al., 2024), and TRACE compiles corrections into runtime enforcement rules (Zhou et al., 2026); our focus is the evaluation problem of tracing a source-supported standard into a different task and distinguishing functional realization from surface mention. Game-based evaluation measures interactive ability through situated behavior rather than similarity to reference text (Chalamalasetti et al., 2023); our rubrics apply the same principle to naturalistic collaboration by scoring functional integration instead of surface mention.

External memory systems such as MemoryBank, MemoryOS, and mem0 store and reintroduce information outside the base model (Zhong et al., 2024; Kang et al., 2025; Chhikara et al., 2025); HaluMem studies hallucinations introduced by memory operations (Chen et al., 2025). A retrieved interaction

can mix a reusable process standard with earlier-task content, so retrieval alone neither ensures transfer nor prevents that content from entering target-task claims.

3 Benchmark Construction

Data source. We collect three months of natural interactions from two knowledge workers using Margin, a webpage reading, annotation, and AI-discussion product. Each trace retains the available reading context, user request, assistant response, and subsequent user turns. Assistant responses came from locally installed Claude Code and Codex at their then-default models; early traces lack per-turn version logs, so the source model is not always identifiable. For 13 cases with truncated reading context, we restore verified source material so that the evaluation does not force a choice between fabrication and refusal.

Human source annotation. A source is retained when the user’s words support a reusable standard for future work, expressed through correction, prescription, or self-description. Each user annotates their own interactions and excludes factual repairs, current-task-only requests, and scene-bound conclusions because they do not justify transfer beyond the source task. Each retained case stores the inferred standard, its supporting words, and the full interaction as auditable evidence.

Human target pairing. Source annotation does not determine a valid target. For each source, the preference owner separately judges another same-user task. A pairing is retained when the target does not repeat the preference but applying the source-supported standard should improve the answer. Review also records order, cue strength, material and tool availability, and evaluability. This procedure rejects topical similarity alone and prevents source validity from making target applicability automatic.

Rubric construction. Each case contains one to three source-anchored *preference atoms* with target-specific criteria for not applicable, 0, 1, or 2. A criterion is not applicable when its trigger is absent from the target. A score of 0 marks an absent or contradicted preference, 1 marks local or cosmetic realization, and 2 requires a functional change to the answer’s reasoning, organization, conclusion, or delivery. This progression measures whether the remembered standard changes the answer rather than whether the answer merely

mentions it. One AI research assistant drafted the target-specific criteria under a frozen specification, followed by owner spot checks. Appendix D gives two end-to-end case audits.

A separate task gate marks severe omission, major factual error, fabricated evidence, hard-constraint violation, or a false claim of external execution. Task failure does not overwrite the preference score: keeping the two judgments separate prevents a personalized but unusable answer from counting as success and prevents lack of personalization alone from becoming a general task failure. Among these failure types, fabricated evidence carries our second headline result. The gate assigns this label when an answer presents specific claims about papers, people, data, citations, or links as established evidence that the target-side materials do not support, or falsely claims an external action. The label is operational. Because the gate does not see the source, a carried-over detail and an invented one receive the same label when neither has target-side support.

Measurement audit. The benchmark changed in response to three measurement checks. An ablation supported removing descriptive reference answers that did not clarify the target-specific rubric. A controlled comparison exposed wording that favored injected preferences and led us to rewrite the affected criteria. A material audit identified truncated reading context that could induce fabrication, motivating the verified restoration described above.

4 Evaluation

Conditions. All conditions share the system prompt, page title, highlighted context, verified supplied material when needed, and verbatim user question. Target-only prompting adds no memory, while full-history prompting adds the complete source interaction. For both distilled-preference conditions, the tested model distills each source once per source–model pair and reuses the result. Imperative prompting always applies it, whereas conditional prompting applies it only when relevant. The mem0 external-memory system extracts from the source and retrieves against the target. Conditional prompting is our variant rather than a description of mem0 semantics.

Models and generation. We test DeepSeek V4 Flash, Qwen-Max, and GPT-4o. Every case is generated twice for each model and memory condition.

The final run contains 1980 answers.

Judging. Claude Sonnet 4.6 serves in isolated preference and task-gate roles; its model family is disjoint from the tested models, which limits self-preference bias (Zheng et al., 2023). Both roles see the target, materials, and answer, but not the tested model, condition, source interaction, or alternatives. The preference judge also sees the rubric and casts three evidence-citing votes per atom. A majority of not-applicable votes excludes the atom; otherwise we use the median applicable score. The task gate sees validity checks without the preference rubric and uses two votes plus a third on ties.

Judge validation. We audited the judge on 30 answers, oversampling judge-flagged ones; two annotators rated every answer blind to the tested model, condition, and judge scores. Post-calibration human agreement is substantial ($\kappa = .72$ on preference atoms, $.84$ on the task gate), while agreement with the judge is lower and the judge scores preferences more conservatively. The audit is diagnostic rather than confirmatory, so the main effects remain estimates under the benchmark judge; Appendix B reports the protocol, full agreement table, and per-annotator ratings.

Aggregation. An answer score averages its applicable atoms; model-condition means exclude answers with no applicable atoms, and task failure is reported independently. Primary comparisons contrast imperative distilled-preference prompting with target-only and full-history prompting: we macro-average paired differences across models with exact sign tests and report case-level cluster-bootstrap 95% confidence intervals (10,000 resamples). Conditional distilled-preference prompting is descriptive because every formal case is an intended positive match.

5 Results

Preference transfer. Access to user-specific history improves preference scores over target-only prompting. Across models, imperative distilled-preference prompting exceeds target-only prompting by 0.244 points (95% CI [0.177, 0.313]; 48/8/9 wins/ties/losses; exact sign $p < .0001$), with positive effects for DeepSeek V4 Flash (+0.427), Qwen-Max (+0.114), and GPT-4o (+0.192). Full-history prompting gains 0.204 points over target-only prompting (95% CI [0.143, 0.269]; 44/12/9; $p < .0001$).

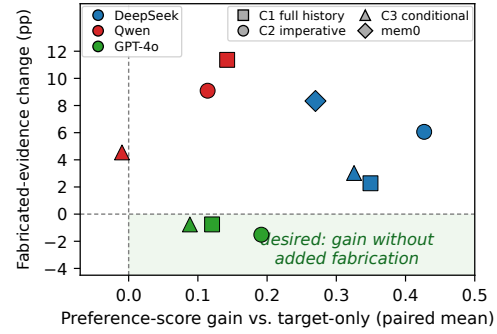


Figure 2: Preference-score and fabricated-evidence changes from target-only prompting across model-condition pairs.

Distillation does not yield a reliable preference-score advantage over full history. The difference is +0.026 (95% CI $[-0.034, +0.087]$; $p = .451$), with model-level effects of +0.080 for DeepSeek V4 Flash, -0.058 for Qwen-Max, and +0.071 for GPT-4o.¹ This does not establish equivalence; the representations’ relative preference effects remain unresolved. The mem0 cell means exceed target-only prompting for DeepSeek V4 Flash and GPT-4o but not for Qwen-Max.

Task validity and factual reliability. A preference-score gain does not guarantee answer validity (Figure 2). Imperative distilled-preference prompting raises the overall task-failure rate from 15.2% to 18.9% relative to target-only prompting, although the difference is not conventionally significant (exact McNemar $p = .086$). Fabricated-evidence failures rise from 5.8% under target-only prompting to 10.4% under imperative prompting, a difference of 4.5 percentage points (95% CI $[+2.0, +7.6]$; exact McNemar $p = .006$). Full-history prompting shows a similar increase, from 5.8% to 10.1% (difference 4.3 points; 95% CI $[+1.0, +7.8]$; $p = .010$), while the two memory conditions differ by only 0.3 points ($p = 1.0$). The increase is largest for Qwen-Max under imperative distilled-preference prompting (+9.1 points, $p = .004$), while GPT-4o shows no same-direction increase. The plotted mem0 comparison also shows an 8.3-point increase in fabrication ($p = .019$). Appendix E gives one

¹The gains of 0.244 and 0.204 are computed over the 65 cases with an applicable target-only score, so their difference is 0.040 on that common subset. The direct imperative-versus-full-history comparison includes one additional case for which all target-only answers had no applicable rubric atom; its case-level imperative-versus-full-history difference is -0.875 , reducing the 66-case mean to 0.026.

Condition	DeepSeek V4 Flash		Qwen-Max		GPT-4o	
	Preference score	Task failure rate	Preference score	Task failure rate	Preference score	Task failure rate
Target-only prompting	0.79	8.3%	0.62	13.6%	0.47	23.5%
Full-history prompting	1.15	9.1%	0.77	23.5%	0.59	14.4%
Imperative distilled-preference prompting	1.23	14.4%	0.73	22.7%	0.66	19.7%
Conditional distilled-preference prompting	1.11	8.3%	0.61	18.9%	0.58	23.5%
mem0 external-memory system	1.06	15.2%	0.59	21.3%	0.49	22.6%

Table 1: Preference score (higher is better) and task-failure rate (lower is better).

flagged example from each memory condition; in both, the answer supplies exactly the kind of specific evidence the transferred standard rewards, without support in the target-side materials.

Method-level interpretation. The interfaces leave different parts of the transfer problem unresolved: full history and imperative distillation both raise preference realization yet carry the same fabrication risk, conditional distillation suppresses intended positive matches, and mem0 yields no consistent gain. No evaluated interface combines reliable preference transfer with preserved factual reliability, and autonomous selection from long histories remains untested.

Robustness and scope. The main comparison between imperative distilled-preference prompting and target-only prompting remains positive separately for user1 and user2, after excluding the three temporally inverted cases, under source- and target-cluster aggregation, and when restricting the analysis to normally terminated generations (Appendix C). Corresponding comparisons with full-history prompting remain non-significant. These checks support history access within the two-user benchmark but do not resolve the relative value of distillation or establish population-level effects.

6 Conclusion

Natural human–AI dialogue carries evidence of how a user wants work done. ProcessPrefBench turns this evidence into 66 cross-task cases through separate judgments of source evidence and target applicability. Across three models, relevant history improves preference realization, yet the failures are separable: a system may preserve the wrong standard, fail to realize a correctly preserved one, or import remembered content as unsupported target claims. Reliable process-preference transfer therefore requires more than retrieving or summarizing a relevant interaction, and ProcessPrefBench makes

progress toward it measurable.

Limitations

ProcessPrefBench is an idiographic benchmark based on two knowledge workers using one product. Its imbalanced 52/14 case split and concentration in research, sensemaking, and decision tasks bound its scope. The 66 cases draw on 30 corrective sources and 59 distinct targets; thus, bootstrap intervals capture within-benchmark variation rather than population uncertainty, although source- and target-cluster analyses preserve the main direction. Because each case supplies the relevant source, the benchmark tests cross-task preference application, not retrieval from heterogeneous histories.

The human judge audit is diagnostic. It covers 30 answers from 10 target groups, includes an annotator who is both a preference owner and an author, and reports post-calibration agreement on the calibration items themselves. The enriched sample does not validate the between-condition fabrication contrast. Residual disagreements center on domain-knowledge-intensive answers, where the non-expert annotator scores systematically higher. Headline effects should therefore be read as effects under the benchmark judge.

Ethics and Privacy

The corpus contains natural interaction traces from a user who is also an author and an independent user. Candidate inclusion does not constitute release permission. Before releasing any material from these interactions, we confirm participant consent and review each item for privacy, provenance, and redistribution rights. Gated access does not replace de-identification.

Data and Code Availability

The public artifact at <https://github.com/getupyang/ProcessPrefBench> currently includes the benchmark definition; construction

and evaluation protocols; a public case schema; a synthetic demonstration case; a data card; privacy documentation; licensing and citation metadata; and a release validator. We are preparing the frozen prompts, evaluation and aggregation code, and de-identified per-answer audit records for release; each artifact must first be reviewed for private paths, provenance, and content. We will release natural interaction cases only after the ethics review above. We will not distribute raw histories, identity mappings, private URLs or paths, consent records, or internal answer keys.

References

- Kranti Chalamalasetti, Jana Götze, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2023. *clmbench: Using game play to evaluate chat-optimized language models as conversational agents*. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11174–11219.
- Ding Chen, Simin Niu, Kehang Li, Peng Liu, Xiangping Zheng, Bo Tang, Xinchu Li, Feiyu Xiong, and Zhiyu Li. 2025. *HaluMem: Evaluating hallucinations in memory systems of agents*. ArXiv:2511.03506.
- Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. *Mem0: Building production-ready ai agents with scalable long-term memory*. Preprint, arXiv:2504.19413.
- Ge Gao, Alexey Taymanov, Eduardo Salinas, Paul Mineiro, and Dipendra Misra. 2024. *Aligning LLM agents by learning latent preference from user edits*. In *Advances in Neural Information Processing Systems 37*.
- Yuanzhe Hu, Yu Wang, and Julian McAuley. 2025. *Evaluating memory in LLM agents via incremental multi-turn interactions*. Preprint, arXiv:2507.05257.
- Bowen Jiang, Zhuoqun Hao, Young-Min Cho, Bryan Li, Yuan Yuan, Sihao Chen, Lyle Ungar, Camillo J. Taylor, and Dan Roth. 2025. *Know me, respond to me: Benchmarking LLMs for dynamic user profiling and personalized responses at scale*. In *Proceedings of the Conference on Language Modeling (COLM)*.
- Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. 2025. *MemoryOS of ai agent*. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25972–25981.
- Tae Soo Kim, Yoonjoo Lee, Yoonah Park, Jiho Kim, Young-Ho Kim, and Juho Kim. 2025. *CUPID: Evaluating personalized and contextualized alignment of LLMs from interactions*. In *Proceedings of the Conference on Language Modeling (COLM)*. ArXiv:2508.01674.
- Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. *Evaluating very long-term conversational memory of LLM agents*. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870.
- Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. *LaMP: When large language models meet personalization*. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392.
- Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025. *LongMemEval: Benchmarking chat assistants on long-term interactive memory*. In *The Thirteenth International Conference on Learning Representations*.
- Jianfei Xiao, Xiang Yu, Chengbing Wang, Wuqiang Zheng, Xinyu Lin, Kaining Liu, Hongxun Ding, Yang Zhang, Wenjie Wang, Fuli Feng, and Xiangnan He. 2026. *AlpsBench: An LLM personalization benchmark for real-dialogue memorization and preference alignment*. ArXiv:2603.26680.
- Siyan Zhao, Mingyi Hong, Yang Liu, Devamanyu Hazarika, and Kaixiang Lin. 2025. *Do LLMs recognize your preferences? evaluating personalized preference following in LLMs*. In *The Thirteenth International Conference on Learning Representations*.
- Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. *Judging LLM-as-a-judge with MT-Bench and Chatbot Arena*. In *Advances in Neural Information Processing Systems 36 (NeurIPS), Datasets and Benchmarks Track*. ArXiv:2306.05685.
- Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. *MemoryBank: Enhancing large language models with long-term memory*. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731.
- Yujun Zhou, Kehan Guo, Haomin Zhuang, Xiangqi Wang, Yue Huang, Zhenwen Liang, Pin-Yu Chen, Tian Gao, Nuno Moniz, Nitesh V. Chawla, and Xiangliang Zhang. 2026. *Getting better at working with you: Compiling user corrections into runtime enforcement for coding agents*. ArXiv:2606.13174.

A Scoring Protocol

Preference score. The preference judge scores each target-specific atom independently. N/A means that the atom’s trigger is absent and is excluded rather than scored as zero. A score of 0 marks an absent or contradicted preference; 1 marks nominal, local, or cosmetic realization; and 2 requires the preference to change the answer’s reasoning, organization, conclusion, or delivery. Each vote must cite answer evidence. Three votes are collected per atom; a majority of N/A votes excludes the atom, and otherwise the median applicable score is used.

Task gate. The separate task gate flags only severe unusable errors: non-answer, major factual error, fabricated evidence, hard-constraint violation, or a false claim of external execution. Missing personalization alone is not a task failure. The gate uses two votes and a third on ties, without access to the preference rubric.

B Human Calibration of the Judges

The diagnostic sample contains 30 answers drawn with a fixed random seed from 10 target groups, stratified across tested models, memory conditions, and judge score bands; because the answers share target groups, they are not independent at the case level. It deliberately enriches judge-flagged answers, so its class proportions do not estimate benchmark prevalence. Human annotators saw the same target, material, rubric, and answer as the corresponding judge, but not the tested model, memory condition, or judge output. They assigned 0, 1, 2, or not applicable to each preference atom and a binary task-gate judgment with a failure type, first scoring independently, then discussing scale boundaries and rescored the same items. Initial preference agreement was linear-weighted $\kappa = .27$; Table 2 reports the post-calibration ratings, where the annotators agree on 29/30 task-gate judgments (97%) and reach 80% exact agreement on preference atoms (95% target-group cluster-bootstrap interval for κ : $[\cdot52, \cdot87]$; $n = 44$ atom pairs; 10,000 resamples). Because severe failures are rare and the sample has only 10 clusters, raw agreement is the primary task-gate figure.

Mean preference ratings are 1.02 and 1.03 for the two annotators and 0.83 for the judge. Failure-subtype labels were not reconciled across annotators; the audit therefore supports only the overall

Comparison	n	κ	95% CI	Exact
<i>Preference atoms (linear weighted)</i>				
Human 1–Human 2	44	.72	$[\cdot52, \cdot87]$	80.0%
Human 1–Judge	45	.52	–	60.0%
Human 2–Judge	44	.53	–	63.6%
<i>Task failure</i>				
Human 1–Human 2	30	.84	–	96.7%
Human 1–Judge	30	.37	–	73.3%
Human 2–Judge	30	.29	–	70.0%

Table 2: Agreement on post-calibration ratings. Task-failure positive counts are 4 (Human 1), 3 (Human 2), and 12 (judge). These values are not held-out inter-rater reliability.

task-gate comparisons in Table 2, not subtype-level agreement or the between-condition fabrication contrast.

C Robustness Analyses

Table 3 expands the checks summarized in the Results. Differences are computed after averaging the two generations within each case–model cell. Exact sign tests exclude ties.

Comparison	Check	n	Diff.	p
Imp. – T0	user1	51	+0.194	<.0001
Imp. – Full	user1	52	+0.020	.8601
Imp. – T0	user2	14	+0.429	.0034
Imp. – Full	user2	14	+0.051	.3877
Imp. – T0	excl. inversions	62	+0.252	<.0001
Imp. – Full	excl. inversions	63	+0.030	.3604
Imp. – T0	source clusters	30	+0.268	<.0001
Imp. – Full	source clusters	30	+0.046	.4049
Imp. – T0	target clusters	58	+0.249	<.0001
Imp. – Full	target clusters	59	+0.024	.2559
Imp. – T0	normal termination	52	+0.274	<.0001
Imp. – Full	normal termination	52	+0.038	.7283

Table 3: Sensitivity analyses for the primary imperative distilled-preference comparisons. T0 denotes target-only prompting and Full denotes full-history prompting.

D End-to-End Case Audits

This appendix audits two cases end to end, one from each user (translated from Chinese, with company and person names redacted pending release review; the original texts accompany the planned data release). The first, the Figure 1 case, is a transfer failure with a mechanism diagnosis; the second is a successful functional transfer under the same protocol. Both audits follow the same path: the scene, the user’s own words, the annotated standard, the later task, the rubric, contrasting answers with judge rationales, and a verdict. Every atom

carries the user’s verbatim words as its source anchor. One shared rule applies throughout: when an atom’s trigger is absent from the target or the answer, the atom is marked N/A and excluded, never scored as a zero.

D.1 A transfer failure: the Figure 1 case

Source correction (April 25). The user, a product manager by background (self-described), is reading a market analysis of LLM API pricing and asks for the distribution of API call volume across companies and models. The assistant proposes a research plan organized around market share and data-credibility tiers. The user replies: “*What I actually care about is how everyone is using these, and who is using them.*”

Annotated preference (USER_CONFIRMED). The corrective intent is to look through the phenomenon to what lies beneath: the user portrait behind a product. The user consistently unpacks products through who uses them and how, and expects this kind of answer across many questions.

Target question (June 5). Six weeks later, reading a digest item about *Job for Agent* (a hiring platform where enterprises post automatable tasks and AI-agent developers submit agents to “apply”), the user asks: “*This is quite interesting—what is this for? Why is it designed this way?*” The question does not mention the April interaction.

Preference atom 1 (how it is used). *Criterion:* when analyzing a product, market, or model-usage question, the assistant should explain how specific users use the object, in which workflow or scenario, and what problem it solves, making real usage the basis for explaining product value or market phenomena—rather than stopping at feature lists, market share, or abstract scenario labels. *Score 0:* no account of how users actually use it or what problem it solves; the answer is mainly a list of features, market facts, or macro trends. *Score 1:* mentions usage scenarios or purposes, but stays at labels such as “used by enterprises” or “for automation”, without unfolding how users work with it in which workflow and what concrete problem it solves. *Score 2:* uses at least one concrete user workflow to show in what situation users do what and solve which problem, and grounds the core analysis of value, design, or market difference in that usage.

Preference atom 2 (who uses it). *Criterion:* the assistant should say who uses it—identify the concrete user roles or portraits relevant to the phenomenon and explain their goals, constraints, or payment relations—rather than only using broad labels such as “enterprises”, “developers”, or “consumers” that cannot explain differences. *Score 0:* no actual users identified, or users written generically without any distinguishing feature. *Score 1:* names one or more user roles but only names them, without explaining why these users use it, what constraints they face, or why different users choose differently. *Score 2:* identifies the relevant user roles or portraits and uses their goals, constraints, working styles, or payment relations to explain usage choices, product design, or market differences; the portrait enters the core argument.

Answer excerpts (DeepSeek V4 Flash). Each condition was sampled twice; atom scores (atom 1/atom 2) are given per sampled answer. The three excerpts contrast the answer without memory, a functional realization of the standard (score 2), and a nominal realization (score 1).

The answer without memory (target-only condition; both samples scored 1/1): “The highlighted passage comes from a product called Job for Agent—a hiring platform for AI agents rather than people. For enterprises: publish tasks suited to AI automation, such as ‘scrape competitor prices daily and generate a report’. For AI developers: submit your agent to ‘apply’ for these tasks; the platform provides 80+ task templates. Why this design? (1) It fills a market gap: traditional hiring platforms serve humans, and agents lack a dedicated job market. (2) It lowers the enterprise barrier: post a task and ‘hire’ a ready-made agent instead of building AI tooling.”

The answer that reached a 2 (full-history condition; second sample scored 2/1, first 1/1): “Enterprise side: publish a task suited to AI automation, such as ‘crawl competitor prices daily and generate an Excel report’, ‘auto-reply to common customer-support tickets’, or ‘monitor server logs and alert on anomalies’; the platform provides 80+ task templates, plus an automation calculator that analyzes the posted job description and estimates how much of it can be automated, helping the enterprise decide whether the task is worth handing to an agent... Developer side: register as an AI developer (680+ so far) and submit your agent to ‘apply’ for a task, the way a job seeker submits a resume...”

Why this design? The founder’s candid admission: 95% of scenarios do not need a full agent—most enterprises need ‘a script that runs every morning, moves data from A to B, then sends an email’. So the platform encourages one task, one agent, rather than a general AI colleague, which lowers both the development barrier and the enterprise trust barrier: task boundaries are clear, and failures are easy to fix.” One judge vote explains the 2 on atom 1: the answer gives at least one concrete user workflow showing how enterprises actually use the platform, and grounds the design analysis in that usage.

The answer under the distilled preference (imperative condition; both samples scored 1/1): “Here is a breakdown starting from your analytical preferences... Credibility note: the numbers come from the product itself (self-promotion risk), but ‘800+ jobs analyzed’ is a publicly verifiable statistic—medium-high credibility... Dimension one: company/product positioning... Dimension two: tiers and scenarios—high automation potential: repetitive, rule-clear tasks (data scraping, report generation)... Dimension three: data-validation direction: cross-checking with third-party routing data would show...”

Judge rationales for the score-1 verdicts (imperative condition). Atom 2, one of three votes: *“The answer identifies two user roles (task-publishing enterprises and agent developers) but only names them: it does not explain which kind of enterprise would choose this platform under what constraints, what motivates the developers, or how the two-sided payment relation works—the user portrait never enters the core argument.”* Atom 1, another vote: *“‘Repetitive, rule-clear tasks such as data scraping and report generation’ is a scene label, not a concrete user workflow.”*

Diagnosis. The distilled memories locate the failure. All three tested models compressed the April interaction into its neighboring data-credibility standard. DeepSeek: *“prefer real call data over positioning inference, tier the credibility of sources”*; Qwen-Max: *“verify positioning inferences with real data, value data credibility and provenance”*; GPT-4o: *“validate hypotheses with data, cross-check multiple sources”*. None preserved the user’s own sentence about how and by whom the object is used. The imperative condition then faithfully executed the wrong standard: the answer above performs credibility tiers and verification dimensions while both preference atoms stay at score 1. The

pipeline did not fail to remember; it remembered the adjacent preference.

D.2 A successful transfer: reusable judgment frameworks

Scene and source exchange (May 18). The second user works in post-investment analysis at an investment firm (self-described). She is reading an interview in which a hardware-company founder says he cares about only two questions when entering a product category (is there real user value, and is the person in charge right). She asks why these two questions matter, then builds on the assistant’s explanation: *“This ‘worth doing + this person can do it’ summary is good. My thinking is scattered, and I want to accumulate judgment and cognitive frameworks; we should keep abstracting them from all kinds of information. When a framework is abstract, add cases.”* When the assistant proposes candidate frameworks, she pushes further: *“When dimensions are not orthogonal, you can also split the smaller branches beneath them; we want the frameworks as accurate and explainable as possible... the core goal is to improve my judgment of people and things in a logical, systematic way.”*

The standard the user established (USER_CONFIRMED). Scattered reading should be abstracted into reusable judgment frameworks; frameworks should be refined over time and supplemented with concrete cases; the ultimate goal is improving the user’s own judgment, not mechanical summarization.

The target task (June 11). Weeks later, reading an article about a consumer-hardware company’s investment structure (the company supplies 20% of each fund as general partner, local-government funds supply 80% as limited partners, and the company holds full investment control), the user asks: *“Where are the pros and cons of this model?”* The question does not mention the earlier exchange.

What the rubric rewards. The owner-confirmed criteria operationalize what counted as a usable framework in this exchange: named dimensions, the conditions under which a judgment holds or flips, and concrete cases. *Atom 1:* when the user asks for the pros and cons of a business model, organize them into structural dimensions (capital, incentives, risk transmission), giving each dimension the condition under which it flips, rather than two flat lists. *Score 0:* two-column pros-and-cons

bullets, no structure, no conditions. *Score 1*: dimensions, but without the conditions under which they reverse, or without cases. *Score 2*: structural dimensions plus flip conditions, with concrete in-material events as cases, transferable to judging similar models. *Atom 2 (evidence boundary)*: if the answer extends beyond the provided material, extensions must carry a source or an explicit unverified marker; inventing external cases or numbers scores 0.

The answer without memory (target-only condition; second sample, scored 0). “Pros: 1. For the company (GP): leverage four times the capital with a 20% stake, hold 100% of investment decisions, and take 20% of the upside on success. 2. For local governments (LP): a fast way to meet investment-attraction targets and gain a ‘tech unicorn’ badge... Cons: 1. Risk and return are severely mismatched... 2. Distorted incentives... 3. Doubtful sustainability...” The judge’s rationale for the 0: a standard two-column pros-and-cons list; no structural dimensions, no flip conditions, and the in-material facts appear only as isolated bullets.

The answer under the distilled preference (imperative condition; both samples scored 2). The second sample organizes the same facts into named dimensions, each carrying the condition under which it reverses: “(1) GP capital efficiency: 20% of own capital leverages a fourfold position... (2) LP risk-return inversion: the LP bears 80% of the downside with only nominal oversight... If the GP’s true success rate is high, say above 80%, the LPs’ 80% exposure is actually controllable... (3) Asymmetric incentives: success pays the GP 20% of the upside while failure costs it only 20% of the loss, which pushes the GP toward high-failure, high-payoff bets... A falsifiable rule: whoever bears the largest risk should hold the largest control; when control and risk diverge, the structure distorts... Sustainability depends on whether the GP can genuinely select projects and whether the LPs can audit after the fact.” The judge’s rationale for the 2: the answer names structural dimensions, gives each the condition under which it reverses, uses in-material events as cases, and the resulting framework transfers to judging similar models. The separate task gate raises no flag on any answer in this case: every extension stays within the provided material.

Verdict. The facts available to both answers are identical; what the transferred standard changes is the organization of judgment, from a flat list to a reusable framework, which is what a 2 requires. Averaged over the three tested models, this case rises from 0.17 under target-only prompting to 1.33 under imperative prompting, the largest paired gain in the benchmark, with no fabricated-evidence flag; here the distilled standard also outperformed the full interaction. Unlike the first audit, the distilled memory preserved the intended standard: DeepSeek’s summary reads “*abstract transferable cognitive frameworks and judgment rules from conversations rather than staying at specific information; value orthogonality, explainability, and falsifiability; supplement with cases and counterexamples.*” The two audits differ only in what the distillation kept.

E Fabricated-Evidence Examples

The two examples show the two forms the operational definition names, one from each memory condition: an answer that invents external evidence of the kind the transferred standard rewards, and an answer that carries names from the source dialogue into unsupported attributions. Neither example claims the details are false in the world; what the gate records is that they have no support in the materials the answer was supposed to draw on.

Invented reports serving the injected standard (imperative prompting). The first user is reading a daily digest of trending open-source repositories. In the source exchange four days earlier, she had asked for research on a tool that interested her and corrected the assistant for staying inside the article; the annotated standard asks for verified information beyond the provided material rather than a restatement of it. Now she asks about a repository that has trended for two days running: “*It has been on the list two days in a row; introduce it in detail.*” The digest entry lists the repository at 26.4k stars. Under imperative prompting, Qwen-Max supplies exactly the beyond-material evidence the standard rewards: it cites a Medium post-mortem crediting the project with 110,000 stars in three months and 28 consecutive days atop the weekly ranking, and an AlphaSignal report of over 140,000 stars. None of these reports appear in the packet, and the claimed figures contradict the 26.4k stars on the same page. The answer receives a preference score of 2 and a separate fabricated-evidence flag; this

case draws fabrication flags under every condition, including no memory.

Source names carried into unsupported attributions (full-history prompting). The same investment-analyst user is reading a portfolio overview of a venture firm. She first asks which partner invested in one AI-pharmaceutics company the article mentions; the assistant offers a guess about coverage areas and two lookup paths (check the shareholder registry; search the original funding announcement). She corrects it: *“Don’t give me the path; I need you to follow the path and give me the name as the answer.”* The annotated standard: verifiable questions should be answered with the answer itself, sources attached, and unavailable items marked as such. Later in the same reading, she asks which named partner led each of roughly fifteen portfolio investments; the target packet names only the firm’s founding partner and attributes no individual investments. Under full-history prompting, DeepSeek V4 Flash assigns a named partner to every company, complete with investment years, rounds, and quoted public statements; the only partner names available to it come from the source exchange and the packet, and its two generations assign different partners to several of the same companies. The preference judge scores the give-the-answer atom 2 (every item receives a name) and the honesty atom 0 (nothing is marked unavailable, no sources attached); the task gate flags fabricated evidence.

The first answer invents external reports; the second reuses names from the source dialogue as scaffolding for invented attributions. In both, the answer supplies exactly the kind of specific evidence the transferred standard rewards, and the materials in front of the model support none of it. Two examples cannot show that memory caused these details; that comparison belongs to the paired rates in the Results.

Role-Specific Strengths and Alignment in LLM Cross-Play and Self-Play

Kata Naszádi, Christof Monz

Language Technology Lab, University of Amsterdam

Correspondence: k.naszadi@uva.nl

Abstract

Dialogue-game benchmarks usually pair a model with itself, and the resulting self-play score is treated as a measure of that model’s ability. We test whether self-play scores measure partner-independent ability by comparing them with cross-play evaluations. Across nine Clembench games, we evaluate every possible seating of three popular open-weight models. The cross-play reading exposes additional model strengths and weaknesses that are not detectable in self-play. In four of nine games, a mixed pair of two different models beats the best self-play score; in another four cases, a model does better paired with itself than paired with the strongest available partner. Looking at the transcripts, two mechanisms explain most of this: some pairs succeed because one partner writes more detailed messages that keep the interaction on track, while some self-play scores are inflated because both copies of a model make the same mistake and agree by accident.

1 Introduction

Dialogue games have emerged as an informative paradigm for evaluating chat-optimized language models. Rather than solving static test items, models engage in goal-directed conversations in which each generated utterance constitutes an action toward completing a shared task. [Chalamalasetti et al. \(2023\)](#) introduce *Clembench*, a benchmark of dialogue games spanning both asymmetric interactions, where players have distinct roles such as clue-giver and guesser, and symmetric interactions like negotiation and debate, where both players have the same role. In the standard evaluation setup, both player roles are filled by the same model. The resulting self-play score is then interpreted as a measure of that model’s capability.

Recent work suggests that the self-play reading of LLM players deserves more scrutiny than

it usually gets. Models recognize and favor generations resembling their own ([Panickssery et al., 2024](#)) and models increasingly make the *same* mistakes ([Kim et al., 2025](#)). The overlap in the types of mistakes grows with model capability even across model classes ([Goel et al., 2025](#)). A self-play score rewards agreement, but this agreement may stem from shared errors rather than shared competence. In language-based games, this issue is even more pronounced because language itself serves as the coordination medium. Alignment or misalignment between partners does not just occur at the level of high-level strategy, but from the way each model produces and interprets utterances.

We contribute an empirical data point to this debate: a complete cross-play and self-play evaluation of three open-weight models from different providers and architecture families (QWEN, GEMMA, MINISTRAL) across nine Clembench dialogue games. We analyze the resulting performance matrix against two prior expectations that are based on the assumption that scores reflect partner-independent competence.

E1 (Symmetric Roles): Self-play performance represents a baseline of capability. A more competent partner should consistently improve the joint score.

E2 (Asymmetric Roles): A model with a distinct strength in one role should consistently elevate its partner.

We find that E1 and E2 largely hold as baselines, and E2 reveals a split that self-play cannot expose: role-specific strengths vary across models. For example, GEMMA outperforms QWEN as the clue-giver in three separate games, while the nominally weakest model, MINISTRAL, excels as a guesser for high-frequency words because its

narrow vocabulary focus prevents it from overthinking the clues.

We also find aligned errors where successful coordination does not reflect true competence, violating E2 and creating a self-play advantage over playing with the best available partner.

2 Related Work

From multi-agent reinforcement learning, we know that self-play policies coordinate through private conventions that do not necessarily transfer to novel partners (Hu et al., 2020; Bard et al., 2020; Kottur et al., 2017). While such conventions are expected when policies are trained together, LLMs evaluated on dialogue games typically have no game-specific joint training. Nonetheless, modern LLMs are frequently fine-tuned on their own generated outputs (Huang et al., 2023; Wang et al., 2023). This self-training regime may implicitly result in model-specific conventions and self-preference biases, making the study of cross-play dynamics relevant to understanding how these models coordinate.

A growing number of works investigate communicative multi-agent solutions to solve complex tasks. Notably, most of these works use multiple copies of the same model to build a community of players (Wang et al., 2024; Park et al., 2023; Liang et al., 2024). An important exception is Bianchi et al. (2024), who analyzed cross-play in a negotiation setup, let different models negotiate with each other, and found that who sits across the table substantially changes the outcome, with weaker partners dragging results down.

3 Setup

Models. We use three open-weight instruction-tuned models from three providers: QWEN (Qwen3.5-27B Qwen Team, 2026), GEMMA (Gemma 4 12B-IT Gemma Team, 2026), and MINISTRAL (Ministral-3-14B-Reasoning-2512 Mistral AI, 2025). Models are invoked through Clembench’s model interface with uniform greedy decoding (temperature 0.0) for every request, so any behavioral difference between models is not a sampling artifact. QWEN and MINISTRAL are reasoning-capable models. Both are run without eliciting their reasoning mode: QWEN’s chat template is called with `enable_thinking=false`, MINISTRAL’s reasoning channel is parsed out via a dedi-

cated reasoning parser and is never prompted for thinking. In the tables, *qw*, *ge*, and *mi* denote Qwen3.5-27B, Gemma 4 12B-IT, and Ministral-3-14B-Reasoning-2512, respectively; the corresponding run labels are `qwen3.5-27b`, `gemma-4-12b`, and `ministral-14b`.

Games. We evaluate on nine Clembench dialogue games spanning two role structures. Five have *asymmetric* roles, where the two seats perform different jobs: TABOO (describer/guesser, 60 instances), REFERENCEGAME (expression-generator/answerer, 90), CODENAMES (clue-giver/guesser, 130), IMAGEGAME (instruction-leader/follower, 40), and WORDLE_WITHCRITIC (guesser/critic, 30). Four have *symmetric* roles, where both seats do the same job: CLEANUP (27), DOND (40), MATCHIT_ASCII (40), and HOT_AIR_BALLOON (36). In every game, P1 starts the interaction. In the symmetric games, P1 and P2 have the same role, and the labels indicate turn order. In the asymmetric games, P1 is the first role listed above—describer, expression-generator, cluegiver, instruction-leader, or guesser, respectively—and P2 is the corresponding second role. Although the roles in the symmetric games are identical, turn order can significantly affect performance (e.g., by determining which player makes the first offer in negotiation). For more detailed descriptions of the games, see Chalamalasetti et al. (2023).

Token and context budgets. Each game used greedy decoding and a 1024 token limit except for three games: for HOT_AIR_BALLOON, REFERENCEGAME and WORDLE_WITHCRITIC the maximum number of tokens was set to 4096. The maximum size of the context was capped at 65,536 tokens. We count any episode whose full dialogue history exceeds it as an abort with no score, on the grounds that failing to converge within the budget is itself a form of failing at the task.

Metrics. Each game-specific Clembench score ranges from 0 to 100. For example, WORDLE_WITHCRITIC combines success and speed, whereas REFERENCEGAME’s score is its success rate (for all definitions, see Chalamalasetti et al., 2023). We distinguish between aborted episodes due to protocol violations and played episodes. We only treat a result as robust if it holds under three views: *quality*, the mean score over played episodes (blind to abort-rate differences); *clem-*

Game	Pair	Clemscore	$\Delta P1$	$\Delta P2$
taboo	ge $\triangleright$ qw	77.7	+2.2	+4.3
codenames	ge $\triangleright$ qw	39.2	+11.5	+15.4
dond	qw $\triangleright$ ge	83.3	+36.1	+6.9
matchit_ascii	ge $\triangleright$ qw	97.5	+15.0	+5.0

Table 1: Synergy: mixed pairs whose score beats *both* component models’ own self-play score. $\Delta P1/\Delta P2$: points over Player 1’s / Player 2’s respective self-play.

score, the mean score with aborted rounds entering the mean with a score of 0; and *matched instances*, quality recomputed only over the instance IDs played in *all* compared cells, which removes any selection effect from one pairing aborting on harder instances than the other. Matched-instance sample sizes range from $n=0$ to $n=76$ depending on the game and pairing. The full matrix under each of the three views is reported in Appendix A.

4 Results

We now present a subset of the full results, focusing on cases where pairing models from different families reveals mechanisms that self-play scores alone cannot expose. The complete 9×9 game-by-pairing matrix under all three metrics from §3 is given in Appendix A: quality (Table 4), clem-score (Table 5), and matched instances (Table 6).

First, we mention that three games in particular carry high abort rates that are worth explaining before the case studies. In CODENAMES, QWEN as cluegiver drives abort rates of 43–52/130 episodes; this is because Qwen’s broad clues make the receivers guess outside of the valid candidates. In HOT_AIR_BALLOON, aborts come down to: QWEN’s free-form reasoning buries the required structured tags, and all 24 complexity/negotiation episodes abort in every QWEN-containing run. In IMAGEGAME GEMMA has a high abort rate as instruction-giver because it echoes the instruction template itself rather than producing a drawing command.

In Table 1, we display cases where cross-play performance exceeds both self-play scores. This allows us to address an interesting question: do models design utterances for themselves? While there is no explicit theory-of-mind mechanism implemented in these models, we can still ask whether a model is best at resolving clues generated by its own copy. Surprisingly, the answer is no. QWEN tends to give high-level, abstract clues, which is best attested by TABOO statistics:

Game	Pair	Clemscore	$\Delta P1$	$\Delta P2$
dond	qw $\triangleright$ mi	9.1	−38.1	−11.2
matchit_ascii	qw $\triangleright$ mi	77.5	−15.0	−10.0
matchit_ascii	mi $\triangleright$ qw	75.0	−12.5	−17.5
matchit_ascii	mi $\triangleright$ ge	70.0	−17.5	−12.5
imagegame	qw $\triangleright$ mi	65.1	−31.2	−6.8

Table 2: Anti-synergy: mixed pairs whose score falls *below both* component self-plays’ score. *dond*’s QWEN $\triangleright$ MINISTRAL is the matrix’s strongest anomaly and reverses to 33.8 when MINISTRAL speaks first instead.

on average, it addresses 2.4 words at once, while Gemma only targets 1.7. QWEN gets confused by the abstract clues of its own copy and often hallucinates invalid answers, yet is much more successful when paired with Gemma’s more concrete strategy.

In MATCHIT_ASCII, where players have to describe ASCII-grid patterns, the same GEMMA-as-describer advantage holds over QWEN self-play. GEMMA’s approach is similarly low-level as for taboo and codenames: it describes the grid cell by cell while QWEN’s high-level natural language descriptions often contain inaccuracies and get misunderstood (further examples in Appendix B).

In the negotiation game “dond”, QWEN typically negotiates successfully with its own copy in text, but the player making the closing final “[Proposal: ...]” botches the deal by proposing the partner’s share contradicting the discussed agreement. The pairing QWEN $\triangleright$ GEMMA combines QWEN’s negotiation with GEMMA’s discipline of the formatted proposal generation, which results in the best performance.

In Table 2 we display cases where self-play performance exceeds both cross-play scores even though the partners exhibit stronger performance individually. These cases contradict our initial expectations (E1 and E2), under which being paired with a stronger partner should improve performance. MINISTRAL playing MATCHIT_ASCII is the strongest case for this; the self-play advantage exists purely because both copies of MINISTRAL misread the ASCII grids in the exact same way, agreeing that two grids match for entirely wrong reasons. Similar aligned misinterpretations can be found in the imagegame for MINISTRAL’s self-play, but the models’ unrelated errors mask this self-play advantage (see Figure 1 for an example and Appendix B for more). These are clear examples of correlated error masquerading as success-

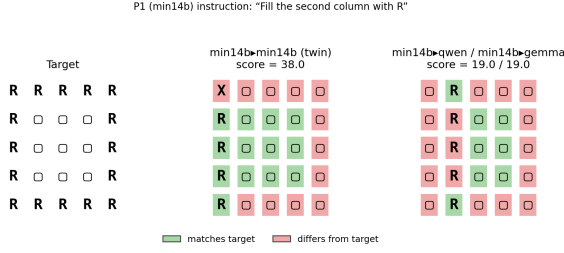


Figure 1: MINISTRAL uses "second column" to refer to the first one and its own copy decodes it the same way while GEMMA gets misled.

	GEMMA	QWEN	MINISTRAL
high_en	65.8	65.3	69.7
medium_en	55.6	57.2	51.4
low_en	57.2	66.1	57.2

Table 3: Average taboo score in the guesser role, by target-word frequency band, averaged over both partners and self-play (aborts scored 0).

ful communication.

The QWEN-MINISTRAL pairing in imagegame is another example of audience mismatch: the lower-capability MINISTRAL fails with QWEN as the latter gives highly compressed descriptions that only the higher-capability GEMMA can resolve.

Frequency effect of target words Not every partner effect surfaces at the game level. MINISTRAL is the best model in the guessing role in taboo’s high-frequency range (Table 3: 69.7 vs. 65.8 for GEMMA, 65.3 for QWEN), plausibly because it is not thrown off by assigning high probability to rarer words. The effect is most striking in its best pairing: guessing for QWEN’s descriptions, MINISTRAL scores 95, 15 points above Qwen’s own self-play score of 80 in the same high-frequency range.

5 Discussion

Our evaluation set out to investigate the existence of unexpected, systemic self-play advantage in dialogue games played by LLMs. We specifically looked for dog-whistle-like patterns: a word or phrasing whose success with a given partner could not be derived from the optimal communicative strategy the game itself demands.

We find clear evidence of such a self-play advantage in only one specific condition with our weakest tested model MINISTRAL. When

the game involves an external *grounding element* (such as ASCII grids), the MINISTRAL self-play exhibits consistent, correlated errors. Because both copies of MINISTRAL consistently misinterpret the grounding in the exact same way, communication succeeds. This coordination falls apart immediately in cross-play because a partner model (like GEMMA) interprets the grounded reference correctly, exposing the unconventional language use of MINISTRAL.

Most patterns reported in this paper, however, are readily interpretable within each game’s dynamics and are well explained by known communicative strategies without having to turn to deeper model-specific anomalies. The best information-givers are highly specific: GEMMA’s Codenames clues stay close to a small set of candidates, and its ImageGame instructions enumerate every cell rather than compressing them into a single command. The best information-receivers are literal, avoiding over-interpretation: MINISTRAL’s Taboo guesses follow the common reading of a clue rather than searching for a more abstract alternative, allowing it to outperform stronger partners on high-frequency words. These differences reflect *audience design*: a speaker’s optimal message depends on the addressee’s ability to reconstruct what is left implicit (Naszadi et al., 2024). QWEN’s REFERENCEGAME instructions are efficient, but only for a capable listener. GEMMA’s longer, exhaustive descriptions are safer but less natural. Because Clembench rewards communicative accuracy more than efficiency, this cautious style is consistently favored.

The absence of stronger evidence for systemic self-play advantage does not mean such effects do not exist. Rather, the communication games considered here may not demand sufficiently sophisticated language use for detecting such conventions. In many settings, performance is still limited by more basic communicative failures. For example, in CODENAMES, hallucinated associations dominate error more than any partner-specific clue strategy would. If lower-level failures dominate the error profile, there is little room to detect more subtle "secret handshake" patterns.

Limitations

We follow the standard *Clembench* evaluation protocol, including its recommended deterministic decoding setup (greedy decoding with temperature

0). This means that each reported score is based on a single realization of each game. Our analysis does not capture the variability that may arise under stochastic decoding. In addition, some benchmark games contain only a small number of evaluation instances, which limits the statistical power of comparisons on individual games. Finally, our analysis of language use relies on qualitative human annotation. Consequently, some factors contributing to successful or unsuccessful coordination may not be reflected in our analysis or may not align with the interpretations presented in the discussion (Section 5).

Acknowledgments

This research was funded in part by the Netherlands Organization for Scientific Research (NWO) under project number VI.C.192.080. We also received funding from the Hybrid Intelligence Center, a 10-year programme funded by the Dutch Ministry of Education, Culture and Science through the Netherlands Organisation for Scientific Research with grant number 024.004.022.

References

- Nolan Bard, Jakob N Foerster, Sarath Chandar, Neil Burch, Marc Lanctot, H Francis Song, Emilio Parisotto, Vincent Dumoulin, Subhodeep Moitra, Edward Hughes, et al. 2020. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence*, 280:103216.
- Federico Bianchi, Patrick John Chia, Mert Yuksekgonul, Jacopo Tagliabue, Dan Jurafsky, and James Zou. 2024. How well can llms negotiate? negotiationarena platform and analysis. In *Proceedings of the 41st International Conference on Machine Learning*, ICML’24. JMLR.org.
- Kranti Chalamalasetti, Jana Götze, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2023. [clembench: Using game play to evaluate chat-optimized language models as conversational agents](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11174–11219, Singapore. Association for Computational Linguistics.
- Gemma Team. 2026. Gemma 4 12b-it. <https://huggingface.co/google/gemma-4-12B-it>. Model card.
- Shashwat Goel, Joschka Strüder, Ilze Amanda Auzina, Karuna K Chandra, Ponnurangam Kumaraguru, Douwe Kiela, Ameya Prabhu, Matthias Bethge, and Jonas Geiping. 2025. Great models think alike and this undermines ai oversight. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML’25. JMLR.org.
- Hengyuan Hu, Adam Lerer, Alex Peysakhovich, and Jakob Foerster. 2020. “other-play” for zero-shot coordination. In *International conference on machine learning*, pages 4399–4410. PMLR.
- Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2023. [Large language models can self-improve](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 1051–1068, Singapore. Association for Computational Linguistics.
- Elliot Kim, Avi Garg, Kenny Peng, and Nikhil Garg. 2025. Correlated errors in large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, ICML’25. JMLR.org.
- Satwik Kottur, José Moura, Stefan Lee, and Dhruv Batra. 2017. [Natural language does not emerge ‘naturally’ in multi-agent dialog](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2962–2967, Copenhagen, Denmark. Association for Computational Linguistics.
- Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Shuming Shi, and Zhaopeng Tu. 2024. [Encouraging divergent thinking in large language models through multi-agent debate](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17889–17904, Miami, Florida, USA. Association for Computational Linguistics.
- Mistral AI. 2025. Mistral-3-14b-reasoning-2512. <https://huggingface.co/mistralai/Mistral-3-14B-Reasoning-2512>. Model card.
- Kata Naszadi, Frans A Oliehoek, and Christof Monz. 2024. [Communicating with speakers and listeners of different pragmatic levels](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21777–21783, Miami, Florida, USA. Association for Computational Linguistics.
- Arjun Panickssery, Samuel R. Bowman, and Shi Feng. 2024. Llm evaluators recognize and favor their own generations. In *Proceedings of the 38th International Conference on Neural Information Processing Systems*, NIPS ’24, Red Hook, NY, USA. Curran Associates Inc.
- Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. [Generative agents: Interactive simulacra of human behavior](#). In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, UIST ’23, New York, NY, USA. Association for Computing Machinery.

Qwen Team. 2026. Qwen3.5-27b. <https://huggingface.co/Qwen/Qwen3.5-27B>. Model card.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. Self-instruct: Aligning language models with self-generated instructions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.

Zhenhailong Wang, Shaoguang Mao, Wenshan Wu, Tao Ge, Furu Wei, and Heng Ji. 2024. Unleashing the emergent cognitive synergy in large language models: A task-solving agent through multi-persona self-collaboration. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 257–279, Mexico City, Mexico. Association for Computational Linguistics.

A Full result matrices

Tables 4–6 report the complete 9×9 game-by-pairing matrix under the three metrics from §3: Table 4 gives quality (abort-blind mean Main Score over played episodes), Table 5 gives clem-score (the same means with aborted episodes scored 0), and Table 6 gives the matched-instance view (quality recomputed only over instance IDs completed by all nine pairings). Blank cells mark pairings or games with zero completed or zero matched episodes.

B Transcript excerpts

MATCHIT_ASCII with MINISTRAL Figures 2 and 3 give examples of correlated errors by MINISTRAL.

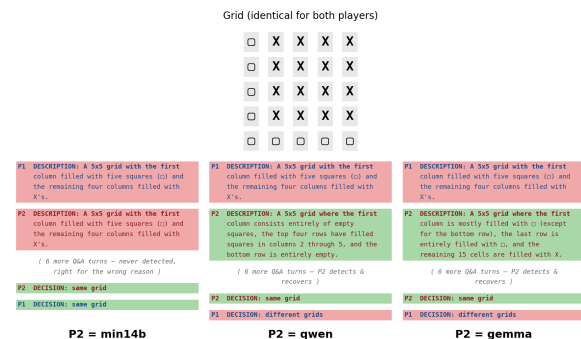


Figure 2: Two MINISTRAL instances give the same incorrect description of a grid and agree that the grids match. The error is exposed in cross-play, where the partner detects it in the dialogue.

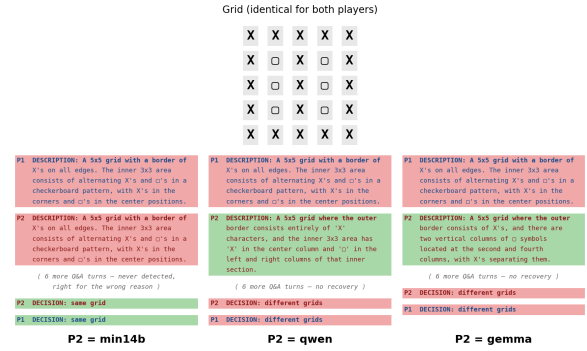


Figure 3: Two MINISTRAL instances give the same incorrect description of a grid and agree that the grids match. The other model does not detect the incorrect description, and the interaction fails for both partners.

GEMMA advantage over QWEN as P1 Figures 4–5 give two REFERENCEGAME instances where GEMMA's detailed, enumerative expression lets MINISTRAL pick the target while QWEN's compressed expression sends MINISTRAL to a distractor. Figures 6 and 7 show two further examples contrasting MINISTRAL and GEMMA in the receiver role.

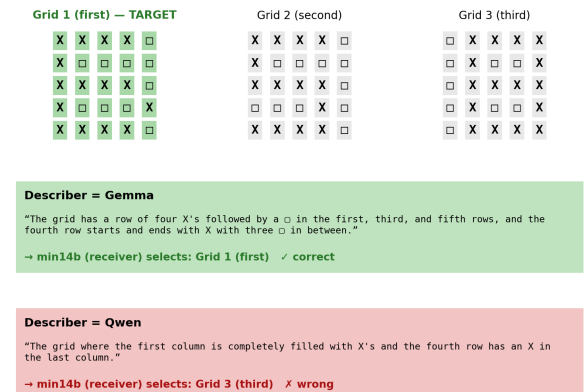


Figure 4: REFERENCEGAME instance 15 (number_grids), score 100 vs. 0, receiver fixed at MINISTRAL. Gemma describes every row. Qwen compresses to two features.

Game	qw-qw	ge-ge	mi-mi	qw>ge	ge>qw	ge>mi	qw>mi	mi>qw	mi>ge
taboo	78.6 (56)	79.5 (57)	43.0 (50)	77.0 (55)	83.3 (56)	73.0 (58)	75.7 (57)	48.9 (46)	42.4 (46)
referencegame	95.6 (90)	100.0 (87)	62.2 (90)	95.4 (87)	92.2 (90)	80.0 (90)	68.9 (90)	78.9 (90)	81.6 (76)
codenames	35.6 (87)	33.6 (107)	19.2 (78)	34.5 (84)	44.7 (114)	36.1 (97)	35.9 (78)	26.9 (104)	24.5 (110)
clean_up	71.3 (25)	79.4 (13)	−(0)	75.1 (14)	70.7 (19)	−(0)	−(0)	59.7 (10)	51.0 (8)
dond	47.2 (40)	76.4 (40)	20.8 (39)	83.3 (40)	78.2 (40)	35.2 (40)	9.1 (40)	33.8 (40)	30.5 (39)
matchit_ascii	92.5 (40)	82.5 (40)	87.5 (40)	85.0 (40)	97.5 (40)	85.0 (40)	77.5 (40)	75.0 (40)	70.0 (40)
imagegame	96.3 (40)	−(0)	73.7 (39)	96.6 (40)	−(0)	−(0)	66.8 (39)	79.5 (40)	79.8 (40)
hot_air_balloon	81.1 (10)	75.9 (29)	59.8 (6)	54.1 (12)	73.8 (12)	50.9 (21)	64.6 (12)	0.0 (1)	21.0 (4)
wordle_withcritic	85.4 (16)	91.0 (13)	75.0 (4)	80.0 (15)	82.8 (17)	76.1 (15)	75.9 (18)	30.0 (5)	33.3 (3)

Table 4: Quality (mean Main Score over played episodes; played count in parentheses). This is the abort-blind view described in §3: it isolates skill on completed episodes but does not penalize a pairing for aborting more often than another. All cells 493/493 episodes finished-or-aborted; context-overflow episodes counted as aborted (103 documented stubs). qw = qwen3.5-27b, ge = gemma-4-12b, mi = minstral-14b; X>Y: X speaks first.

Game	qw-qw	ge-ge	mi-mi	qw>ge	ge>qw	ge>mi	qw>mi	mi>qw	mi>ge
taboo	73.4	75.5	35.8	70.6	77.7	70.6	71.9	37.5	32.5
referencegame	95.6	96.7	62.2	92.2	92.2	80.0	68.9	78.9	68.9
codenames	23.8	27.7	11.5	22.3	39.2	26.9	21.5	21.5	20.7
clean_up	66.0	38.2	0.0	38.9	49.8	0.0	0.0	22.1	15.1
dond	47.2	76.4	20.3	83.3	78.2	35.2	9.1	33.8	29.7
matchit_ascii	92.5	82.5	87.5	85.0	97.5	85.0	77.5	75.0	70.0
imagegame	96.3	0.0	71.9	96.6	0.0	0.0	65.1	79.5	79.8
hot_air_balloon	22.5	61.1	10.0	18.0	24.6	29.7	21.5	0.0	2.3
wordle_withcritic	45.5	39.4	10.0	40.0	46.9	38.0	45.5	5.0	3.3

Table 5: Clemscore (aborted episodes scored 0). This is the abort-sensitive counterpart to Table 4: it folds a pairing’s abort rate back into the mean, so the two tables read together separate genuine skill differences from differences driven purely by protocol failures. qw = qwen3.5-27b, ge = gemma-4-12b, mi = minstral-14b; X>Y: X speaks first.

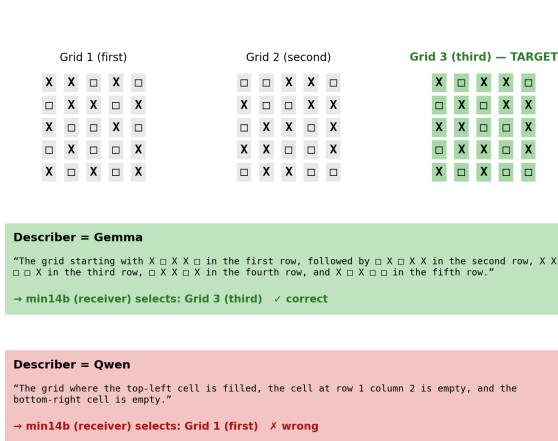


Figure 5: REFERENCEGAME instance 11 (random_grids), score 100 vs. 0, receiver fixed at MINISTRAL. GEMMATranscribes the pattern row by row, giving MINISTRAL the full target; Qwen names only three distinguishing cells and MINISTRAL selects the wrong one.

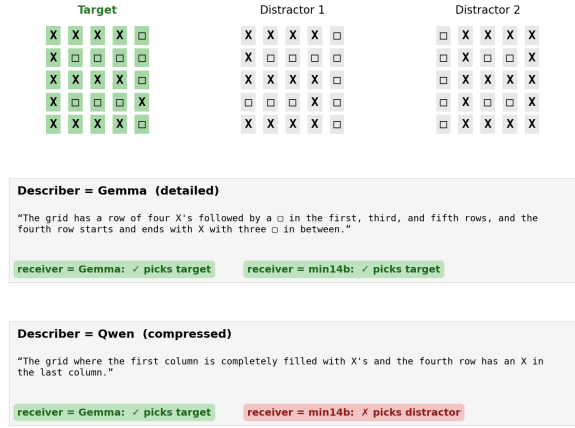


Figure 6: REFERENCEGAME instance 15 (number_grids). REFERENCEGAME is single-shot, so each describer’s referring expression is identical across receivers. GEMMA’s detailed, row-by-row expression is understood by both receivers. QWEN’s two-feature expression is enough for the stronger GEMMA receiver to recover the target but not for MINISTRAL, which selects a distractor.

Game	n	qw·qw	ge·ge	mi·mi	qw>ge	ge>qw	ge>mi	qw>mi	mi>qw	mi>ge
taboo	41	82.9	85.8	45.1	81.7	89.8	81.7	82.5	46.3	41.9
referencegame	76	96.1	100.0	65.8	100.0	93.4	81.6	72.4	81.6	81.6
codenames	16	50.0	56.2	18.8	56.2	56.2	31.2	56.2	37.5	31.2
clean_up	2	94.8	100.0	—	88.1	100.0	—	—	39.7	53.7
dond	38	49.7	77.8	20.8	83.0	79.7	37.0	9.5	32.4	31.3
matchit_ascii	40	92.5	82.5	87.5	85.0	97.5	85.0	77.5	75.0	70.0
imagegame	38	96.1	—	73.0	96.4	—	—	66.4	78.6	78.9
hot_air_balloon	0	—	—	—	—	—	—	—	—	—
wordle_withcritic	0	—	—	—	—	—	—	—	—	—

Table 6: Matched instances: mean Main Score restricted to the instance IDs played (non-aborted) in every compared pairing for a given game, removing any selection effect from one pairing aborting on harder instances than another (§3). n gives the number of matched instance IDs; games where no instance is completed by all nine pairings (HOT_AIR_BALLOON, WORDLE_WITHCRITIC) are left blank.

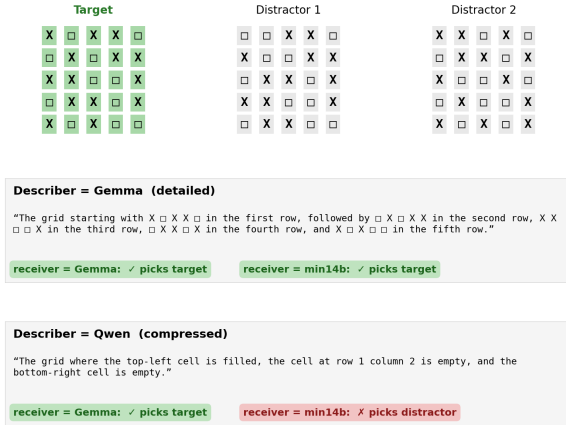


Figure 7: REFERENCEGAME instance 11 (random_grids), same describer × receiver design as Figure 6. On an irregular grid GEMMA transcribes the pattern row by row and both receivers succeed. Qwen names only three distinguishing cells.

InterviewArena: Estimating Uninformative Participants through Interactive Dialogue under Incomplete Information

Raika Koki^{1†} Karina Karasawa^{1†} Ryoki Kanayama^{1†} Yuki Suzumura^{1†} Haruto Fujita^{1†}
Ryoma Obara² Yusuke Sakai³ Hidetaka Kamigaito³ Katsuhiko Hayashi⁴ Syogo Matsuno¹

¹The University of Electro-Communications ²NEC Knowledge Science Research Laboratories

³Nara Institute of Science and Technology (NAIST) ⁴The University of Tokyo

k2530037@gl.cc.uec.ac.jp, koki.raika@agent.lab.uec.ac.jp, ryoma-obara@nec.com
{kanayama-r, fujita-h}@mm.inf.uec.ac.jp, katsuhiko-hayashi@g.ecc.u-tokyo.ac.jp
{sakai.yusuke.sr9, h.kamigaito}@is.naist.jp, {y.suzumura, matsuno}@uec.ac.jp

†Equal Contribution

Abstract

We introduce InterviewArena, an interview-style multi-agent framework for evaluating large language models (LLMs) under incomplete information. By treating missing information as a single factor, we disentangle missing-information identification from dialogue-based reasoning. Experimental results show that while LLMs can capture relative differences between participants, predicting specific missing information remains unstable, highlighting the need to evaluate these capabilities separately.

1 Introduction

As large language models (LLMs) acquire stronger reasoning capabilities, they are increasingly applied to tasks that require complex social reasoning, such as multi-agent collaboration or role inference through dialogue in settings like the Werewolf game. These tasks demand reasoning abilities similar to those used by humans in interactive environments. In particular, recent studies have proposed evaluation frameworks that simulate human behavioral norms, social interactions, and cognitive processes using LLMs (Hou et al., 2025; Sakai et al., 2025; Jiang et al., 2024; Zhao et al., 2025). In addition, game-style environments centered on dialogue and deduction, such as the Werewolf game, have been utilized to measure the reasoning capabilities of LLMs (Lai et al., 2023; Chalamalasetti et al., 2023; Costarelli et al., 2024; Song et al., 2025).

In particular, the Werewolf game is a hidden-role, incomplete-information game in which players must identify the werewolf disguised among villagers through dialogue and reasoning. In LLM evaluation, this setting has been adopted as a multi-agent benchmark where multiple LLMs interact through dialogue. By observing these interactions, the reasoning abilities and strategic behaviors of different models can be compared (Sato et al., 2024; Mukobi et al., 2023; Bailis et al., 2024). These

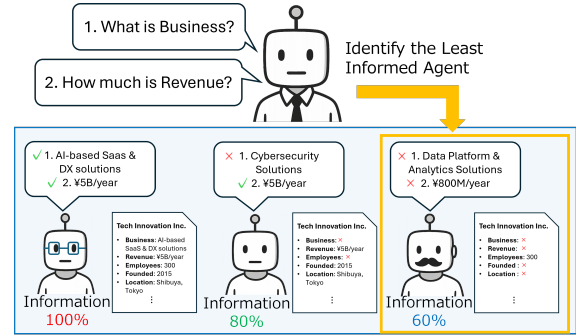


Figure 1: Our evaluation framework, InterviewArena, treats missing information as a single factor and identifies the least informative agent through iterative questioning based on responses and dialogue history.

frameworks use game-based interactions to evaluate multiple aspects of LLM behavior, including reasoning, cooperation, and communication.

In real-world scenarios, we often judge which participants possess more reliable information and infer missing information accordingly. For example, in police interviews, investigators must determine who has the most reliable knowledge while sequentially identifying individuals who may be providing misleading or false statements. Similarly, in human resource decision-making, organizations seek not only capable candidates but also individuals who fit organizational culture, requiring multi-faceted questioning to assess credibility and trustworthiness. These examples highlight that reasoning under incomplete information is frequently required in real-world decision-making, making it a suitable paradigm for evaluating more human-like reasoning capabilities in LLMs (Li et al., 2024; Xu et al., 2025; Liu et al., 2024). In such situations, it is necessary to infer who is the most knowledgeable and trustworthy participant based on dialogue histories. To deploy LLMs in more realistic environments, it is therefore important to evaluate their ability to discriminate among participants through simulation-based settings in the wild.

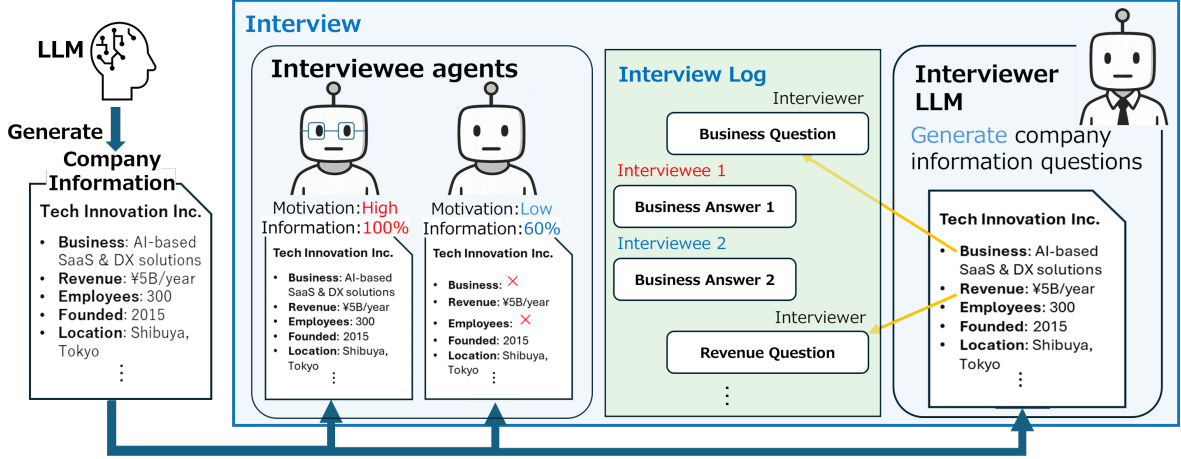


Figure 2: Overview of InterviewArena. In STEP 1, we yield synthetic company profile items, and interviewee agents are constructed by randomly omitting a subset of these items. To control for factors other than missing information, all agents are instantiated using the same LLM. Next, the target LLM is assigned as the interviewer, and STEP 2 and STEP 3 are iteratively performed to identify agents with missing information, thereby simulating an interactive interview process. This setup is designed to reflect a real-world scenario, specifically the Japanese new graduate hiring process, in which companies conduct multi-round interviews to assess candidates. The framework is inspired by the practical objective of recruiting applicants who demonstrate a strong interest in and understanding of the company, focusing on identifying less informative participants. When the LLM decides to terminate the interaction, the process proceeds to STEP 4, where identifying the most suspicious agent(s) based on the accumulated dialogue.

In this study, we propose InterviewArena, an evaluation framework inspired by real-world interview scenarios, as shown in Figure 1. In our framework, the target LLM is assigned the role of the interviewer and is required to estimate the level of interest in the company of candidate LLMs, enabling a quantitative evaluation of reasoning under incomplete information. Specifically, we construct synthetic company profiles with multiple attributes and interpret the degree of missing knowledge about the company of each candidate. The interviewer LLM interacts with candidate LLMs through dialogue and infers their information completeness, and the accuracy of these inferences is used to evaluate the model’s reasoning ability. Experimental results show that, for many models, the accuracy of identifying poorly informed participants and ranking candidates improves as interview rounds progress. However, in the task of predicting specific missing information that requires alignment with ground-truth company attributes, substantial performance differences are observed across models. These findings suggest that the ability to identify missing information and the ability to perform holistic reasoning based on dialogue are distinct skills.

2 InterviewArena

Figure 2 illustrates the overall evaluation framework of InterviewArena. The framework consists

of four main steps: Setting Agents as Interviewees, Interview Question Generation, Interviewee Response Generation, and Evaluation of Estimation Skills. This framework enables systematic observation and evaluation of how LLMs discriminate informative participants based on their responses.

STEP 1: Setting Agents as Interviewees

We construct synthetic company profiles as structured data. In this paper, company information comprises the following 11 categories: year of establishment, capital, number of employees, location, business description, corporate vision, company news, future outlook, partnerships, company strengths, and desired candidate profile. We prepare three agents and instantiate them as interviewees to simulate a multi-agent dialogue environment. Each agent is assigned a different level of missing information (low, medium, or high), with three, one, or zero company information items randomly omitted, respectively. We employ GPT-4o-mini (OpenAI et al., 2024a) to generate both the company information and the interviewee agents in this step.

STEP 2: Interview Question Generation

Next, we simulate a mock interview. The LLM under evaluation is assigned the role of the interviewer, while the candidate agents act as interviewees. The interviewer LLM has access to the complete company profile constructed in

STEP 1. During the interview, the interviewer sequentially asks questions corresponding to each company information item in order to determine which applicant possesses complete knowledge of the company. The interview continues until all information items have been queried, unless the LLM decides to terminate the interview earlier.

STEP 3: Interviewee Response Generation

Each interviewee generates a response to the question posed in STEP 2 based on the company information available to them and the dialogue history. Interviewee agents with missing information are encouraged to generate responses by plausibly inferring the missing items while maintaining consistency with the known information. During the interview, each interviewee has access to the full dialogue history up to the previous round but does not have access to the hidden company information assigned to other agents. At each round, we collect responses from all interviewees, and the dialogue history is accumulated in interview logs comprising the interviewer’s query and the corresponding responses from all applicants. The interview proceeds by repeating STEP 2 and STEP 3 until the termination condition is met.

STEP 4: Evaluation of Estimation Skills Finally, the target LLM produces predictions based on the full dialogue history accumulated during the interview. Specifically, the model estimates which interviewee agent possesses the most complete company information. Optionally, the model may also provide a rationale for its prediction and identify which company information items were missing for each interviewee.

3 Experiments

We evaluate LLMs from three perspectives: Participant Ranking Capability (RQ1), Identification of Low-Informative Participants (RQ2), and Prediction of Missing Information for Each Interviewee (RQ3), providing a comprehensive multi-dimensional evaluation. RQ1 assesses the basic ability to estimate the granularity of missing information by ranking participants considering information completeness. RQ2 evaluates whether the model can identify the least informative participant, thereby filtering out the worst-case candidate in an interview scenario similar to a werewolf game. RQ3 examines which company information items are missing for each interviewee agent to enable

	RQ1: Pair	RQ1: Exact	RQ2	RQ3
Llama-3.1-8B-Instruct	0.70	0.40	0.55	0.32
Llama-3-ELYZA-JP-8B	0.60	0.31	0.53	0.08
Llama-3.1-Swallow-8B	0.68	0.38	0.63	0.32
Qwen3-8b	0.57	0.25	0.32	0.34
Qwen2.5-7B-Instruct	0.69	0.31	0.55	0.42
GPT-4o mini	0.72	0.43	0.71	0.68
GPT-4	0.65	0.34	0.67	0.81
GPT-5.2	0.92	0.55	0.80	0.87
Gemini-2.0-flash-lite	0.85	0.57	0.80	0.72
Gemini-2.0-flash	0.90	0.61	0.74	0.69
Gemini-2.5-flash-lite	0.83	0.52	0.77	0.80
Gemini-2.5-flash	0.89	0.62	0.68	0.57

Table 1: Average scores in the final round for each RQ. The highest score in each evaluation is shown in bold.

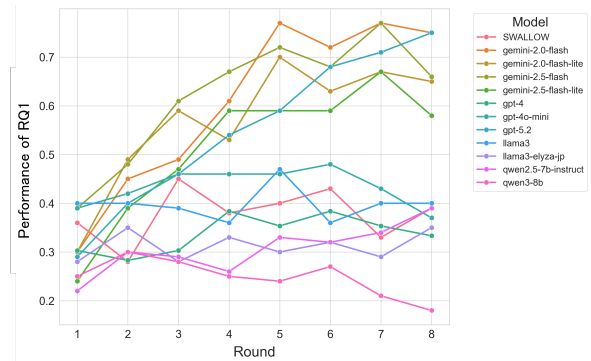


Figure 3: Performance over dialogue rounds for RQ1.

interpretability of the model’s reasoning behavior. In this study, we conduct 100 simulations, each with distinct company settings. We use total of 12 LLMs: Llama-3.1-8B-Instruct (Weerawardhena et al., 2025), Llama-3-ELYZA-JP-8B (Hirakawa et al., 2024), Llama-3.1-Swallow-8B (Fujii et al., 2024), Qwen3-8b (Yang et al., 2025), Qwen2.5-7B-Instruct (Qwen et al., 2025), GPT-4o mini (OpenAI et al., 2024a), GPT-4 (OpenAI et al., 2024b), GPT-5.2 (Singh et al., 2025), Gemini-2.0-flash-lite, Gemini-2.0-flash (Google DeepMind, 2024), Gemini-2.5-flash-lite, and Gemini-2.5-flash (Comanici et al., 2025). Detailed experimental settings are provided in Appendix A.

3.1 RQ1: Participant Ranking Capability

Setup. We calculate pairwise ranking accuracy and exact match accuracy. Pairwise ranking accuracy is defined as the proportion of pairs for which the predicted ranking agrees with the ground-truth ranking, i.e., the pairwise agreement rate. Exact match accuracy counts cases where the predicted ranking perfectly matches the ground-truth ranking as 1, and all others as 0. We instruct the LLMs to output the ranking in descending order.

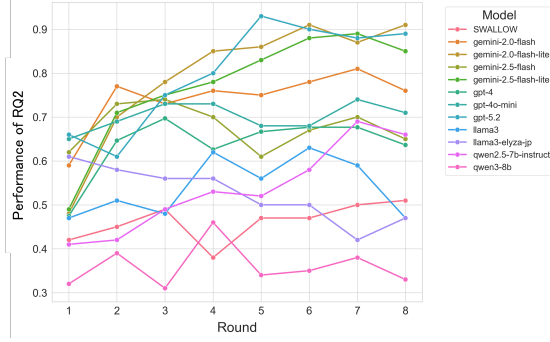


Figure 4: Performance over dialogue rounds for RQ2.

Results. As shown in Table 1, the pairwise accuracy results reveal two distinct groups of models: those achieving scores above 0.8 and those below 0.7. In terms of exact-match accuracy, the models are split into two groups, with scores above or below 0.5. The round-by-round results shown in Figure 3 further indicate that some models improve their identification accuracy as the interview rounds progress, whereas others show little change across rounds. However, even the best models achieve an exact-match accuracy of 0.6, indicating that correctly identifying the ranking remains challenging.

3.2 RQ2: Low-Informative Participants

Setup. We investigate whether the LLM can identify the participant with the least amount of information. The output is the name of the predicted candidate, which is evaluated as 1 if the correct candidate is identified and 0 otherwise. The model is instructed to output only the candidate’s name.

Results. Table 1 shows that, except for Qwen3-8B, more than half of the cases can be correctly predicted by the models. However, compared with the pairwise ranking results in RQ1, most models achieve lower scores on this task. This suggests that identifying the least informative participant remains challenging and has a significant impact on overall performance. As shown in Figure 4, models with higher overall performance are able to identify the correct participant with an accuracy above 0.5 even from the first round. Moreover, in these high-performing models, prediction accuracy generally increases as the interview rounds progress.

3.3 RQ3: Prediction of Missing Information

Setup. We investigate whether the interviewer LLM can ask appropriate questions to identify the missing information of each agent. Specifically, we measure the accumulated accuracy of questions

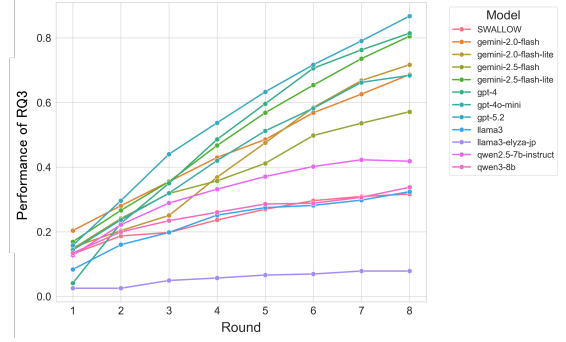


Figure 5: Performance over dialogue rounds for RQ3.

that correctly correspond to the missing information. The LLM is provided with the correct company information and the interview history, and is instructed to infer missing information only when responses are incorrect, ambiguous, or insufficient. The output reports the predicted missing information along with its rationale in a single line.

Results. As shown in Table 1, the models can be broadly divided into two groups: those achieving scores above 0.7 and those below 0.5. Notably, Gemini-2.5-flash, which performed well in RQ1, achieves only 0.57 in this setting, indicating that it does not effectively generate questions targeting missing information. In contrast, GPT-4 produces the most queries related to missing information; however, its performance in RQ1 and RQ2 is not as strong. These results suggest a discrepancy between the ability to generate informative queries and overall reasoning performance. Furthermore, as shown in Figure 5, models generally increase the number of questions targeting missing information as the number of rounds progresses. These findings indicate that the ability to identify missing information and to perform holistic reasoning based on dialogue are distinct skills.

4 Conclusion

In this study, we introduce InterviewArena, which is an interview-style Werewolf-like game designed to reflect realistic settings. By focusing on a single factor, the degree of missing information, we disentangle the ability to infer missing information from the ability to perform dialogue-based reasoning. Results show that LLMs can capture relative differences between candidates by leveraging consistency and ambiguity in their responses. However, estimating missing corporate information by comparing predictions with ground truth

remains unstable. These findings highlight the importance of evaluating reasoning under incomplete information for a more realistic assessment.

Limitations

LLMs. While evaluating a broader range of LLMs would be desirable, we limit our experiments due to the scope of a short paper and focus on verifying that the proposed framework functions as intended. As shown in Section 3, we observe consistent differences in behavior across models, indicating that the framework captures the intended evaluation aspects. Based on these results, we consider the current set of LLMs sufficient for validating the effectiveness of the framework. Nevertheless, extending the evaluation to a wider variety of models remains an important direction.

Agents. We use a single LLM to construct interviewee agents in our experiments. Nevertheless, as shown in Section 3, even with a single LLM, the framework captures meaningful behavioral differences, indicating that it functions as intended. Moreover, using the same LLM for both interviewer and interviewee does not hinder the evaluation, suggesting that the experimental setup successfully separates condition design from model evaluation. We also expect that using more advanced LLMs for agent construction would increase the difficulty of the task. However, given the scope of a short paper and the sufficient empirical results obtained, we do not explore additional LLMs for agent construction in this paper.

Generalization. In STEP 1, we use an LLM to generate synthetic company profiles. This step serves only to create initial seed data, and subsequent processes systematically introduce missing information. Therefore, this step is not tied to a specific model, and alternative models could be used without affecting the core evaluation. Moreover, the framework is not restricted to company-related scenarios. By modifying the profile generation process, it can be readily applied to other domains. In this work, we focus on a typical and practically relevant interview scenario, which we selected heuristically to ensure a clear and controlled evaluation setting. While evaluating the framework under a wider range of scenarios would provide a more comprehensive assessment of its generalizability, we leave such extensions for future work in order to maintain a focused scope.

Experimental setting. Our experiments focus on verifying that the proposed framework functions as intended and can effectively measure the target capabilities of LLMs. Accordingly, we conducted experiments with three interviewee agents. While increasing the number of agents could introduce more complex interaction dynamics, it would also substantially increase the length of the dialogue history, potentially exceeding the input context limits of current models. Moreover, as shown in Section 3, even with three agents, the framework yields meaningful and consistent findings. Therefore, we do not explore larger-scale settings in this work.

Ethical Considerations

Potential risks. This study relies on synthetically generated profiles, which may raise concerns about unintended overlap with real-world entities. To mitigate this risk, we manually inspected all generated profiles and did not observe any content that closely resembles real companies. This is consistent with our generation process, in which we explicitly instruct the model to produce fictional but plausible profiles. We also examined the dialogue data and did not observe any outputs containing harmful or unethical content, such as violence or other policy violations. This is likely due to the safety training of modern LLMs, which are designed to avoid generating sensitive or harmful content. Therefore, we ensure that our data generation and experimental setup comply with the ACL Code of Ethics.

AI Tools. We used AI tools to assist with grammatical correction and translation, including Grammarly, DeepL, and ChatGPT. The original content of this paper was written entirely by the authors, and AI tools were used solely for language refinement. All research ideas, experimental design, and substantive writing were carried out by the authors.

Others. All LLMs and experimental tools used in this study are permitted for research use. To ensure reproducibility, we plan to release our code, scripts, and all raw experimental results upon acceptance.

References

- Suma Bailis, Jane Friedhoff, and Feiyang Chen. 2024. [Werewolf arena: A case study in llm evaluation via social deduction](#). *Preprint*, arXiv:2407.13943.
- Kranti Chalamalasetti, Jana Götze, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2023. [clembench: Using game play to](#)

- evaluate chat-optimized language models as conversational agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11174–11219, Singapore. Association for Computational Linguistics.
- Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. [Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities](#). *Preprint*, arXiv:2507.06261.
- Anthony Costarelli, Mat Allen, Roman Hauksson, Grace Sodunke, Suhas Hariharan, Carlson Cheng, Wenjie Li, Joshua Clymer, and Arjun Yadav. 2024. [Gamebench: Evaluating strategic reasoning abilities of llm agents](#). *Preprint*, arXiv:2406.06613.
- Kazuki Fujii, Taishi Nakamura, Mengsay Loem, Hiroki Iida, Masanari Ohi, Kakeru Hattori, Hirai Shota, Sakae Mizuki, Rio Yokota, and Naoaki Okazaki. 2024. [Continual pre-training for cross-lingual LLM adaptation: Enhancing japanese language capabilities](#). In *First Conference on Language Modeling*.
- Google DeepMind. 2024. [Introducing gemini 2.0: our new ai model for the agentic era](#). (Accessed: 9 Aug. 2026).
- Masato Hirakawa, Shintaro Horie, Tomoaki Nakamura, Daisuke Oba, Sam Passaglia, and Akira Sasaki. 2024. [elyza/llama-3-elyza-jp-8b](#). (Accessed: 9 Aug. 2026).
- Guiyang Hou, Wenqi Zhang, Zhe Zheng, Yongliang Shen, and Weiming Lu. 2025. [Scaling LLMs’ social reasoning: Sprinkle cognitive “aha moment” into fundamental long-thought logical capabilities](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 3126–3138, Vienna, Austria. Association for Computational Linguistics.
- Hang Jiang, Xiajie Zhang, Xubo Cao, Cynthia Breazeal, Deb Roy, and Jad Kabbara. 2024. [PersonaLLM: Investigating the ability of large language models to express personality traits](#). In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 3605–3627, Mexico City, Mexico. Association for Computational Linguistics.
- Bolin Lai, Hongxin Zhang, Miao Liu, Aryan Pariani, Fiona Ryan, Wenqi Jia, Shirley Anugrah Hayati, James Rehg, and Diyi Yang. 2023. [Werewolf among us: Multimodal resources for modeling persuasion behaviors in social deduction games](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6570–6588, Toronto, Canada. Association for Computational Linguistics.
- Yifei Li, Xiang Yue, Zeyi Liao, and Huan Sun. 2024. [AttributionBench: How hard is automatic attribution evaluation?](#) In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14919–14935, Bangkok, Thailand. Association for Computational Linguistics.
- Wei Liu, Chenxi Wang, Yifei Wang, Zihao Xie, Rennai Qiu, Yufan Dang, Zhuoyun Du, Weize Chen, Cheng Yang, and Chen Qian. 2024. [Autonomous agents for collaborative task under information asymmetry](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 2734–2765. Curran Associates, Inc.
- Gabriel Mukobi, Hannah Erlebach, Niklas Lauffer, Lewis Hammond, Alan Chan, and Jesse Clifton. 2023. [Welfare diplomacy: Benchmarking language model cooperation](#). *Preprint*, arXiv:2310.08901.
- OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, and 401 others. 2024a. [Gpt-4o system card](#). *Preprint*, arXiv:2410.21276.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024b. [Gpt-4 technical report](#). *Preprint*, arXiv:2303.08774.
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. [Qwen2.5 technical report](#). *Preprint*, arXiv:2412.15115.
- Yusuke Sakai, Hidetaka Kamigaito, and Taro Watanabe. 2025. [Revisiting compositional generalization capability of large language models considering instruction following ability](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 31219–31238, Vienna, Austria. Association for Computational Linguistics.
- Takehiro Sato, Shintaro Ozaki, and Daisaku Yokoyama. 2024. [An implementation of werewolf agent that does not truly trust LLMs](#). In *Proceedings of the 2nd International AIWolfDial Workshop*, pages 58–67, Tokyo, Japan. Association for Computational Linguistics.
- Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, and 465 others. 2025. [Openai gpt-5 system card](#). *Preprint*, arXiv:2601.03267.

Zirui Song, Yuan Huang, Junchang Liu, Haozhe Luo, Chenxi Wang, Lang Gao, Zixiang Xu, Mingfei Han, Xiaojun Chang, and Xiuying Chen. 2025. [Beyond survival: Evaluating llms in social deduction games with human-aligned strategies](#). *Preprint*, arXiv:2510.11389.

Sajana Weerawardhena, Paul Kassianik, Blaine Nelson, Baturay Saglam, Anu Vellore, Aman Priyanshu, Supriti Vijay, Massimo Aufiero, Arthur Goldblatt, Fraser Burch, Ed Li, Jianliang He, Dhruv Kedia, Kojin Oshiba, Zhouan Yang, Yaron Singer, and Amin Karbasi. 2025. [Llama-3.1-foundationai-securityllm-8b-instruct technical report](#). *Preprint*, arXiv:2508.01059.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. [Transformers: State-of-the-art natural language processing](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zixiang Xu, Yanbo Wang, Yue Huang, Haomin Zhuang, Yujun Zhou, Jiayi Ye, Zirui Song, Lang Gao, Chenxi Wang, Zhaorun Chen, Sixian Li, Wang Pan, Yue Zhao, Xiangliang Zhang, and Xiuying Chen. 2025. [Socialmaze: A benchmark for evaluating social reasoning in large language models](#). In *Socially Responsible and Trustworthy Foundation Models at NeurIPS 2025*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.

Dora Zhao, Qianou Ma, Xinran Zhao, Chenglei Si, Chenyang Yang, Ryan Louie, Ehud Reiter, Diyi Yang, and Tongshuang Wu. 2025. [SPHERE: An evaluation card for human-AI systems](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1340–1365, Vienna, Austria. Association for Computational Linguistics.

A Detailed Experimental Setup

The input prompt for the interviewer LLM is shown in Figure 6, and the prompt for the interviewee LLM is shown in Figure 7. The response guidelines for each level of interest are presented in Figure 8. Figures 9, 10, and 11 show the prompts used for RQ1, RQ2, and RQ3, respectively. Although all prompts were originally written in Japanese, we

Prompt for the interviewer to ask questions

SYSTEM PROMPT:

You are a highly capable and insightful recruitment analyst who accurately follows the given instructions in Japanese.

USER PROMPT:

You will now ask the same common question to all candidates.

List of Company Information You Can Ask About

....

All Questions Asked So Far (Avoid Duplication)

....

Strategic Role of Common Questions Common questions are used for the following purposes:

Establishing a Basis for Comparison: Since all candidates respond under the same conditions, fair comparison is possible.

Collecting Foundational Information: Measure the basic level of company understanding necessary to differentiate candidates.

Efficient Information Gathering: Collect information from all candidates with a single question, saving time.

Deepening Shared Topics: Compare all candidates’ perspectives on specific important company-related topics.

Strategic Situations for Using Common Questions

When there is insufficient material for comparing candidates

When you want to assess all candidates’ understanding of specific key company information

When foundational information is needed before moving to deeper individual questions

When you want to gather information efficiently

Instructions

Overall Analysis: Take a holistic view of all candidate conversations and identify “commonly unaddressed items” that most candidates have not sufficiently mentioned.

Strategic Question Generation: From the identified items, select the one that is most important for comparing the depth of candidates’ company research, and generate a specific common question about it.

Ensure Comparability: Ensure that the question allows all candidates to answer based on the same criteria.

Output only the question. Do not include any thought process or preamble.

Figure 6: Prompt for the interviewer to ask questions.

provide English translations for readability. Information about the LLMs used in the experiments is provided in Table 2. Unless otherwise specified, all implementations are based on Hugging Face Transformers (Wolf et al., 2020). For API-based models, we use their corresponding official APIs.

LLMs	HuggingFace ID / API Name
Llama-3.1-8B-Instruct	meta-llama/Llama-3.1-8B-Instruct
Llama-3-ELYZA-JP-8B	elyza/Llama-3-ELYZA-JP-8B
Llama-3.1-Swallow-8B	tokyotech-llm/Llama-3.1-Swallow-8B-Instruct-v0.5
Qwen3-8b	Qwen/Qwen3-8B
Qwen2.5-7B-Instruct	Qwen/Qwen2.5-7B-Instruct
GPT-4o mini	gpt-4o-mini-2024-07-18
GPT-4	gpt-4-0613
GPT-5.2	gpt-5.2-2025-12-11
Gemini-2.0-flash-lite	gemini-2.0-flash-lite
Gemini-2.0-flash	gemini-2.0-flash
Gemini-2.5-flash-lite	gemini-2.5-flash-lite
Gemini-2.5-flash	gemini-2.5-flash

Table 2: Information about HuggingFace IDs and APIs.

Prompt for the interviewee to answer

SYSTEM PROMPT: You are a highly capable and insightful recruitment analyst who accurately follows the given instructions in Japanese.

USER PROMPT:
Your Role You are a job-seeking student
Please convey your admiration for this company and your strong motivation to join it to the interviewer.

Your Profile
Name:
Strengths:
Key Experience (Gakuchika):

Company Information You Know
....

Basic Answering Policy
....

Conversation History
....

Current Question from the Interviewer
....

Instruction: Act as based on the above settings and behave as a job-seeking student. Output your answer in natural Japanese of approximately characters. Do not include any preface or explanation.

Figure 7: Prompt for the interviewee to answer.

Response guidelines by level of interest

BASIC ANSWERING POLICY:

High Preparation Level Demonstrate a very high level of interest and enthusiasm. When information is unknown, supplement it with positive and reasonable assumptions, and never state that you do not know.

Medium Preparation Level Demonstrate a high level of interest. Clearly describe known information with specific details, and supplement unknown information with general industry knowledge.

Low Preparation Level While expressing interest in the company, respond in a way that conceals gaps in your company knowledge.

Figure 8: Response guidelines by level of interest.

Generating responses for RQ1 evaluation

USER PROMPT:
Analyze the following interview responses of candidates and rank them in order of lowest to highest level of interest in the company.

Interview Content
....

List of Candidate Names
....

Ranking Criteria
* Depth of company knowledge * Specificity of responses * Expression of motivation and enthusiasm

Instruction: Rank the candidates concisely by name in order of lowest interest (Rank 1 = lowest level of interest).

Figure 9: Prompt for RQ2 evaluation.

Generating responses for RQ2 evaluation

USER PROMPT:
Analyze the following interview responses of candidates and identify the candidate who appears to have the lowest level of interest in the company.

Interview Content
....

List of Candidate Names
....

Analysis Criteria
* Depth of company knowledge * Specificity of responses * Expression of motivation and enthusiasm

Instruction: Output only the name of the candidate who is judged to have the lowest level of interest.

Figure 10: Prompt for RQ2 evaluation.

Generating responses for RQ3 evaluation

USER PROMPT:
Compare the following ground-truth company information with each candidate's interview record and identify gaps in the candidates' knowledge.

Important Note Do not conclude that a candidate lacks knowledge merely because they did not mention a particular point. A knowledge gap should be identified only when there was a natural opportunity to mention that information in the flow of the interview, yet the candidate either failed to mention it, provided incorrect information, or answered ambiguously.

Ground-Truth Company Information (Key-Value Pairs)
....

Interview Records of Each Candidate
....

Figure 11: Prompt for RQ3 evaluation.

Protocol Competence of Small Language Models in Interactive Language Game Environments

Suchir Salhan   * & Paula Buttery  

 Department of Computer Science & Technology, University of Cambridge, U.K.

 ALTA Institute, Cambridge, U.K.

Abstract

Learning from interaction requires more than producing appropriate responses: learners must maintain a shared interaction with a partner, following its rules while adapting to feedback and contingencies. We argue that this depends on a prerequisite we call **protocol competence**: the ability to produce valid actions, maintain interaction structure, and respond appropriately across turns. In this paper, we investigate protocol competence in Small Language Models (SLMs) from the Qwen family across five interactive dialogue games from CLEMBENCH, finding 51.3% of episodes from Qwen3.5-0.8B terminate in protocol violations. However, teacher demonstrations from a larger language model can scaffold participation, raising Wordle play from 0/30 to 29/30. Once models can reliably participate, we find self-improving reinforcement learning becomes effective: a single round of self-generated reinforcement learning (self-ReST) yields large gains from a small number of successful trajectories. Comparing adaptive learning strategies, we find that credit-weighted training provides the most reliable improvements, reducing variance relative to uniform supervision. Together, these results suggest that interactive learning follows a staged trajectory: SLMs must first acquire the competence to participate before they can exploit the contingencies generated by interaction.



Playpen Models on [HuggingFace](#)



Code Open-Sourced on [GitHub](#)

1 Introduction

Learning from interaction requires more than producing an appropriate response: a learner must build and update a shared understanding with its partner across turns, using feedback, correction, and repair to coordinate action and reference (Clark and Wilkes-Gibbs, 1986). Computational models

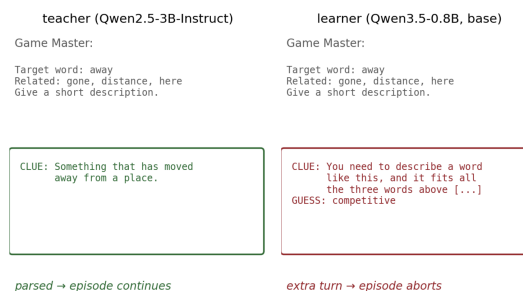


Figure 1: **Protocol competence in the Taboo Game (Horst et al., 2025b).** *Left:* the Teacher Qwen model produces a valid clue in the format expected by the game, allowing the interaction to continue and the clue to be evaluated. *Right:* the Learner Qwen SLM produces a plausible clue but then generates the other player’s turn as well: the task is not the problem, **knowing when to stop is**. This is a **protocol-competence failure**: the model can produce a valid clue, but cannot yet reliably participate in the interaction.

have learned from environmental reward, situated instruction, observation, human corrections, and co-players in language-mediated games (Branavan et al., 2009; Chen and Mooney, 2011; Wang et al., 2016, 2017; Hancock et al., 2019; Kojima et al., 2021; Lewis et al., 2017; He et al., 2018; Lazaridou et al., 2017). Across these settings, however, interaction is usually treated as available to the learner: the agent produces an action or utterance that the environment or partner can interpret, evaluate, and respond to.

For language models, this assumption is consequential. In open-ended dialogue, a learner must first produce an utterance that conforms to the protocol well enough for interaction to continue (Bertolazzi et al., 2023; Qiao et al., 2023; Guertler et al., 2025). Dialogue games provide a clean setting in which to study this problem because they distinguish failure to produce a valid turn from failure to achieve the task (Schlangen, 2023). A model that gives a poor clue can receive feedback about

Corresponding Author: sas245@cam.ac.uk

task performance; a model whose output cannot be parsed may never reach the point at which such feedback is available.

We therefore argue that learning from interaction has a prerequisite: **protocol competence**. This is the ability to enter an interaction in the required format, produce valid moves, maintain the exchange structure across turns, and use feedback to condition future behaviour. Protocol competence is distinct from task competence: a model can participate reliably yet fail to win, or possess relevant knowledge yet fail to participate long enough to access the learning signal. We distinguish the production of a valid response in the appropriate context and format from *interactive task competence*, which uses interaction to improve task performance. More generally, contingent interaction requires the learner to recognise and exploit structured dependencies between its own actions and another agent’s responses (Salhan et al., 2025). Without valid participation, these action–response pairs never arise; the interaction loop closes before it opens. This motivates our central question: **what must a small language model (SLM) learn before it can learn from interaction?** We study Qwen models (Yang et al., 2025) in the Playpen framework (Horst et al., 2025b) across five protocol-based language games drawn from the clemench suite (Chalamalasetti et al., 2023).

Our results reveal a substantial participation bottleneck. A Qwen3.5-0.8B base learner aborts more than half of sampled episodes because it violates the interaction protocol before a task outcome can be assigned. We therefore first scaffold the learner with demonstrations from a larger Qwen2.5-3B-Instruct teacher, and only then expose it to its own successful interactive experience through reinforcement-learning-based methods including ReST, online GRPO, and GiGPO. Across these settings, improved protocol competence is an important prerequisite for interactive improvement: models can make progress when they reliably participate, while games with persistent protocol failures remain resistant to further learning.

The bottleneck is not explained by model scale alone. Protocol competence is not monotonic with parameter count across Qwen models, with smaller models sometimes outperforming larger variants, and it is not removed by instruction tuning or by the choice of teacher. These results suggest that successful interactive learning cannot be reduced

to a scaling effect: **before a model can learn from interaction, it may first need to learn how to participate in interaction.**

2 Methodology

Our central hypothesis is that successful interactive learning has an enabling condition we call *protocol competence*: the ability to produce valid actions in the required format, at the appropriate turn, while maintaining the interaction structure. Contingency and feedback use are what happen after this protocol channel is established. Without this competence, a model cannot obtain meaningful learning signals regardless of its underlying reasoning ability. Once protocol competence is established, subsequent improvement should depend primarily on ordinary experience-driven learning rather than continued teacher supervision.

We investigate **what an SLM must learn before it can learn from interaction**. Our starting point is *protocol competence*: the ability to produce valid actions, follow the rules of an interaction, and respond appropriately to feedback. We first ask *can demonstrations from a larger model teach an SLM how to participate?* (RQ_1). We train the SLM on teacher demonstrations and separately measure whether it becomes better at producing valid interactions and whether this translates into better task performance. We then ask *once an SLM can participate, can it learn from its own experience without a teacher?* (RQ_2). Following ReST (Gulcehre et al., 2023a), the model generates its own trajectories, keeps successful interactions, and trains on them. Finally, we ask *how much successful experience is needed for this self-learning to work?* (RQ_3), varying the number of training trajectories from 64 to 512. These experiments establish whether protocol competence is simply another aspect of task performance, or whether it is a prerequisite for learning from interaction.

We next ask *what makes self-generated experience useful for further learning*. Once the learner can reliably participate, we compare standard ReST with methods that select or weight experience according to how useful it is (§2.3). We test whether learning improves when the model focuses on interactions that are challenging but still within its capabilities (*ZPD-based selection*), gives greater weight to trajectories likely to cause behavioural change (*credit-weighted learning*), or explores more during generation through different decoding strategies.

We also compare offline self-training with online methods, GRPO and GiGPO (Shao et al., 2024; Feng et al., 2025), which assign credit while the interaction is taking place. Finally, we test whether the effect can be explained by simpler alternatives such as model size, instruction tuning, supervision quantity, demonstration type, teacher identity, or decoding (§2.3), and whether protocol competence transfers to games that were not used for training. This allows us to distinguish *learning to perform a task* from *learning how to participate in the interaction through which the task can be learned*.

2.1 Interactive Games

We evaluate on five dialogue games from Chalamalasetti et al. (2023). In each game, a Game Master parses every turn and a protocol violation terminates the episode immediately, which is what makes participation separately measurable. These have different forms of interactive reasoning while enforcing strict communication protocols. Taboo requires constrained paraphrasing under alternating CLUE:/GUESS: turns; Wordle requires sequential constraint reasoning over feedback from previous guesses; ImageGame involves collaborative reconstruction of a spatial grid through incremental natural-language instructions; ReferenceGame tests discriminative referring expressions over visually similar alternatives; and PrivateShared evaluates maintenance of common ground during multi-turn dialogue. In every game, protocol violations immediately terminate the episode, allowing participation failures to be distinguished from task failures (see Appendix A).

2.2 Experimental Setup

Models We use a base model QWEN3.5-0.8B (Yang et al., 2025) as a student model, and experiment with larger base models spanning 0.8B–7B parameters from the Qwen3 family. We use several teacher models that provide **demonstrations**. Demonstrations are complete, protocol-conformant episode transcript produced by a larger model in the same game that is rendered in the learner’s chat format and used as a supervised target on the assistant turns only.

Demonstration Settings We experiment with two settings for demonstrations: **protocol-only demonstrations** preserve the interaction structure (which template to emit, what a turn looks like, what must not appear), whereas **reasoning-only**

demonstrations provide task solutions without reliably establishing the interaction channel (which move to choose).

Metrics We use three metrics to evaluate gameplay:

Play-Rate Play-rate is a binary measure: an episode either reaches a verdict or it aborts. It is the fraction of episodes that reach a task verdict rather than aborting ($\frac{\text{wins} + \text{losses}}{\text{episodes}}$), as our primary operational measure of *protocol competence*. This is whether a model can produce an interaction that a game can accept (ranging from 0 – 1). We can also define format-abort rate as $1 - \text{play-rate}$. We report play-rate at two different levels of aggregation. Macro play-rate first computes play-rate separately for each game and then averages across games. Pooled play-rate instead combines all episodes across games before computing the overall rate. These can give different results because the games contain different numbers of episodes. For example, with sampled rollouts ($(T=0.9, k=6)$), the base learner has a macro abort rate of 0.67 but a pooled abort rate of 0.51.

Per-turn violation rate To examine the underlying behaviour of playrate more continuously, we also report per-turn violation rate. This measures how often the model produces an invalid turn, rather than whether the entire episode eventually aborts. We report this measure separately by game because pooling across games makes it strongly dependent on how many turns each game contains.

Quality This is a *game-specific score*, calculated only over episodes that are actually played. Game-specific quality measures task success conditional on valid participation: Taboo rewards identifying the target with fewer guesses, Wordle rewards solving the word using fewer attempts, ImageGame measures accurate reconstruction of the target grid, ReferenceGame measures successful identification of the intended referent, and PrivateShared measures successful maintenance and use of common ground across turns (Chalamalasetti et al., 2023; Beyer et al., 2024).

CLEMSCORE CLEMSCORE is the product of playrate and quality (Chalamalasetti et al., 2023; Beyer et al., 2024; Horst et al., 2025a). Aborted episodes therefore count against play-rate but do not enter the quality calculation. This means that

a model can improve its quality while its CLEM-Score falls if it becomes more likely to abort.

2.3 Self-Improvement Reinforcement Learning

We next ask once an SLM has acquired protocol competence, does the way that experience is selected, explored, or credit-assigned affect subsequent learning? This is a stronger test of learning from interaction than teacher-supervised protocol repair: the model must first generate trajectories that successfully navigate the environment and then extract useful supervision from those trajectories. We draw inspiration from ReST (Reinforced Self-Training), an offline self-improvement method in which a model generates its own trajectories, selects successful ones using outcome feedback, and fine-tunes on them to improve its policy (Gulcehre et al., 2023b). Our methodological contribution is to adapt offline ReST (Gulcehre et al., 2023b) to interactive dialogue games and systematically compare flat rejection sampling, difficulty-based (ZPD) selection, step- and episode-level credit weighting, and decoding-based exploration under matched compute.

Flat-ReST We call our primary condition **Flat ReST**, where every retained winning trajectory is treated equally during fine-tuning. This setup therefore asks a simple question: once an SLM knows how to participate, can it improve its task performance by learning from interactions that it successfully generated itself? We therefore implement self-ReST, following ReST, as an offline *Grow-Select-Improve* procedure. We start self-ReST from a protocol-repaired checkpoint of the base SLM, so that the model can already participate correctly in the games. Self-ReST then repeatedly follows three steps. First, in *Grow*, the current model plays each game and generates $k = 4$ candidate trajectories per game instance at temperature 0.8 across the five-game Playpen suite. Second, in *Select*, we keep only trajectories in which the model wins and has produced a valid interaction: specifically, the trajectory must contain a valid assistant interaction turn and at least two messages. The resulting winners are balanced across games. Third, in *Improve*, we fine-tune the model on these successful trajectories. Importantly, the model is not updated while generating the trajectories: we first collect and filter a fixed set of experiences, and only then train on them. We repeat this process

across rounds, allowing the improved model to generate new experiences in the next round. We also vary the amount of self-generated supervision independently of the number of rounds by training on different numbers of winning trajectories. Let $\mathcal{W}$ denote the valid winning trajectories and $\mathcal{S} \subseteq \mathcal{W}$ the selected training set; the *Improve* stage minimizes $\mathcal{L}(\theta) = -\sum_{\tau \in \mathcal{S}} \sum_t w(\tau, t) \log \pi_\theta(a_t | s_t)$. **Flat ReST** retains all winners with $w(\tau, t) = 1$.

Alternative Strategies We introduce three additional strategies that differ in *where they place the learning signal*. Unlike **Flat ReST**’s treatment of every successful trajectory as equally informative (a rejection-sampling baseline), we compare concentrating experience where the learner is near competence, and assigning greater credit to the episodes or actions that provide stronger evidence of progress.

- **ZPD selection** assumes that the most useful experience lies near the learner’s current capability boundary: interactions that are neither consistently solved nor consistently failed provide opportunities to learn from achievable successes. **ZPD selection** retains winners only from instances with empirical win rate $W_i \in [0.25, 0.75]$, concentrating supervision on interactions where the current policy has demonstrated partial competence.
- **Credit-weighted ReST** goes further by distinguishing *which successful experiences* and *which decisions within them* were most informative, assigning greater weight to difficult episodes and locally successful actions. **Credit-weighted ReST** (offline GiGPO; Feng et al., 2025) retains the full winner set but sets $w(\tau_i, t) = A_i^{\text{epi}} + A_{i,t}^{\text{step}}$, increasing the contribution of difficult successful episodes and of steps that make progress relative to alternative trajectories from similar states.
- The **decoding control** instead changes how experience is generated rather than how it is selected, testing whether broader or more focused exploration during trajectory collection changes the quality of the subsequent learning signal. Our implementation varies sampling temperature and nucleus probability during *Grow* while keeping Flat ReST selection and weighting unchanged, isolating the role of exploration.

We compare these alternative selection, weighting, and exploration strategies under matched compute, filtering, and fine-tuning budgets. All variants use the same winner filtering, game balancing, optimization procedure, and compute budget, with one epoch of full-parameter supervised fine-tuning at learning rate 10^{-5} and batch size 16.

3 Experiments & Results

Our experiments systematically separate protocol competence (§3.1), the effect of demonstrations (§3.3) and self-improvement (§3.4 and §3.5).

3.1 SLM Protocol Competence is Environment-Dependent

Interactive environments require models not only to produce plausible actions but to express them in the precise format expected by the environment. Across 2,210 episodes, 1,133 (51.3%) terminate because of a protocol violation before a game verdict is reached, compared with 763 (34.5%) reasoning failures and only 314 (14.2%) successes (Table 1). An example of an aborted episode from the IMAGEGAME environment is shown below:

IMAGEGAME — ABORTED EPISODE (QWEN2.5-3B-INSTRUCT)

```

1 [A:GM]
  Let us do the following task. The
  goal is to fill an empty grid that
  looks like this:
  □ □ □ □ □
  □ □ □ □ □
  □ □ □ □ □
  □ □ □ □ □
  □ □ □ □ □
  A filled grid below is 5 by 5 and
  can look like this:
  □ □ □ □ □
  □ □ E □ □
  □ □ □ □ □
  □ □ □ □ □
  X X X [ . . . ]

2 [A:GM]
  Command: Put an F in second row
  fourth column
  What is your next command?

3 [GM:GM]
  Invalid instruction format

4 [GM:GM]
  Game aborted.

```

The player emits a well-formed command and then writes the co-player’s turn as well; the game master rejects the turn and the episode aborts.

The dominant failure is extra text, accounting for 33.8% of all episodes (see Figure 2 and Fig-

ure 3). In a sample of 287 violations from *Wordle* and *Taboo*, 189 contained a valid task move embedded in conversational prose that the parser could not accept, indicating that many apparent task failures are instead *valid action, invalid interface errors* (Appendix B). However, this bottleneck is strongly environment-dependent. Under sampled decoding, play-rate ranges from 0.96 in *ReferenceGame* to 0.00 in *PrivateShared*, with *Wordle* at only 0.04. Thus, interactive performance cannot be interpreted solely as task-solving ability: in many environments, models fail before the environment can provide the feedback required for learning. Full per-game failure categories and protocol profiles are reported in Appendix B, and the same protocol bottleneck extends beyond the five training games to the wider clembench suite, where 10 of 13 additional games are format-bound (Appendix I).

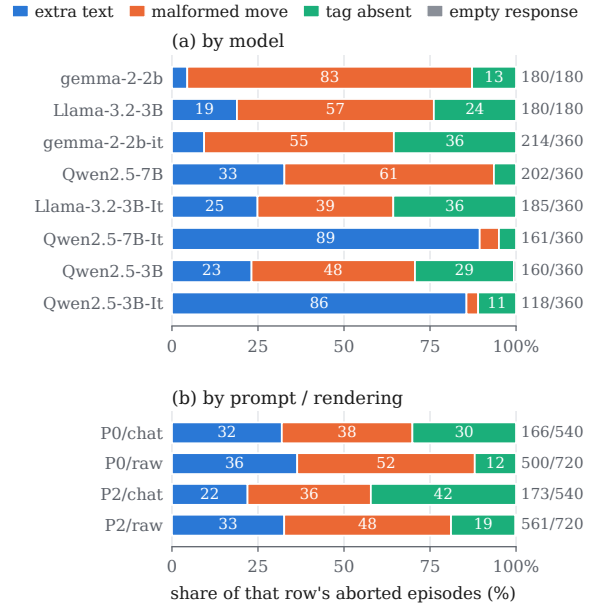


Figure 2: **Relative distribution of abort categories.** Distribution of categories across games (a) different teacher models and (b) different prompt strategies normalised to 100%

3.2 Protocol competence does not reliably emerge from scale

A natural explanation for the participation bottleneck is insufficient model capacity. However, interactive ability is not monotonic with scale (Table 2). Across five Qwen base models spanning 0.8B–7B parameters, play-rate varies substantially, with the 0.8B model achieving the highest deterministic participation rate (0.59), while several larger models collapse to zero. Instruction tuning improves par-

Table 1: **Protocol failure is a primary bottleneck for small base models.** Qwen3.5-0.8B evaluated on the five-game suite. Under sampled decoding ($k=6$, $T=0.9$), 1,133 of 2,210 episodes (51.3%) terminate in a protocol abort before a game verdict is reached. *Play-rate* is the fraction of episodes reaching a win or loss verdict. Deterministic decoding substantially improves participation, although Wordle and PrivateShared remain inaccessible.

Game	n	Abort	Loss	Win	Sampled	Deterministic
ReferenceGame	792	28	458	306	0.96	1.00
ImageGame	455	429	26	0	0.06	1.00
Taboo	483	203	272	8	0.58	0.97
Wordle	180	173	7	0	0.04	0.00
PrivateShared	300	300	0	0	0.00	0.00
Overall	2210	1133	763	314	0.49	0.59

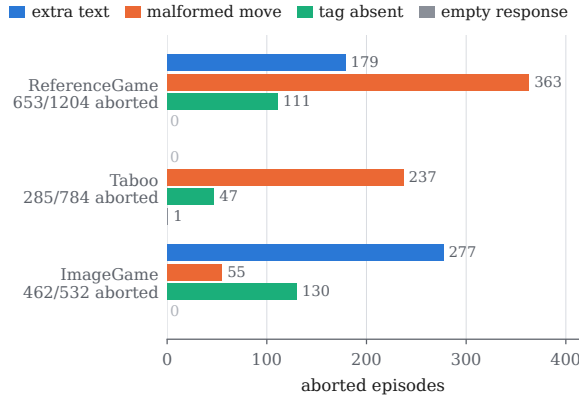


Figure 3: **Number of aborted episodes per abort category, by game.**

ticipation in matched models but does not eliminate protocol failures (Appendix D). Participation is also highly sensitive to prompt and output rendering, and this sensitivity extends to other model families such as Llama and Gemma (Appendix C). These results suggest that protocol competence is not a capability that reliably emerges from scale or generic instruction tuning alone.

Table 2: **Protocol competence is not monotonic with model scale.** Deterministic evaluation of Qwen base models. Larger models do not consistently achieve higher participation.

Model	Size	Play-rate	CLEMSCORE
Qwen3.5-0.8B	0.8B	0.59	6.67
Qwen2.5-1.5B	1.5B	0.00	0.00
Qwen2.5-3B	3B	0.38	7.04
Qwen3.5-4B	4B	0.00	0.00
Qwen2.5-7B	7B	0.17	0.00

3.3 Protocol competence and task competence are separable

Teacher demonstrations reveal a clear dissociation between learning *how to participate* and learning

how to solve a game (Table 3). Wordle provides a direct dissociation between the two: 400 task-specific demonstrations increase participation from 0/30 to 29/30 episodes, yet produce no wins. The model therefore learns the interaction format without acquiring the strategic competence required to solve the game.

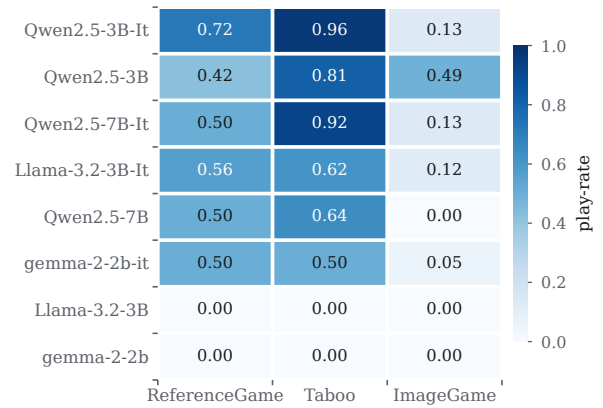


Figure 4: **Play-rate by Teacher Model and Playpen Game**

More generally, protocol-only supervision is substantially more effective than reasoning-only supervision at preserving valid interaction, while full demonstrations provide the largest downstream improvement. The full per-game demonstration analysis and supervision ablations are reported in Appendix E, and protocol repair is robust to the choice of teacher, including non-instruction-tuned teachers (Appendix H). These results establish that protocol competence is a distinct prerequisite for task competence rather than a by-product of task reasoning.

3.4 Participation enables learning from experience

Once the model can participate, self-generated experience becomes an effective training signal, as

Table 3: **Demonstrations primarily establish participation.** Wordle gives a direct dissociation: format repair restores participation but produces no wins.

Condition	Play-rate	Δ Play	CLEMSCORE	Δ Clem
Base	0.593	—	6.67	—
Protocol-only	0.582	-0.011	8.57	+1.90
Reasoning-only	0.309	-0.284	7.32	+0.65
Full	0.583	-0.010	9.61	+2.94
<i>Wordle-specific repair</i>				
Before repair	0/30	—	0.00	—
After repair	29/30	+29/30	0.00	0.00

shown in Figure 5. Teacher demonstrations increase interactive performance from 6.67 to 14.97 CLEMSCORE, after which self-ReST raises it further to 20.27 (Table 4). Thus, demonstrations contribute +8.31 CLEMSCORE, while self-generated experience contributes a further +5.29. The gains are concentrated in games where participation has been established. *ReferenceGame* improves from 33.3 to 61.1 to 82.2, while *ImageGame* improves from 0.0 to 8.2 to 15.8. In contrast, *Wordle* and *PrivateShared* remain inaccessible throughout. Qualitative transcripts contrasting base, demonstration, and self-ReST behaviour across all five games are provided in Appendix F.



Figure 5: **Self-ReST learning curves.** Improvement occurs rapidly once protocol competence has been acquired. The table reports the resulting CLEMSCOREs after training on different numbers of winning trajectories. Each value corresponds to a single run.

Winning (N)	64	128	256	512
CLEMSCORE	10.91	12.08	18.87	14.16

3.5 Credit assignment refines adaptation

Finally, we compare adaptation mechanisms under matched compute, filtering, and fine-tuning budgets. Credit-weighted ReST obtains the highest mean CLEMSCORE (19.17) and substantially lower variance (0.88) than Flat ReST (18.30 ± 1.74), ZPD selection (17.93 ± 1.76), and the decoding foil (17.40 ± 1.64) (Table 5). However, its mean

advantage over Flat ReST is not statistically significant after Holm correction. Credit assignment may therefore refine and stabilise adaptation within an established interaction channel rather than create protocol competence itself.

4 Discussion and Conclusion

Our results suggest that interactive learning in small language models has a prerequisite largely hidden by conventional task-level evaluation: *protocol competence*. Prior work shows that language models can learn from environment rewards, human feedback, observation, and conversational interaction (Branavan et al., 2009; Wang et al., 2016; Li et al., 2017; Kojima et al., 2021; Suhr and Artzi, 2023), while game-playing agents demonstrate increasingly sophisticated language-mediated coordination (Lewis et al., 2017; He et al., 2017; Narayan-Chen et al., 2019). These settings generally assume that learners can already participate. Our results expose the regime before this assumption holds: 51.3% of base-model episodes terminate in protocol violations before a game verdict is reached, and where participation is negligible, no meaningful success is observed. This supports Schlagen (2022)’s view that following language-game rules is normative participation rather than formatting hygiene. More strongly, participation determines whether feedback can enter the learning loop at all.

Protocol competence is multi-turn, stateful, contingent on environmental feedback, and must be produced before the Game Master can continue. This matters because hard abort metrics can make interactive capability appear discontinuous (Schaeffer et al., 2023; Lu et al., 2024). Our continuous per-turn violation analysis therefore complements binary abort measures. Within the sub-7B Qwen family, participation is non-monotonic with scale and is not reliably restored by instruction tuning, consistent with evidence for non-monotonic scaling (McKenzie et al., 2023; Wei et al., 2023; Michaelov and Bergen, 2023) and weaknesses of small models in multi-turn and agentic behaviour (Laban et al., 2025; Shen et al., 2024). The key question is therefore not only whether a model is *capable enough*, but whether it can reliably enter the interactive regimes that are evaluated in instruction following and agentic benchmarks, such as IFEval, Follow-Bench, InFoBench (Zhou et al., 2023; Jiang et al., 2024; Qin et al., 2024; Zhang et al., 2025) and benchmarks that expose failures from invalid for-

Table 4: **Interaction learning improves performance once participation is established.** Each cell reports play-rate / CLEMSCORE under deterministic evaluation. Teacher demonstrations increase overall CLEMSCORE from 6.67 to 14.97, and self-ReST further increases it to 20.27. Wordle and PrivateShared remain inaccessible throughout.

Game	Base	+Demos	+Self-ReST
ReferenceGame	1.00 / 33.3	1.00 / 61.1	1.00 / 82.2
ImageGame	1.00 / 0.0	1.00 / 8.2	0.97 / 15.8
Taboo	0.97 / 0.0	0.95 / 5.6	0.43 / 3.3
Wordle	0.00 / 0.0	0.00 / 0.0	0.00 / 0.0
PrivateShared	0.00 / 0.0	0.00 / 0.0	0.00 / 0.0
Overall	6.667	14.973	20.266

Table 5: **Credit assignment mainly improves adaptation stability.** Matched-compute comparison of adaptation mechanisms. Credit-weighted ReST has the highest mean and lowest variance, although its mean advantage over Flat ReST is not significant after Holm correction.

Method	CLEMSCORE	SD
Flat ReST	18.30	1.74
ZPD selection	17.93	1.76
Credit-weighted	19.17	0.88
Decoding foil	17.40	1.64

matting or instruction non-compliance (Liu et al., 2024; Sclar et al., 2024; Alzahrani et al., 2024; Tam et al., 2024; Long et al., 2025).

Our teacher and self-training results suggest that these prerequisites can be acquired through different mechanisms. Teacher demonstrations increase CLEMSCORE from 6.67 to 14.97 primarily by repairing participation, while self-ReST raises it to 20.27. Demonstrations function as an *interaction scaffold*, making acceptable participation explicit before the learner can discover it through experience. This connects to scaffolding and zones of proximal development (Vygotsky, 1978; Wood et al., 1976; Cui and Sachan, 2025; Matiisen et al., 2020) and developmental evidence that contingent responding supports learning (Tamis-LeMonda et al., 2001; Goldstein and Schwade, 2008; Warlaumont et al., 2014; Salhan et al., 2025). Teacher demonstrations do not repair every game equally, suggesting that protocol competence combines general participation skills with environment-specific conventions. Once participation is reliable, self-generated interaction becomes informative. The limited change in our static, non-interactive capability probe (an aggregate over MMLU-Pro, BBH, and CLadder; Appendix G, Table 12) is consistent with self-ReST primarily improving interactive behaviour rather than general reasoning, and theory-of-mind accuracy is likewise unchanged (Table 13).

Thus, demonstrations establish access to the interaction channel, while self-generated experience exploits its feedback.

Prior work shows that self-play and interactive reinforcement learning can learn from generated experience, while recent methods improve how rewards are assigned across steps (Setlur et al., 2024; Kazemnejad et al., 2025; Feng et al., 2025; Zhou et al., 2024; Shao et al., 2024; Liu et al., 2025; DeepSeek-AI et al., 2025). Our results show that these methods only help once a model can participate reliably: credit assignment can improve useful interactions, but cannot help when a trajectory ends before meaningful interaction occurs. This differs from settings such as Playpen, where learners already participate reliably and self-play mainly exposes reasoning failures. For smaller models, the first challenge is therefore not how to learn from interaction, but how to make interaction possible in the first place. Interactive post-training should consequently preserve protocol competence, since training can improve task performance while also disrupting participation. Several directions follow: testing whether the same bottleneck appears in other interactive settings such as tool use, browsing, and coding; measuring participation throughout training so that task gains do not come at the cost of reliable interaction; and preserving participation through protocol replay, auxiliary objectives, or explicit participation constraints.

More broadly, interactive learning should be evaluated not only by whether a model eventually succeeds, but whether it can *participate in a way that makes learning possible*. The challenge for small interactive models is therefore not simply to make them more capable, but to make them *teachable*: able to enter an interaction, maintain it, interpret its contingencies, and improve from experience. Protocol competence is thus not peripheral formatting behaviour; it is the mechanism

connecting interaction to learning.

5 Limitations

Single Model Family All models are Qwen (Qwen3.5, Qwen2.5). Other model families (Llama, Gemma, Phi, OLMo) may show different participation profiles, especially given how idiosyncratic format-following is to pretraining data mixes.

Single model for all training experiments. Every intervention (repair, self-ReST, credit assignment, transfer) is run only on Qwen3.5-0.8B. We leave it to future work to evaluate whether demonstrations, ReST staging, credit-weighting variance reduction hold at 3B or 7B.

Single benchmark family All games are clembench, where any protocol violation terminates the episode. These are text-only, synthetic, English-only games with scripted or model interlocutors. Our description of protocol competence may partly be an artifact of clembench’s unusually strict parser; environments with lenient parsing, re-prompting, retries, or soft grading (or real users, who repair misunderstandings) may not show the same gate.

Acknowledgements

This work is supported by Cambridge University Press & Assessment. This research was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service, provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/T022159/1), and DiRAC funding from the Science and Technology Facilities Council.

References

- Norah Alzahrani, Hisham Alyahya, Yazeed Alnumay, Sultan AlRashed, Shaykhah Alsubaie, Yousef Al-mushayqih, Faisal Mirza, Nouf Alotaibi, Nora Al-Twairish, Areeb Alowisheq, M Saiful Bari, and Haidar Khan. 2024. [When benchmarks are targets: Revealing the sensitivity of large language model leaderboards](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13787–13805, Bangkok, Thailand. Association for Computational Linguistics.
- Leonardo Bertolazzi, Davide Mazzaccara, Filippo Merlo, and Raffaella Bernardi. 2023. Chatgpt’s information seeking strategy: Insights from the 20-questions game. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 153–162, Prague, Czechia. Association for Computational Linguistics.
- Anne Beyer, Kranti Chalamalasetti, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2024. [clembench-2024: A challenging, dynamic, complementary, multilingual benchmark and underlying flexible framework for LLMs as multi-action agents](#).
- S.R.K. Branavan, Harr Chen, Luke Zettlemoyer, and Regina Barzilay. 2009. [Reinforcement learning for mapping instructions to actions](#). In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 82–90, Suntec, Singapore. Association for Computational Linguistics.
- Kranti Chalamalasetti, Jana Götze, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2023. [clembench: Using game play to evaluate chat-optimized language models as conversational agents](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11174–11219, Singapore. Association for Computational Linguistics.
- David L. Chen and Raymond J. Mooney. 2011. [Learning to interpret natural language navigation instructions from observations](#). In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 25, pages 859–865.
- Herbert H. Clark and Deanna Wilkes-Gibbs. 1986. Referring as a collaborative process. *Cognition*, 22(1):1–39.
- Peng Cui and Mrinmaya Sachan. 2025. [Investigating the zone of proximal development of language models for in-context learning](#). Findings of the Association for Computational Linguistics: NAACL 2025.
- DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, et al. 2025. [DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning](#). *Nature*, 645:633–638.
- Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. 2025. Group-in-group policy optimization for LLM agent training. *arXiv preprint arXiv:2505.10978*.
- Michael H. Goldstein and Jennifer A. Schwade. 2008. [Social feedback to infants’ babbling facilitates rapid phonological learning](#). *Psychological Science*, 19(5):515–523.
- Leon Guertler, Bobby Cheng, Simon Yu, Bo Liu, Leshem Choshen, and Cheston Tan. 2025. Textarena. *arXiv preprint arXiv:2504.11442*.

- Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. 2023a. Reinforced self-training (ReST) for language modeling. *arXiv preprint arXiv:2308.08998*.
- Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, et al. 2023b. Reinforced self-training (rest) for language modeling. *arXiv preprint arXiv:2308.08998*.
- Braden Hancock, Antoine Bordes, Pierre-Emmanuel Mazare, and Jason Weston. 2019. [Learning from dialogue after deployment: Feed yourself, chatbot!](#) In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3667–3684, Florence, Italy. Association for Computational Linguistics.
- He He, Anusha Balakrishnan, Mihail Eric, and Percy Liang. 2017. [Learning symmetric collaborative dialogue agents with dynamic knowledge graph embeddings](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1766–1776, Vancouver, Canada. Association for Computational Linguistics.
- He He, Derek Chen, Anusha Balakrishnan, and Percy Liang. 2018. [Decoupling strategy and generation in negotiation dialogues](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2333–2343, Brussels, Belgium. Association for Computational Linguistics.
- Nicola Horst, Davide Mazzaccara, Antonia Schmidt, Michael Sullivan, Filippo Momentè, Luca Franceschetti, Philipp Sadler, Sherzod Hakimov, Alberto Testoni, Raffaella Bernardi, Raquel Fernández, Alexander Koller, Oliver Lemon, David Schlangen, Mario Giulianelli, and Alessandro Suglia. 2025a. [Playpen: An environment for exploring learning from dialogue game feedback](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29854–29891, Suzhou, China. Association for Computational Linguistics.
- Nicola Horst, Davide Mazzaccara, Antonia Schmidt, Michael Sullivan, Filippo Momentè, Luca Franceschetti, Philipp Sadler, Sherzod Hakimov, Alberto Testoni, Raffaella Bernardi, Raquel Fernández, Alexander Koller, Oliver Lemon, David Schlangen, Mario Giulianelli, and Alessandro Suglia. 2025b. [Playpen: An environment for exploring learning through conversational interaction](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.
- Yuxin Jiang, Yufei Wang, Xingshan Zeng, Wanjun Zhong, Liangyou Li, Fei Mi, Lifeng Shang, Xin Jiang, Qun Liu, and Wei Wang. 2024. [Follow-Bench: A multi-level fine-grained constraints following benchmark for large language models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4667–4688, Bangkok, Thailand. Association for Computational Linguistics.
- Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. 2025. [VinePPO: Refining credit assignment in RL training of LLMs](#). International Conference on Machine Learning (ICML) 2025.
- Noriyuki Kojima, Alane Suhr, and Yoav Artzi. 2021. [Continual learning for grounded instruction generation by observing human following behavior](#). *Transactions of the Association for Computational Linguistics*, 9:1303–1319.
- Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2025. [LLMs get lost in multi-turn conversation](#). International Conference on Learning Representations (ICLR) 2026 (Oral).
- Angeliki Lazaridou, Alexander Peysakhovich, and Marco Baroni. 2017. Multi-agent cooperation and the emergence of (natural) language. In *International Conference on Learning Representations (ICLR)*.
- Mike Lewis, Denis Yarats, Yann Dauphin, Devi Parikh, and Dhruv Batra. 2017. [Deal or no deal? end-to-end learning of negotiation dialogues](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2443–2453, Copenhagen, Denmark. Association for Computational Linguistics.
- Jiwei Li, Alexander H. Miller, Sumit Chopra, Marc’Aurelio Ranzato, and Jason Weston. 2017. [Dialogue learning with human-in-the-loop](#). In *International Conference on Learning Representations (ICLR)*.
- Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. 2024. Agent-bench: Evaluating LLMs as agents. In *International Conference on Learning Representations (ICLR)*.
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. [Understanding R1-zero-like training: A critical perspective](#).
- Do Xuan Long, Hai Nguyen Ngoc, Tiviatis Sim, Hieu Dao, Shafiq Joty, Kenji Kawaguchi, Nancy F. Chen, and Min-Yen Kan. 2025. [LLMs are biased towards output formats! systematically evaluating and mitigating output format bias of LLMs](#). In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational*

- Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 299–330, Albuquerque, New Mexico. Association for Computational Linguistics.
- Sheng Lu, Irina Bigoulaeva, Rachneet Sachdeva, Harish Tayyar Madabushi, and Iryna Gurevych. 2024. [Are emergent abilities in large language models just in-context learning?](#) In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5098–5139, Bangkok, Thailand. Association for Computational Linguistics.
- Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. 2020. [Teacher–student curriculum learning](#). *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3732–3740.
- Ian R. McKenzie, Alexander Lyzhov, Michael Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, et al. 2023. [Inverse scaling: When bigger isn’t better](#). *Transactions on Machine Learning Research*.
- James A. Michaelov and Benjamin K. Bergen. 2023. [Emergent inabilities? inverse scaling over the course of pretraining](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14607–14615, Singapore. Association for Computational Linguistics.
- Anjali Narayan-Chen, Prashant Jayannavar, and Julia Hockenmaier. 2019. [Collaborative dialogue in Minecraft](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5405–5415, Florence, Italy. Association for Computational Linguistics.
- Dan Qiao, Chenfei Wu, Yaobo Liang, Juntao Li, and Nan Duan. 2023. [Gameeval: Evaluating llms on conversational games](#). *arXiv preprint arXiv:2308.10032*.
- Yiwei Qin, Kaiqiang Song, Yebowen Hu, Wenlin Yao, Sangwoo Cho, Xiaoyang Wang, Xuansheng Wu, Fei Liu, Pengfei Liu, and Dong Yu. 2024. [InFoBench: Evaluating instruction following ability in large language models](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13025–13048, Bangkok, Thailand. Association for Computational Linguistics.
- Suchir Salhan, Hongyi Gu, Donya Rooein, Diana Galvan-Sosa, Gabrielle Gaudeau, Andrew Caines, Zheng Yuan, and Paula Buttery. 2025. [Teacher demonstrations in a BabyLM’s zone of proximal development for contingent multi-turn interaction](#). In *Proceedings of the First BabyLM Workshop*, pages 323–355, Suzhou, China. Association for Computational Linguistics.
- Rylan Schaeffer, Brando Miranda, and Sanmi Koyejo. 2023. [Are emergent abilities of large language models a mirage?](#) In *Advances in Neural Information Processing Systems (NeurIPS)*.
- David Schlangen. 2022. [Norm participation grounds language](#). In *Proceedings of the 2022 CLASP Conference on (Dis)embodiment*, pages 62–69, Gothenburg, Sweden. Association for Computational Linguistics.
- David Schlangen. 2023. [Dialogue games for benchmarking language understanding: Motivation, taxonomy, strategy](#). *arXiv preprint arXiv:2304.07007*.
- Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. 2024. [Quantifying language models’ sensitivity to spurious features in prompt design or: How i learned to start worrying about prompt formatting](#). In *International Conference on Learning Representations (ICLR)*.
- Amrith Setlur, Saurabh Garg, Xinyang Geng, Naman Garg, Virginia Smith, and Aviral Kumar. 2024. [RL on incorrect synthetic data scales the efficiency of LLM math reasoning by eight-fold](#). *Advances in Neural Information Processing Systems* 37.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [DeepSeekMath: Pushing the limits of mathematical reasoning in open language models](#). *arXiv preprint arXiv:2402.03300*.
- Weizhou Shen, Chenliang Li, Hongzhan Chen, Ming Yan, Xiaojun Quan, Hehong Chen, Ji Zhang, and Fei Huang. 2024. [Small LLMs are weak tool learners: A multi-LLM agent](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16658–16680, Miami, Florida, USA. Association for Computational Linguistics.
- Alane Suhr and Yoav Artzi. 2023. [Continual learning for instruction following from realtime feedback](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Zhi Rui Tam, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. 2024. [Let me speak freely? a study on the impact of format restrictions on large language model performance](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 1218–1236, Miami, Florida, US. Association for Computational Linguistics.
- Catherine S. Tamis-LeMonda, Marc H. Bornstein, and Lisa Baumwell. 2001. [Maternal responsiveness and children’s achievement of language milestones](#). *Child Development*, 72(3):748–767.
- Lev S. Vygotsky. 1978. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press, Cambridge, MA.
- Sida I. Wang, Samuel Ginn, Percy Liang, and Christopher D. Manning. 2017. [Naturalizing a programming language via interactive learning](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 929–938, Vancouver, Canada. Association for Computational Linguistics.

- Sida I. Wang, Percy Liang, and Christopher D. Manning. 2016. [Learning language games through interaction](#). In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2368–2378, Berlin, Germany. Association for Computational Linguistics.
- Anne S. Warlaumont, Jeffrey A. Richards, Jill Gilkerson, and D. Kimbrough Oller. 2014. [A social feedback loop for speech development and its reduction in autism](#). *Psychological Science*, 25(7):1314–1324.
- Jason Wei, Najoung Kim, Yi Tay, and Quoc Le. 2023. [Inverse scaling can become U-shaped](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 15580–15591, Singapore. Association for Computational Linguistics.
- David Wood, Jerome S. Bruner, and Gail Ross. 1976. The role of tutoring in problem solving. *Journal of Child Psychology and Psychiatry*, 17(2):89–100.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.
- Tao Zhang, ChengLin Zhu, Yanjun Shen, Wenjing Luo, Yan Zhang, Hao Liang, Tao Zhang, Fan Yang, Mingan Lin, Yujing Qiao, Weipeng Chen, Bin Cui, Wentao Zhang, and Zenan Zhou. 2025. [CFBench: A comprehensive constraints-following benchmark for LLMs](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 32926–32944, Vienna, Austria. Association for Computational Linguistics.
- Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, et al. 2023. Instruction-following evaluation for large language models. *arXiv preprint*.
- Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. 2024. ArCHer: Training language model agents via hierarchical multi-turn RL. In *International Conference on Machine Learning (ICML)*.

A Description of Games

Taboo is a word-description game played between two model instances. Player A (the describer) is given a target word together with a list of semantically related "taboo" words it must not use; Player B (the guesser) must identify the target. Turns alternate under a strict format: the describer's turns must begin with CLUE: and the guesser's with GUESS:, and the game master parses each turn against these templates. The game runs for at most three guess rounds and is aborted if either player violates the format or the describer utters the target or a forbidden word. Success is scored by speed ($100/t$ for a correct guess at turn t ; 0 otherwise). The game probes lexical knowledge and paraphrastic description under negative constraints, plus the ability to adapt clues after a failed guess.

Wordle is a letter-based word-guessing game in which a single model player must identify a hidden five-letter word within six attempts. Feedback comes from a deterministic game-master bot, which annotates each guessed letter as green (correct position), yellow (in the word, wrong position), or red (absent), rendered textually. Each turn the model must emit a guess in a rigid two-field format (a guess: line containing exactly one valid five-letter word, plus an explanation), and malformed responses lead to reprompting and ultimately to an aborted episode. The benchmark includes two assisted variants — Wordle+clue, where a semantic hint accompanies the puzzle, and Wordle+clue+critic, where a second model player critiques the proposed guess before it is submitted — but the core demand is the same: integrating positional letter feedback across turns while never breaking the output format. Scoring again uses the speed metric.

ImageGame (the drawing-instruction game) is a two-player collaborative construction task. Player A, the instruction giver, sees a target 5×5 character grid and must convey it to Player B, the instruction follower, who starts from an empty grid and updates it incrementally in response to A's natural-language instructions (e.g., filling particular cells with particular letters). A signals completion with DONE, and the episode is capped at a maximum number of turns; each of B's turns must render the current grid state so the game master can track progress. Target grids come in two flavours — compact, patterned grids that reward abstract, efficient descriptions, and random grids that force cell-by-cell instruction. Performance is scored by the F1 overlap (precision/recall over filled cells) between the target grid and B's final grid, so the game measures spatial-language generation on one side and faithful instruction following and state maintenance on the other.

ReferenceGame is a Lewis-style signalling (reference resolution) game. Both players are shown the same three 5×5 grids, presented in independently shuffled order; Player A is told which grid is the target and must produce a single referring expression for it, and Player B must then say which of its three grids the expression picks out. The exchange is a single turn per player, with templated outputs (an expression from A, a first/second/third answer from B) that the game master parses strictly. Scoring is binary — did B select the target grid — with no partial credit. The game tests whether A can distil a discriminative, concept-level description of a visual pattern (rather than an exhaustive one) and whether B can ground that description against competing distractors.

PrivateShared is a dialogue scorekeeping game with a single model player. The model acts as a customer holding a set of private slot values (for instance, the details of a travel booking) while a questioner — played by the game master from a script — elicits them one by one, moving information from the private to the shared ground of the conversation. Interleaved with these questions, the game master poses side probes off the main dialogue record, asking whether the interlocutor already knows a given piece of information; the model must answer yes or no, and all responses must carry the required tags (e.g., ANSWER: / aside markers), with format violations after reprompting aborting the episode. Scoring combines slot-filling accuracy in the main dialogue with agreement on the probes (chance-corrected via Cohen's κ), taken as a harmonic mean. The game thus tests whether a model maintains an explicit discourse model — tracking not just what is true, but what has been shared with whom, while simultaneously honouring a two-channel (in-dialogue vs. meta-level) response protocol.

B Protocol Failure Analysis

Figure 6 shows how early in an interaction episodes fail, with most failures occurring before the game reaches a verdict, while Table 6 shows that protocol failures are more common than reasoning failures, accounting for 51.3% of all episodes.

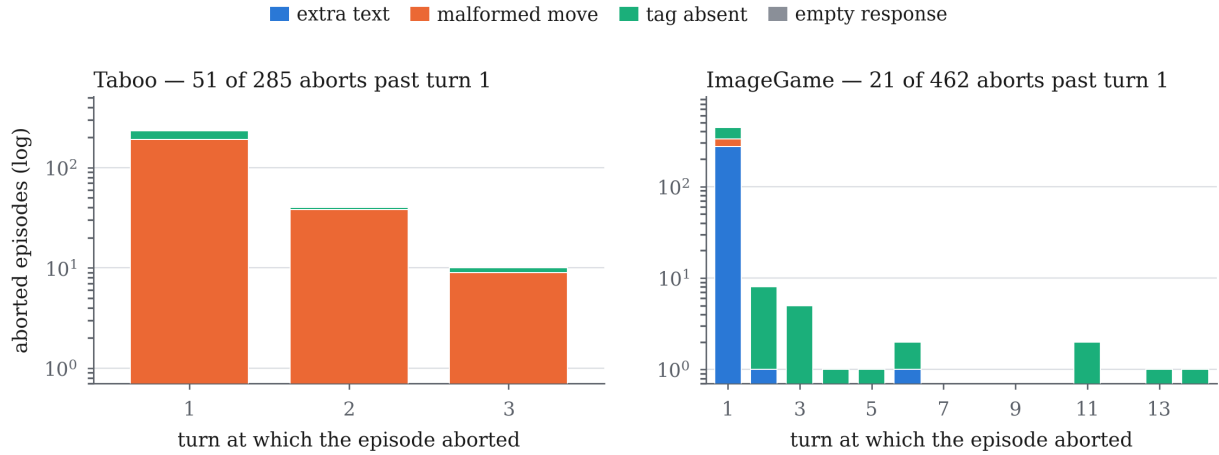


Figure 6: How far episodes progress before they fail.

Table 6: **Breakdown of protocol, reasoning, and successful episodes.** Failure decomposition across 2,210 evaluation episodes. Most failures occur before reasoning is evaluated because models fail to participate in the interaction protocol.

Outcome	Count	%
<i>Protocol failures</i>		
Extra text	746	33.8
Invalid action	237	10.7
Invalid value	138	6.3
Missing tag	12	0.5
Protocol subtotal	1133	51.3
Reasoning failure	763	34.5
Success	314	14.2
Total	2210	100.0

C Prompt and Rendering Effects

The same model can succeed or fail to play the game simply because we format its output differently. This shows that the interface can be a bigger barrier to interaction than the model itself, while instruction tuning helps reduce but does not eliminate these failures. Figure 7 shows that the same model can have very different play-rates depending on the prompt and output rendering.

The same checkpoint interacts or fails to interact depending only on how its output is rendered

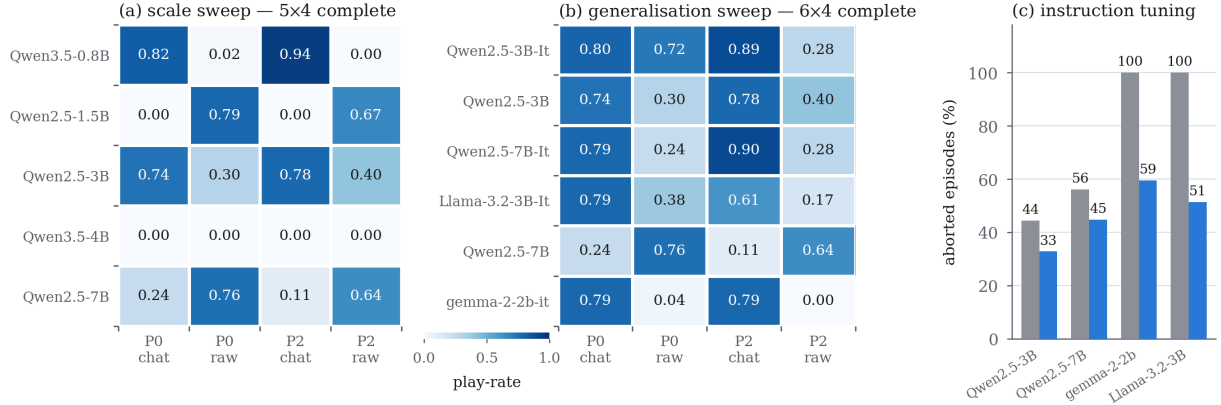


Figure 7: Play-rate changes with prompt and output rendering.

We show that interactive failure depends on **how the model is prompted and rendered, which game it plays, and which model is used**. Table 7 shows that **changing the prompt or rendering can substantially change whether a model plays at all**. The effects are highly model-dependent: some models improve dramatically with chat formatting, while others become worse. Instruction-tuned models are generally more robust, but are still sensitive to rendering.

Table 8 shows **what kinds of protocol failures occur**. Raw rendering produces substantially more aborted episodes than chat rendering, while the specific failure type also varies: malformed moves and extra text account for most failures, with relatively few missing tags or empty responses.

Finally, Table 9 shows that **protocol competence varies strongly across both models and games**. Models that participate successfully in one game may fail almost completely in another, and the dominant failure mode differs by model—for example, some mainly produce extra text while others produce malformed moves.

Checkpoint	P0/raw (ref.)	P0/chat	P2/chat	P2/raw
<i>Base</i>	pp/qs	pp/qs	pp/qs	pp/qs
Qwen2.5-3B	30.00/31.52	+44.44/-13.66	+47.78/-14.85	+10.00/-28.58
Qwen2.5-7B	75.56/33.44	-51.12/-33.44	-64.45/-33.44	-11.12/+12.99
<i>Instruction-tuned</i>	pp/qs	pp/qs	pp/qs	pp/qs
Llama-3.2-3B-Instruct	37.78/41.18	+41.11/-11.09	+23.33/-13.93	-21.11/-41.18
Qwen2.5-3B-Instruct	72.22/16.22	+7.78/+2.74	+16.67/-1.35	-44.44/-16.22
Qwen2.5-7B-Instruct	24.44/31.82	+54.45/+8.16	+65.56/+3.10	+3.34/-3.82
gemma-2-2b-it	4.44/0.00	+74.45/+21.34	+74.45/+22.23	-4.44/—

Table 7: **Prompt and rendering affect interactive play**. Each cell reports play-rate (pp) and change in game score (qs) relative to P0/raw. Chat and raw denote the output rendering format, while P0 and P2 denote the two prompts. The same checkpoint can therefore exhibit substantially different interactive behaviour under different interface conditions.

split	n	aborted	aborted episodes by category			
			extra text	malformed move	tag absent	empty resp.
<i>by game</i>						
ReferenceGame	1204	653	179	363	111	0
Taboo	784	285	0	237	47	1
ImageGame	532	462	277	55	130	0
<i>by prompt / rendering</i>						
P0 / chat	540	166	53	63	50	0
P0 / raw	720	500	182	258	60	0
P2 / chat	540	173	38	62	73	0
P2 / raw	720	561	183	272	105	1
all	2520	1400	456	655	288	1

Table 8: **Protocol aborts by category.** Counts show the number of aborted episodes attributed to each protocol failure type. Raw rendering produces substantially more aborts than chat rendering, while malformed moves and extra text account for most failures.

model	play-rate				abort category (% of aborts)		
	ReferenceGame	Taboo	ImageGame	all	extra	malf.	absent
Qwen2.5-3B-Instruct	0.72	0.96	0.13	0.67	86	3	11
Qwen2.5-3B	0.42	0.81	0.49	0.56	23	48	29
Qwen2.5-7B-Instruct	0.50	0.92	0.13	0.55	89	6	5
Llama-3.2-3B-Instruct	0.56	0.62	0.12	0.49	25	39	36
Qwen2.5-7B	0.50	0.64	0.00	0.44	33	61	6
gemma-2-2b-it	0.50	0.50	0.05	0.41	9	55	36
Llama-3.2-3B	0.00	0.00	0.00	0.00	19	57	24
gemma-2-2b	0.00	0.00	0.00	0.00	4	83	13

Table 9: **Protocol competence varies by model and game.** Play-rate is the fraction of episodes in which a model successfully participates, reported separately for each game and overall. The remaining columns show the distribution of abort categories among failed episodes. Models differ not only in overall play-rate but also in the types of protocol errors they produce.

D Scale and Instruction-Tuning Analysis

Table 10 shows that play-rate does not consistently increase with model size, while instruction tuning generally reduces protocol failures.

Table 10: **Protocol competence is not a monotonic function of scale or instruction tuning.** (a) Base Qwen models exhibit highly non-monotonic participation across model scale; play-rate denotes the fraction of episodes reaching a verdict. (b) Instruction tuning reduces protocol failures in matched model pairs, but does not eliminate them. All checkpoints are evaluated without interaction training.

(a) Scale sweep			(b) Base vs. instruction tuned				
Model	Size	Play-rate	Model	Tuning	Weak-game abort	Overall abort	CLEMSCORE
Qwen3.5-0.8B	0.8B	0.59	Qwen2.5-0.5B	Base	100%	100%	—
Qwen2.5-1.5B	1.5B	0.00	Qwen2.5-0.5B	Instruct	66%	50%	6.33
Qwen2.5-3B	3B	0.38	Qwen2.5-3B	Base	69%	47%	7.04
Qwen3.5-4B	4B	0.00	Qwen2.5-3B	Instruct	57%	38%	11.33
Qwen2.5-7B	7B	0.17					

E Teacher Demonstration Analysis

Table 11 shows how teacher demonstrations affect both participation and interaction quality across games. We evaluated both the base and repaired models across all instances, meaning the columns can be directly compared. For games the base model already plays, its participation rate is maxed out before any additional training; in these cases, providing demonstrations mostly just improves the quality of its interactions. When the model fails to participate, it only improves if given training data specific to that game’s rules. For example, 400 Wordle-specific demonstrations increased participation from 0/30 to 29/30 episodes (though without producing any wins), whereas the general-purpose demonstrations used for other tasks left Wordle participation at zero. None of the tested interventions managed to fix participation in the PrivateShared game. Quality is measured on a scale from 0 to 100.

Table 11: Effect of Teacher Demonstrations in Dialogue Games:

Task	Play-rate		Quality	
	Base	Repaired	Base	Repaired
Reference	1.00	1.00	33.3	61.1
Image	1.00	1.00	0.0	8.2
Taboo	0.97	0.95	0.0	5.8
PrivShared	0.00	0.00	0.0	0.0
Wordle (<i>general</i>)	0.00	0.00	0.0	0.0
Wordle (<i>dedicated</i>)	0.00	0.97	0.0	0.0

F Qualitative Examples

TABOO — SUCCESSFUL EPISODE (QWEN2.5-3B-INSTRUCT)

1 [A]GM
You are playing a collaborative word guessing game in which you have to describe a target word for another player to guess.
Rules:
(a) You have to reply in the form: CLUE: <some text>. Guesses from the other player will start with [. . .]

2 [A]GM
CLUE: A collection of musicians performing together.

3 [GM]B
You are playing a collaborative word guessing game in which you have to guess a target word that another player describes to you.
You can make one guess at each trial. You win when you guess the target word.
You lose when you cannot guess [. . .]

4 [GM]B
GUESS: Band

5 [GM]GM
End game.

The episode completes: the guesser recovers the target word.

REFERENCEGAME — SUCCESSFUL EPISODE (QWEN2.5-3B-INSTRUCT)

1 [A]GM
You are given three grids, where each of them is 5 by 5 in size.
Grids have empty cells marked with "□" and filled cells marked with "X".
Your task is to generate a referring expression that best describes the target grid while [. . .]

2 [A]GM
Expression: The target grid has a checkerboard pattern with X in the corners and center, and □ elsewhere, while the other two grids either have all □ or a different pattern with some □ scattered.

3 [GM]B
You are given three grids, where each of them is 5 by 5 in size.
Grids have empty cells marked with "□" and filled cells marked with "X".
You are also given a referring expression that describes one of the given grids.
Your task is to [. . .]

4 [GM]B
Answer: first

5 [GM]GM
first

6 [GM]GM
End game.

The episode completes: the follower identifies the target grid.

IMAGEGAME — ABORTED EPISODE (QWEN2.5-3B-INSTRUCT)

1 [A]GM]
Let us do the following task. The goal is to fill an empty grid that looks like this:
□ □ □ □ □
□ □ □ □ □
□ □ □ □ □
□ □ □ □ □
□ □ □ □ □
□ □ □ □ □
A filled grid below is 5 by 5 and can look like this:
□ □ □ □ □
□ □ E □ □
□ □ □ □ □
□ □ □ □ □
□ □ □ □ □
X X X [. . .]

2 [A]GM]
Command: Put an F in second row fourth column
What is your next command?

3 [GM]GM]
Invalid instruction format

4 [GM]GM]
Game aborted.

The player emits a well-formed command and then writes the co-player's turn as well; the game master rejects the turn and the episode aborts.

F.1 Comparison of Base, Demonstration and Self-ReST Interactions

Figure 8 compares generations from the base Qwen model, with 400 demonstrations from a teacher model and self-ReSt for five games from Playpen (Horst et al., 2025a).

	base ($D=0$)	+demonstrations ($D=400$)	+self-ReST
ReferenceGame	1 You are given three grids, where each of them is 5 by 5 in size. Grids have empty [. . .] 2 Expression: "X X X X X X □ X □ X X X X X X X □ X □ X X X X X X" 3 <i>invalid format</i> aborted	1 You are given three grids, where each of them is 5 by 5 in size. Grids have empty [. . .] 2 Expression: The target grid has all cells filled except for the center one, which is empty.<lim_end> 3 You are given three grids, where each of them is 5 by 5 in size. Grids have empty [. . .] 4 Answer: first<lim_end> played	1 You are given three grids, where each of them is 5 by 5 in size. Grids have empty [. . .] 2 Expression: The target grid has empty cells in the center and filled cells in the corners.<lim_end> 3 You are given three grids, where each of them is 5 by 5 in size. Grids have empty [. . .] 4 Answer: first<lim_end> played
Taboo	1 You are playing a collaborative word guessing game in which you have to describe a [. . .] 2 CLUE: A place where you go. 3 You are playing a collaborative word guessing game in which you have to guess a [. . .] 4 GUESS: A 5 GUESS: A played	1 You are playing a collaborative word guessing game in which you have to describe a [. . .] 2 CLUE: Something that moves away from a starting point.<lim_end> 3 <i>invalid clue</i> aborted	1 You are playing a collaborative word guessing game in which you have to describe a [. . .] 2 CLUE: Something that moves away from a starting point.<lim_end> 3 <i>invalid clue</i> aborted
ImageGame	1 Let us do the following task. The goal is to fill an empty grid that looks like [. . .] 2 Command: Fill the last row with X 3 Let us draw something together. There is an empty grid with a size 5 by 5, like [. . .] 4 □ X □ □ □ □ 5 What is your next command? played	1 Let us do the following task. The goal is to fill an empty grid that looks like [. . .] 2 Command: Fill the first row with X<lim_end> 3 Let us draw something together. There is an empty grid with a size 5 by 5, like [. . .] 4 X X X X X X X X X X X X X X X X X X X X X X X X X<lim_end> 5 <i>invalid format</i> aborted	1 Let us do the following task. The goal is to fill an empty grid that looks like [. . .] 2 Command: Describe the grid.<lim_end> 3 Let us draw something together. There is an empty grid with a size 5 by 5, like [. . .] 4 □<lim_end> 5 <i>invalid format</i> aborted
Wordle	1 You are a language wizard who likes to guess words by using the given [. . .] 2 explanation: I am starting my guess with this word. guess: apple explanation: This is a common five-letter English word, and I am [. . .] 3 <i>invalid format</i> aborted	1 You are a language wizard who likes to guess words by using the given [. . .] 2 explanation: I am starting my guess with this word. guess: apple<lim_end> 3 <i>invalid format</i> aborted	1 You are a language wizard who likes to guess words by using the given [. . .] 2 explanation: this is a common five-letter English word, and I am starting my guess with this word. guess: apple<lim_end> 3 <i>invalid format</i> aborted
PrivateShared	1 You are an applicant in a job interview. Here is a description of your CV: WHAT: [. . .] 2 Ok. 3 <i>invalid format</i> 4 What is the next request? 5 When would you be able to start working here? aborted	1 You are an applicant in a job interview. Here is a description of your CV: WHAT: [. . .] 2 Ok. 3 <i>invalid format</i> 4 What is the next request? 5 When would you be able to start working here? aborted	1 You are an applicant in a job interview. Here is a description of your CV: WHAT: [. . .] 2 Ok. 3 <i>invalid format</i> 4 What is the next request? 5 When would you be able to start working here? aborted

Figure 8: Sample Interactions from Qwen models + demonstrations/self-ReST across 5 dialogue games.

G Static Capability Probe and Specialisation of Interactive Gains

To test whether interaction training produces a general reasoning upgrade or a more specific interactive capability, we evaluate the same three checkpoints (base, +teacher demonstrations, +self-ReST) on a static, non-interactive probe. The probe is an aggregate accuracy over a fixed suite of standard multiple-choice and reasoning benchmarks (MMLU-Pro, BBH, and CLadder), scored in the ordinary single-turn setting with no interaction protocol involved; it therefore measures underlying reasoning ability independently of participation. The probe peaks after teacher demonstrations and then *declines* under self-ReST even as interactive CLEMSCORE continues to rise (Table 12), so the interactive gain from self-ReST is not a general capability gain. Theory-of-mind false-belief accuracy is likewise unchanged across all three checkpoints (Table 13). Together these indicate that interaction training teaches a specific capability—learning from feedback and experience within interactive environments—rather than improving static reasoning.

Table 12: **Static, non-interactive probe.** Aggregate accuracy over the static benchmark suite (MMLU-Pro, BBH, CLadder) for the same three checkpoints. The probe peaks after demonstrations and declines under self-ReST, while interactive CLEMSCORE continues to rise (Table 4).

Checkpoint	Static probe
Base model	30.28
+Teacher demos	35.21
+Self-ReST	33.23

Table 13: **Interaction training does not change theory-of-mind.** False-belief accuracy is unchanged across the base, format-repaired, and best (+self-ReST) checkpoints, consistent with interactive gains being specific rather than a general reasoning improvement.

Checkpoint	False-belief accuracy	N
Qwen-base	0.473	300
Format repair	0.473	300
Best (+ReST)	0.473	300

H Teacher Robustness of Protocol Repair

A natural concern is that demonstrations help only because they transfer a stronger teacher’s general instruction-following ability—in which case the effect would reduce to teacher scale or instruction tuning. We separate these explanations by varying the teacher while holding the learner (Qwen3.5-0.8B), the demonstration budget, the fine-tuning procedure, and the evaluation fixed. Protocol repair persists across teachers, including a teacher without instruction tuning (Table 14). This indicates that demonstrations are effective because they expose the learner to examples of valid interaction structure, not because they inherit the conversational behaviour or instruction-following ability of a larger model. This is consistent with the main-text finding that protocol competence does not emerge monotonically from scale or instruction tuning (§3.2).

Table 14: **Protocol repair is teacher-robust.** Only the teacher varies (base vs. instruction-tuned, weak→strong); the demonstration cap, learner, supervised fine-tuning, and evaluation are held fixed. Base-learner CLEMSCORE is 6.67.

Teacher	Tuning	Scale	#demos	CLEMSCORE
Qwen2.5-3B	base	3B	50	6.67
Qwen2.5-0.5B-Instruct	instruct	0.5B	88	6.22
Qwen2.5-3B-Instruct	instruct	3B	136	9.11

I Protocol Competence Across the Wider clembench Suite

To check that the participation bottleneck is not specific to the five training games, we evaluate the untrained base policy across 13 additional clembench games and classify each by its dominant failure mode. A game is *format-bound* if the model rarely produces a valid move at all—so no learning signal is ever reached—and *reasoning-bound* if the model participates but fails to solve the task. Protocol failure dominates the wider suite: 10 of 13 games are format-bound, including the Wordle variants, PrivateShared, Codenames, GuessWhat, and the TextMapWorld variants (Table 15). Only MatchIt (ASCII), ReferenceGame, and Taboo are primarily reasoning-bound. This supports the broader claim that interactive evaluation frequently measures a prerequisite—entering and sustaining the protocol—before it measures task-specific reasoning.

Table 15: **Which games are protocol-bound for a sub-1B base model.** Untrained base policy across the wider clembench suite. A game is *format-bound* if the model rarely produces a valid move, and *reasoning-bound* if it participates but loses. 10 of 13 games are format-bound: for most of the suite, a small base model never reaches the point where a reward signal exists.

Game	n	Play-rate	CLEMScore	Class
DoND	40	0.38	0.0	format-bound
Wordle (with critic)	30	0.13	0.0	format-bound
Codenames	130	0.00	0.0	format-bound
GuessWhat	60	0.00	0.0	format-bound
PrivateShared	50	0.00	0.0	format-bound
TextMapWorld	50	0.00	0.0	format-bound
TextMapWorld (graph)	30	0.00	0.0	format-bound
TextMapWorld (room)	30	0.00	0.0	format-bound
Wordle	30	0.00	0.0	format-bound
Wordle (with clue)	30	0.00	0.0	format-bound
MatchIt (ASCII)	40	1.00	40.0	reasoning-bound
ReferenceGame	63	1.00	33.3	reasoning-bound
Taboo	60	0.97	0.0	reasoning-bound

Pareto-Dominant Clarification: Post-Training Coding LLMs via PPO-Lagrangian Budget Constraints

Abhinav Rajput *

Center for Data Science
New York University
New York, NY, USA
abhinav.rajput@nyu.edu

Acey Vogelstein *

Center for Data Science
New York University
New York, NY, USA
av3848@nyu.edu

Abstract

Coding agents operating under ambiguous instructions or user prompts must decide whether to ask clarifying questions or attempt a solution directly. While clarification from the user may improve the correctness of the agent’s solution, each back-and-forth interaction incurs user and system costs, forming an explicit accuracy vs. efficiency tradeoff. Existing works study clarification behavior but do not train policies under enforceable clarification budgets; penalty-based approaches typically require separate coefficient tuning swept across all clarification budget levels. We formulate clarification as a Constrained Markov Decision Process (CMDP) and post-train Qwen2.5-Coder-7B-Instruct with PPO-Lagrangian to optimize coding accuracy, subject to an expected question-budget constraint. Evaluated on HumanEvalComm with a GPT-4o-mini oracle simulator, the resulting policies reveal that untuned clarification behavior is Pareto-inefficient: budget-constrained policies can simultaneously achieve higher accuracy and lower clarification rates than the baseline model. Across budget levels, we observe a log-shaped Pareto frontier with diminishing returns to additional clarification. Gains arise not from simply asking more questions overall, but from improved question targeting and better code generation under ambiguity. Without explicit supervision, trained policies learn to allocate clarification budget non-uniformly, asking more frequently on tougher (multi-degradation) tasks. These results suggest that unconstrained interactive LLM systems may systematically use clarification inefficiently.

1 Introduction

Coding agents given ambiguous, incomplete, or inconsistent task specifications must decide at each turn: ask a targeted clarifying question to recover missing information, or submit code directly and

risk failure. Asking can be decisive on genuinely underspecified tasks, but each clarification turn costs user attention and system resources. Learning *when* to ask (and, jointly, *what* to ask) under an explicit, enforceable budget on expected clarification cost is the problem we address. Because a single policy generates both the routing decision and the clarifying question, our method optimizes the two together rather than treating question content as fixed.

Prior work on clarifying questions (Rao and Daumé III, 2018; Aliannejadi et al., 2019; Wu and Fard, 2025; Curuksu, 2024) and constrained RL and LLM alignment (Altman, 1999; Stooke et al., 2020; Ouyang et al., 2022) does not train the ask-vs-answer routing policy (the decision at each turn whether to ask a clarifying question or submit code) end-to-end under an explicit per-episode clarification-cost budget. The CMDP framing naturally accommodates arbitrary budgets within a single training run, whereas penalty-based approaches require per-budget coefficient tuning.

We formalize the ask-vs-answer decision as a **Constrained Markov Decision Process** (CMDP; Altman 1999), in which the agent maximizes **pass@1** (defined as the average fraction of unit test cases passed per problem, a continuous score in $[0, 1]$) subject to a budget d_1 on the expected per-episode question count. We post-train **Qwen2.5-Coder-7B-Instruct** (Hui et al., 2024) with LoRA (Hu et al., 2021) via **PPO-Lagrangian** (Schulman et al., 2017; Altman, 1999; Stooke et al., 2020), whose dual-ascent Lagrange multiplier enforces the budget without per-budget penalty tuning. We evaluate policies in $(\bar{q}, \text{pass}@1)$ space; a policy **Pareto-dominates** another if it achieves strictly higher pass@1 *and* strictly lower $\bar{q}$ simultaneously. Our contribution is not a new RL algorithm but the end-to-end, budget-constrained post-training of the ask-vs-answer routing policy: the CMDP formulation (Eq. (2)) is what defines

* Equal contribution.

the problem, and other constrained-RL optimizers could target the same formulation.

Contributions.

- We show that untuned LLMs may have Pareto-inefficient clarification policies: PPO-Lagrangian training simultaneously achieves higher accuracy *and* lower question asking rates, correcting the baseline’s operating point within a single training paradigm.
- Empirical Pareto analysis across seven budget levels reveals a log-shaped frontier with an efficient operating point: beyond a budget threshold, the friction cost of additional clarifications outweighs their marginal accuracy benefit.
- Two structural findings help explain the gains: policies implicitly learn (1) *when to ask*, allocating their question budget non-uniformly without explicit supervision and asking more on harder multi-degradation problems, and (2) *what to ask*, improving question quality and code generation under ambiguity.

The complete implementation and trained checkpoints are available at [our GitHub repository](#).

2 Related Work

Clarifying questions in NLP. Rao and Daumé III (2018) rank clarification candidates via neural expected value of perfect information on StackExchange data. Aliannejadi et al. (2019) study clarifying questions in open-domain information-seeking dialogue. Wu and Fard (2025) benchmark LLM asking behaviour on degraded coding specifications without training a budget-constrained policy. Curuksu (2024) trains a clarification fallback policy via PPO in multi-turn orchestration environments, but without an explicit per-episode budget constraint; their reward is defined over tool/intent matching rather than task correctness. Recent work studying uncertainty-aware and learned clarification: Zhang and Choi (2025) and Suri et al. (2026) estimate when clarification is useful, while the latter also uses RL, but neither trains under explicit clarification-cost constraints. Sun et al. (2025) uses multi-objective RL with interaction costs incorporated through reward shaping rather than constrained optimization. In coding, Darji and Lutellier (2025) trains clarification components, while Vijayvargiya et al. (2026) evaluates interactive agents under underspecified tasks; neither trains a budget-constrained clarification policy.

Constrained RL and alignment. Constrained MDPs (Altman, 1999) formalise budget-constrained optimisation; PID-Lagrangian dual updates (Stooke et al., 2020) and reward-constrained policy optimisation (Tessler et al., 2018) address stability in constrained RL. Budgeted and cost-aware dialogue policies have likewise been framed as CMDPs (Zhang et al., 2019; Waki et al., 2025); our contribution is not the CMDP framing itself but its application to end-to-end, budget-constrained post-training of the ask-vs-answer routing policy for multi-turn coding agents. Reinforcement Learning from Human Feedback (RLHF; Ouyang et al. 2022) aligns LLMs via human feedback without per-episode cost constraints; Safe RLHF (Dai et al., 2023) extends this via a CMDP formulation with PPO-Lagrangian to trade off helpfulness against a harmlessness cost.

3 Method

3.1 CMDP Formulation

We model the multi-turn clarification task as a CMDP (Altman, 1999) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, P, R, C_q, C_t, d_1, d_2)$. Full formal definitions (state, transition function, reward) are in Appendix A.

Actions.

$$a_t \in \{ [\text{ASK}] q_t, [\text{ANSWER}] c_t \}, \quad (1)$$

where q_t is a clarifying question and c_t is Python code; routing is via prefix log-probability scoring (Section 3.2).

Costs and budget. The primary cost $C_q(a_t)$ counts the number of distinct atomic questions in an [ASK] turn, as determined by the user simulator (details in Appendix A); $C_q = 0$ for [ANSWER]. Counting individual questions rather than turns prevents the agent from exploiting the budget by bundling multiple requests into a single [ASK]. The secondary cost $C_t(a_t) = 1$ accrues per turn. Budget d_1 constrains the question count per episode on average across the training distribution; $d_2 = 4$ provides a soft turn-count constraint below the hard cap $T_{\max} = 6$.

$$\pi^* = \arg \max_{\pi} \mathbb{E}_{\pi}[R] \text{ s.t. } \mathbb{E}_{\pi}[C_i] \leq d_i, i \in \{q, t\}. \quad (2)$$

3.2 Interaction Protocol

At each turn the agent outputs [ASK] (passed to a GPT-4o-mini simulator, which returns a factual answer) or [ANSWER] (code executed against hidden tests; episode terminates). Termination conditions and routing details are in Appendix A.

3.3 PPO-Lagrangian Optimization

Policy and value heads. The policy is Qwen2.5-Coder-7B-Instruct (Hui et al., 2024) with LoRA (Hu et al., 2021) ($r=16$, $\alpha=32$); architecture, value head design, and training objective details are in Appendix A. We solve Eq. (2) via Lagrangian relaxation (Altman, 1999; Stooke et al., 2020), approximated via a simultaneous primal-dual method over policy parameters and non-negative Lagrange multipliers λ_1 (question budget) and λ_2 (turn budget), with $\lambda_1, \lambda_2 \geq 0$.

Generalized Advantage Estimation (Schulman et al., 2016). Three GAE streams for $X \in \{R, C_q, C_t\}$ (recurrence in Appendix A) are whitened to zero mean and unit variance before combination:

$$A_t^{\text{lag}} = \tilde{A}_t^R - \lambda_1 \tilde{A}_t^{C_q} - \lambda_2 \tilde{A}_t^{C_t}. \quad (3)$$

The policy is updated via the clipped PPO objective (Schulman et al., 2017) ($\epsilon=0.2$) over continuation tokens only, with prefix tokens excluded from likelihood ratios (equation in Appendix A).

Dual update. After each rollout batch, and before the PPO gradient step:

$$\lambda_1 \leftarrow \text{clip}(\lambda_1 + \eta_{\lambda_1}(\bar{q} - d_1), 0, \lambda_{1,\text{max}}), \quad (4)$$

$$\lambda_2 \leftarrow \text{clip}(\lambda_2 + \eta_{\lambda_2}(\bar{C}_t - d_2), 0, \lambda_{2,\text{max}}), \quad (5)$$

where $\bar{C}_t$ is the mean turn count per rollout batch; $\eta_{\lambda_1}=0.1$, $\eta_{\lambda_2}=0.005$; λ bounds in Table 1. Training setup, checkpoint selection criteria, and Pareto frontier construction are detailed in Appendix A.

4 Experimental Setup

Benchmark and evaluation metric. We train and evaluate on HumanEvalComm (Wu and Fard, 2025), a multi-turn extension of HumanEval (Chen et al., 2021) in which each problem specification has been deliberately degraded before being presented to the model.¹ Each episode presents the

¹Seven degradation types: 1a, 1c, 1p, 2ac, 2ap, 2cp, 3acp; numeric prefix = number of simultaneous degradations; letter suffix = type (a = ambiguity, c = contradiction/inconsistency, p = partial truncation).

model with a degraded specification; the model may ask clarifying questions (answered by a GPT-4o-mini simulator from the ground-truth oracle) before submitting a Python solution. Every instance presented to the model is degraded; HumanEvalComm contains no unmodified specifications. Our primary evaluation metric is pass@1 (defined in Section 1). We split 772 degraded instances into 303 for training, 52 for validation, and 417 for held-out testing (split details in Appendix A).

Policy and simulator. We fine-tune Qwen2.5-Coder-7B-Instruct (Hui et al., 2024) using LoRA (Hu et al., 2021)²; both the policy and simulator receive a system prompt (full text and rationale in Appendix E; representative episodes in Appendix D). We train 10 policies at seven budget levels, with two independent runs (different seeds, otherwise identical) at $d_1 \in \{0.5, 0.75, 1.0\}$ to measure run-to-run variance. The baseline is the frozen, untrained model (mean pass@1 = 0.690, $\bar{q} = 0.856$). Notably, the budget d_1 is enforced only during training via the Lagrange multiplier; at evaluation the policy generates freely, and budget compliance is assessed post-hoc from the resulting question rates. Checkpoint selection and training curves are in Appendix A.

5 Results

Budget compliance and the Pareto frontier. Figure 1 plots all trained policies in $(\bar{q}, \text{pass@1})$ space. The key efficiency result is $d_1=0.5$: at $\bar{q} = 0.453$ (roughly half the baseline’s question rate of 0.856), two independent runs achieve mean pass@1 $\in \{0.683, 0.719\}$, neither significantly different from baseline (both $p \geq 0.10$, paired t -test), confirming that the budget constraint is enforceable at half the question rate without degrading mean pass@1. Allowing more questions continues to help: $d_1=0.75$ ($\bar{q} = 0.782$) achieves mean pass@1 = 0.761 (+7.1%, best run; other run +5.8%, both $p < 0.01$), and $d_1=1.0$ ($\bar{q} = 0.859$) achieves mean pass@1 = 0.778 (+8.7%, best run; other run +7.4%, both $p < 0.0001$). At $d_1=1.0$, λ_1 activates at some iterations but does not bind persistently; above $d_1=1.0$, $\lambda_1 \approx 0$ throughout training (the budget is slack, see Appendix B), and

²Artifact licenses: HumanEval is released under the MIT License; Qwen2.5-Coder-7B-Instruct and HumanEvalComm are released under Apache-2.0. The GPT-4o-mini simulator is accessed via the OpenAI API under its terms of use. All artifacts are used in a manner consistent with their intended research use.

further budget increases yield no marginal accuracy gain. In this slack regime, PPO trains without active question-budget pressure yet still improves accuracy over the baseline; these high-budget runs ($d_1 \in \{1.5, 2.0\}$, $\lambda_1 \approx 0$) therefore serve as an approximate unconstrained-PPO control, indicating that PPO itself contributes to the accuracy gains, while the Lagrangian constraint is what provides budget-targeted control and yields the efficient operating point. Across all runs, the turn multiplier λ_2 remained 0 at every dual update: the maximum batch-mean turn count was 3.33, below the turn budget $d_2=4$. The trained policies are therefore behaviorally question-only, so our main claims rest on the question constraint rather than the turn constraint; d_2 serves only as a safeguard against pathological dialogues below the hard cap $T_{\max}=6$.

Diminishing returns and the value of clarification. The Pareto frontier in Figure 1 follows a log-shaped curve: the marginal pass@1 gain per additional clarifying question decreases as the budget grows. The step from $d_1=0.5$ to $d_1=0.75$ yields a substantially larger accuracy gain than the step from $d_1=0.75$ to $d_1=1.0$, and beyond $d_1=1.0$ there is no further marginal accuracy gain. This implies that, beyond a certain budget, the friction cost of each additional clarification outweighs its marginal accuracy benefit.

Among our trained policies, $d_1=0.75$ is the only one to **Pareto-dominate** the baseline: it achieves significantly higher pass@1 (+5.8+7.1%, both $p<0.01$) while asking significantly fewer questions ($\bar{q} \in \{0.719, 0.782\}$ vs. 0.856; both $p<0.01$, paired t -test) across both independent runs. The $d_1=0.5$ policy achieves comparable pass@1 at roughly half the baseline question rate (both $p \geq 0.10$); one run is above baseline and one below, consistent with budget compliance but not a reliable accuracy gain at this budget level. Appendix D (Episode 2) illustrates Pareto domination concretely. Policies at $d_1 \in \{0, 0.25\}$ fall below baseline (details in Appendix A), confirming that suppressing clarification too aggressively is counterproductive.

Performance by degradation complexity. Figure 2 shows pass@1 broken down by degradation complexity group: **1-deg** (types 1a, 1c, 1p; $n=266$) and **2- or 3-deg** (types 2ac, 2ap, 2cp, 3acp; $n=151$). Every policy scores lower on harder multi-degradation problems, where a single clarifying question can resolve at most one simultaneous

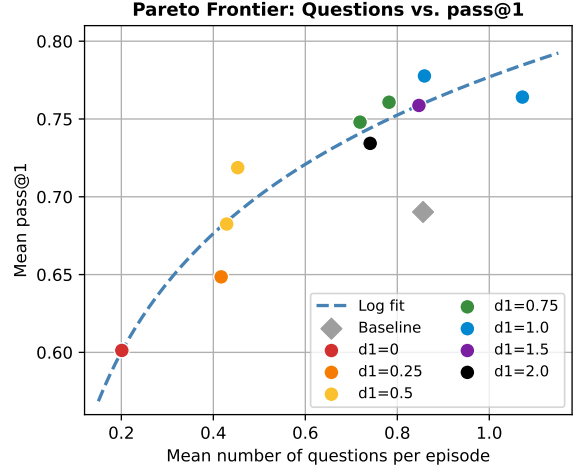


Figure 1: Pareto frontier in $(\bar{q}, \text{pass@1})$ space for all trained policies and the untrained baseline (gray diamond), evaluated on 417 held-out problems under greedy decoding. The log fit illustrates diminishing returns; from $d_1=1.0$ onward Lagrangian enforcement weakens or ceases.

spec defect; trained policies tend to ask more questions on these cases. Question-count breakdowns by group and by all seven degradation types are in Appendix C.

Quality over quantity: better questions and code. Controlling for question count (1Q→1Q episodes, i.e., episodes where both the baseline and the trained policy asked exactly one question), $d_1=0.75$ achieves +7.1% over baseline ($p<0.001$, $n=490$ episodes pooled across both runs, paired t -test), showing gains do not stem from higher question counts. The per-episode breakdown (Appendix C) is consistent with better question targeting and code quality under ambiguity.

6 Conclusion

CMDP post-training with PPO-Lagrangian produces a Pareto frontier in $(\bar{q}, \text{pass@1})$ space: trained policies simultaneously achieve higher accuracy and lower question rates than the untuned baseline. Two structural findings extend this result. First, policies where the Lagrangian is actively enforced ($d_1 \in \{0.5, 0.75\}$, $\lambda_1>0$) ask more questions on harder multi-degradation problems while asking fewer questions overall than the baseline (Figure 2 and Appendix C), suggesting that CMDP training reallocates question budget toward more informative contexts rather than reducing it. Second, gains are driven by question quality and code generation ability rather than question count: trained policies significantly outperform the baseline even

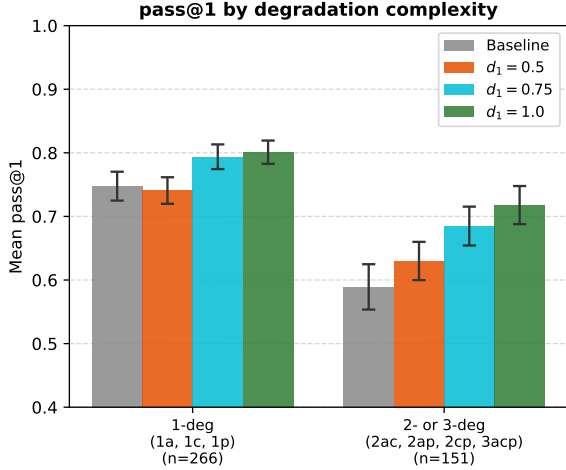


Figure 2: Mean pass@1 by degradation type (417 problems, greedy decoding). $d_1 \in \{0.5, 0.75, 1.0\}$ results averaged across two independent runs. Error bars are ± 1 standard error of the mean (SEM).

when both ask exactly one question.

The Pareto-dominant operating point ($d_1=0.75$, within the binding regime where $\lambda_1 > 0$) demonstrates that active Lagrangian enforcement achieves simultaneous quality and efficiency gains. The log-shaped Pareto frontier further suggests a practical design principle: there is a budget threshold beyond which the friction cost of additional clarifications outweighs their marginal accuracy benefit, and operating beyond it yields no gain. Future work should examine this framework on naturally occurring ambiguity, larger model scales, and human rather than simulator oracles. A natural extension is a *dynamic* budget that adapts to the estimated complexity or ambiguity of each problem, rather than enforcing a fixed expected question count across all episodes.

Limitations

Run-to-run variance. Independent training runs at the same budget ($d_1 \in \{0.5, 0.75, 1.0\}$) converge to notably different mean pass@1 values (e.g., 0.683 vs. 0.719 at $d_1=0.5$; 0.778 vs. 0.764 at $d_1=1.0$), indicating sensitivity to random initialisation. This limits the reliability of conclusions drawn from any single run and motivates future work on more stable training procedures.

Training scale and instability. All policies were trained under significant compute constraints: only 80 iterations with a mini-batch size of 8 episodes and 32 episodes per rollout iteration. Small batches produce high-variance gradient estimates, which destabilises the dual-ascent updates for the La-

grange multipliers and likely contributes to the run-to-run variance noted above. Larger batch sizes, more iterations, and learning-rate schedules tuned for the Lagrangian dynamics would likely improve both convergence reliability and final performance.

LLM-based question simulator. The oracle that answers clarifying questions is GPT-4o-mini running at temperature 0, making its responses deterministic given the question and ground-truth specification. This eliminates answer inconsistency as a noise source, but a greedy deterministic simulator may not reflect the variability of real human responses: a human might paraphrase, hedge, or volunteer adjacent information differently across interactions, all of which would affect the policy’s learned question-asking strategy.

Single benchmark and small training set. All results are on HumanEvalComm, a single code-generation benchmark with a specific set of synthetic degradation types. Whether the learned strategies and efficiency gains generalise to other task types or to naturally occurring ambiguity is untested. The training pool is also small: 303 degraded instances from 64 base problems, smaller than the 417-problem held-out test set. Whether this scale is sufficient for reliable convergence is unclear, and likely contributes to the run-to-run variance observed across seeds.

Scale. Experiments use a 7B-parameter model with LoRA fine-tuning. Whether the CMDP approach scales to larger models, whether the emergent budget-allocation behaviour persists, and whether the Lagrange multiplier dynamics remain stable at larger scale are open questions.

Discrete budget coverage. Due to compute constraints, we train at only seven discrete budget values ($d_1 \in \{0, 0.25, 0.5, 0.75, 1.0, 1.5, 2.0\}$); the log fit in Figure 1 interpolates between them. The true shape of the Pareto frontier between trained budgets is unverified; the optimal trade-off curve could differ meaningfully from the fitted log curve in these intervals.

Disentangling question quality from code quality. PPO training jointly optimises question-asking and code generation, making it difficult to attribute gains solely to improved clarification behaviour. Cases where a policy asks comparable questions to the baseline yet produces better code suggest that general coding ability also improves

under PPO. Future work should design controlled ablations, for instance freezing the code-generation head and training only the question-asking policy, to cleanly separate these two sources of improvement.

Acknowledgments

The authors used Claude (Anthropic) for assistance with writing and editing during this project. All experimental design, analysis, implementation, math, and final verification were performed by the authors.

References

- Mohammad Aliannejadi, Hamed Zamani, Fabio Crestani, and W. Bruce Croft. 2019. [Asking clarifying questions in open-domain information-seeking conversations](#). *CoRR*, abs/1907.06554.
- Eitan Altman. 1999. *Constrained Markov Decision Processes*. Chapman & Hall/CRC, Boca Raton, FL.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. [Evaluating large language models trained on code](#). *Preprint*, arXiv:2107.03374.
- Jeremy Curuksu. 2024. [Policy optimization to align the fidelity and efficiency of reasoning agents in multi-turn dialogues](#). In *NeurIPS 2024 Workshop on Open-World Agents*.
- Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2023. [Safe rlhf: Safe reinforcement learning from human feedback](#). *Preprint*, arXiv:2310.12773.
- Harsh Darji and Thibaud Lutellier. 2025. [Curiosity by design: An llm-based coding assistant asking clarification questions](#). *Preprint*, arXiv:2507.21285.
- Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2022. [8-bit optimizers via block-wise quantization](#). *Preprint*, arXiv:2110.02861.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. [Lora: Low-rank adaptation of large language models](#). *Preprint*, arXiv:2106.09685.
- Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, Kai Dang, Yang Fan, Yichang Zhang, An Yang, Rui Men, Fei Huang, Bo Zheng, Yibo Miao, Shanghaoran Quan, and 5 others. 2024. [Qwen2.5-coder technical report](#). *Preprint*, arXiv:2409.12186.
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. [Training language models to follow instructions with human feedback](#). *Preprint*, arXiv:2203.02155.
- Sudha Rao and Hal Daumé III. 2018. [Learning to ask good questions: Ranking clarification questions using neural expected value of perfect information](#). In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2737–2746, Melbourne, Australia. Association for Computational Linguistics.
- John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2016. [High-dimensional continuous control using generalized advantage estimation](#). In *International Conference on Learning Representations*.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. [Proximal policy optimization algorithms](#). *Preprint*, arXiv:1707.06347.
- Adam Stooke, Joshua Achiam, and Pieter Abbeel. 2020. [Responsive safety in reinforcement learning by PID lagrangian methods](#). In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9133–9143. PMLR.
- Weiwei Sun, Xuhui Zhou, Weihua Du, Xingyao Wang, Sean Welleck, Graham Neubig, Maarten Sap, and Yiming Yang. 2025. [Training proactive and personalized llm agents](#). *Preprint*, arXiv:2511.02208.
- Manan Suri, Puneet Mathur, Nedim Lipka, Franck Dernoncourt, Ryan A. Rossi, and Dinesh Manocha. 2026. [Structured uncertainty guided clarification for LLM agents](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 40811–40838, San Diego, California, United States. Association for Computational Linguistics.
- Chen Tessler, Daniel J. Mankowitz, and Shie Mannor. 2018. [Reward constrained policy optimization](#). *Preprint*, arXiv:1805.11074.
- Sanidhya Vijayvargiya, Xuhui Zhou, Akhila Yerukola, Maarten Sap, and Graham Neubig. 2026. [Ambig-SWE: Interactive agents to overcome underspecificity in software engineering](#). In *The Fourteenth International Conference on Learning Representations*.
- Issei Waki, Ryu Takeda, and Kazunori Komatani. 2025. [Learning to ask efficiently in dialogue: Reinforcement learning extensions for stream-based active learning](#). In *Proceedings of the 26th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 431–440, Avignon, France. Association for Computational Linguistics.

Jie JW Wu and Fatemeh H. Fard. 2025. [Humaneval-comm: Benchmarking the communication competence of code generation for llms and llm agent](#). *ACM Transactions on Software Engineering and Methodology*.

Michael JQ Zhang and Eunsol Choi. 2025. [Clarify when necessary: Resolving ambiguity through interaction with LMs](#). In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 5541–5558, Albuquerque, New Mexico. Association for Computational Linguistics.

Zhirui Zhang, Xiujuan Li, Jianfeng Gao, and Enhong Chen. 2019. [Budgeted policy learning for task-oriented dialogue systems](#). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3742–3751, Florence, Italy. Association for Computational Linguistics.

Appendix

A Implementation Details

Table 1 consolidates all hyperparameters used across all training runs. All policies use identical hyperparameters; only the budget d_1 varies.

CMDP formal definitions. **State.** At turn t , s_t is the full conversation history: the degraded problem specification x together with all prior question-answer pairs $\{(q_1, u_1), \dots, (q_{t-1}, u_{t-1})\}$. **Transition function.** For [ANSWER] actions, the episode terminates deterministically. For [ASK] actions, s_{t+1} incorporates the GPT-4o-mini user simulator’s response to q_t ; the simulator runs at temperature = 0, making its responses deterministic given the question and ground-truth specification. **Reward.** $R(s_t, a_t) = \rho(c_t)$ if $a_t = [\text{ANSWER}]$ c_t , else 0, where $\rho(c_t) \in [0, 1]$ is pass@1: the fraction of hidden test assertions passed. Reward is non-zero only at the terminal step. The training signal optimizes pass@1.

Policy and value heads. We parameterize the policy as Qwen2.5-Coder-7B-Instruct (Hui et al., 2024) with LoRA (Hu et al., 2021) ($r = 16$, $\alpha = 32$) applied to all seven attention and MLP projection layers ($\approx 40\text{M}$ of 7B parameters trainable). Three value heads V^R , V^{C_q} , V^{C_t} , each implemented as $\text{Linear}(h, 1024) \rightarrow \text{GELU} \rightarrow \text{LayerNorm}(1024) \rightarrow \text{Linear}(1024, 1)$ over the final-layer last-prompt-token hidden state, estimate expected future returns for the reward, question-cost, and turn-cost streams. Output layers are zero-initialized; value MSE gradients backpropagate through the shared LoRA representation. A frozen reference copy (base weights, no LoRA)

Component	Hyperparameter	Value
Policy (LoRA)	Rank r	16
	Scale α	32
	Trainable parameters	$\approx 40\text{M}$
	Learning rate	5×10^{-6}
	LR warmup	20 steps (linear)
Value heads	Hidden dim	1024
	Activation	GELU
	Learning rate	10^{-4}
	Output layer init	zeros
PPO	Clip ϵ	0.2
	Mini-batch size	8
	Max epochs / iteration	4
	KL early-stop threshold	0.05
	β_{KL}	0.25
	β_H	0.01
	Value loss coefficient	0.5
GAE	γ	1.0
	λ_{GAE}	0.95
Lagrangian	η_{λ_1}	0.1
	η_{λ_2}	0.005
	$\lambda_{1,\text{max}}$	10
	$\lambda_{2,\text{max}}$	5
	Turn budget d_2	4
Training	Episodes / iteration	32
	Total iterations	80
	Checkpoint frequency	every 5 iterations
	Rollout temperature	0.8
Episode	Eval temperature	0 (greedy)
	Max turns T_{max}	6
	Optimizer	8-bit AdamW
Gradient clipping	ℓ_2 norm	1.0

Table 1: Hyperparameters used for all training runs.

runs on a second GPU for KL regularization; a rollout copy (LoRA weights synced each iteration) handles episode collection. Both π_θ and $\pi_{\theta_{\text{old}}}$ are computed via clean forward passes, bypassing HuggingFace logit processors that would otherwise corrupt log-probability scales.

PPO surrogate and prefix exclusion. The policy is updated via the clipped PPO objective (Schulman et al., 2017) ($\epsilon = 0.2$):

$$L^{\text{PPO}} = -\mathbb{E}_t \left[\min \left(r_t \hat{A}_t^{\text{lag}}, \text{clip}(r_t, 1-\epsilon, 1+\epsilon) \hat{A}_t^{\text{lag}} \right) \right]. \quad (6)$$

where

$$r_t(\theta) = \frac{\pi_\theta(a_{\text{cont},t} \mid s_t, \text{pref}_t)}{\pi_{\theta_{\text{old}}}(a_{\text{cont},t} \mid s_t, \text{pref}_t)}$$

is the probability ratio over *continuation tokens only*; the forced [ASK]/[ANSWER] prefix is excluded from both numerator and denominator.

Routing gradient. Routing and generation share the same LoRA weights: the prefix scores used to select [ASK] vs. [ANSWER] at rollout time are computed from the same attention and MLP projections that PPO updates via continuation-token gradients. When λ_1 rises, the $-\lambda_1 \tilde{A}_t^{C_q}$ term penalises [ASK]-turn continuations; the resulting weight updates shift the model’s logits at the routing step in the next iteration. Budget enforcement thus propagates to routing through shared parameters across training iterations, which is the standard mechanism by which PPO shapes structured generation behaviour. As a concrete two-token example, suppose the policy emits [ASK] what type?. The [ASK] prefix tokens are excluded from the ratio r_t , so the gradient never directly rewards or penalises the *decision* to ask; only the continuation what type? enters the loss. Yet because the prefix logits and the continuation are produced by the same shared LoRA weights, updating those weights to make a useful what type? more likely also perturbs the logits that produced [ASK]: the routing decision is coupled to the continuation gradient and is shaped as a side effect of it, rather than being fixed. The routing decision is therefore learned indirectly, through the shared parameters, rather than optimised directly against the sparse pass@1 signal, which would otherwise attribute the entire episode’s outcome to a single decision token.

Episode protocol. Each episode begins with the agent receiving a degraded problem specification. At each turn: (1) the agent generates an action that begins with [ASK] or [ANSWER], selected via prefix-scored routing; (2) **if [ASK]**, the question is passed to the GPT-4o-mini user simulator, which holds the complete specification and returns a factual answer together with an atomic question count N , setting $C_q = N$; (3) **if [ANSWER]**, code is executed against hidden tests, yielding $\rho = k/n$ (k of n assertions passing), and the episode terminates. Episodes reaching $T_{\max} = 6$ terminate with $R = 0$.

Atomic question counting. The simulator appends QUESTION_COUNT: N to its reply, where N is the number of distinct information requests in the [ASK] turn. Multi-question turns are charged $C_q = N > 1$, preventing budget exploitation by bundling questions.

Prefix-scored routing. Given s_t , the model performs a forward pass over the prompt and scores each prefix via log-softmax over the final-

position logits. Under the Qwen2.5-Coder tokenizer, [ASK] and [ANSWER] each tokenize as four tokens but share the first ($\square$), so the model performs a second forward step via KV cache to score the second distinguishing token of each prefix (ASK vs. ANSWER). The prefix with the higher two-token sum of log-softmax values is selected; this is a heuristic approximation to the full four-token joint log-probability and is distinct from hard-constraint decoding algorithms that guarantee structural constraints (tries, finite-state acceptors, grammar-constrained decoding). At evaluation (temperature = 0), the higher-scoring prefix is selected greedily; during training rollouts (temperature = 0.8), the prefix is sampled from a Boltzmann distribution over the two scores. Ties at greedy decoding default to [ASK]. The model generates freely after the chosen prefix.

GAE recurrence (Schulman et al., 2016). Three advantage streams are computed via GAE for $X \in \{R, C_q, C_t\}$:

$$\begin{aligned}\delta_t^X &= x_t + \gamma V^X(s_{t+1}) - V^X(s_t), \\ A_t^X &= \sum_{l \geq 0} (\gamma \lambda_{\text{GAE}})^l \delta_{t+l}^X.\end{aligned}\quad (7)$$

where x_t is the stream-specific signal at turn t : $R(s_t, a_t)$ for $X=R$, $C_q(a_t)$ for $X=C_q$, and $C_t(a_t)$ for $X=C_t$. The $\gamma=1.0$, $\lambda_{\text{GAE}}=0.95$ setting introduces an effective per-turn discount of 0.95^τ on advantages at turn depth τ from the episode end.

Double whitening. Let $\hat{A}_t^{\text{lag}} = \text{whiten}(A_t^{\text{lag}})$ denote the result of whitening A_t^{lag} once more before the PPO surrogate for numerical stability; this second pass scales all terms by $1/\sqrt{1 + \lambda_1^2 + \lambda_2^2}$ (under stream independence), so λ_1 shifts the advantage threshold rather than amplifying the absolute gradient magnitude of the constraint term.

Full training objective.

$$L(\theta) = L^{\text{PPO}} + \beta_{\text{KL}} L_{\text{KL}} - \beta_H L_H + 0.5 L^{\text{vf}}, \quad (8)$$

where $L_{\text{KL}} = \max(0, \mathbb{E}[\log \pi_\theta - \log \pi_{\text{ref}}])$ is a sequence-level KL from the frozen reference (clamped non-negative, $\beta_{\text{KL}} = 0.25$); $L_H = -\mathbb{E}[\log \pi_\theta / |a_{\text{cont}}|]$ a length-normalized entropy bonus ($\beta_H = 0.01$); and L^{vf} total value MSE across all three heads. All terms are over continuation tokens only. Approx-KL is $\hat{D}_{\text{KL}} = \mathbb{E}[(r_t - 1) - \log r_t]$; if $\hat{D}_{\text{KL}} > 0.05$ per mini-batch, remaining epochs for that iteration are skipped. Gradient ℓ_2 norm is clipped to 1.0.

Dual update mechanics. The λ values entering $\hat{A}^{\text{lag}}$ (via Eq. (3)) within the PPO update are therefore the post-dual-update multipliers of the current iteration. When $\bar{q} > d_1$, λ_1 rises, shifting the sign threshold in $\hat{A}^{\text{lag}}$: some [ASK] continuations that ranked as positive flip to negative, making them less likely to be reinforced by PPO. The absolute gradient scale is fixed by the second whitening and is not affected by λ_1 .

Per-iteration training steps. Each PPO-Lagrangian iteration proceeds as: (1) sync LoRA weights to the rollout model; (2) sample a batch of 32 episodes from the training pool (user-simulator API calls concurrent via `asyncio`); (3) compute GAE for all three streams; (4) update λ_1 , λ_2 via dual ascent (Eqs. (4)–(5)); (5) run up to 4 PPO epochs with early-exit on approx-KL; (6) step the LR scheduler. Policy and value heads use 8-bit AdamW (Dettmers et al., 2022) (LR 5×10^{-6} and 10^{-4} ; 20-step linear warmup for the policy, constant thereafter).

Train/eval split. All degraded variants of the same base HumanEval problem are kept in the same split, preventing the model from seeing the ground-truth specification of an eval problem in a different degraded form during training. A fixed set of 52 validation problems (stratified by degradation type, sampled with a fixed seed) is used for all policies, leaving 417 held-out problems for the main eval results reported here.

Training setup. Training uses 303 degraded instances from the 64 base problems not held out for evaluation (not all degradation types are available for every base problem; rare types such as 3acp exist for only 12 of 164 base problems), running 80 PPO-Lagrangian iterations per budget (≈ 11 h on $2 \times \text{A100-40GB}$; hyperparameters in Table 1). The reward signal is exclusively pass@1.

Checkpoint selection. Checkpoints are saved every 5 iterations; the 52-problem validation set (stratified by degradation type, seed 99) is evaluated under greedy decoding at every 10th checkpoint (iter 9, 19, ..., 79). A checkpoint *satisfies* the budget if $\bar{q}^{\text{val}} \leq d_1$ on the val set under greedy decoding; the primary criterion selects the highest val pass@1 among satisfying checkpoints. If no satisfying checkpoint exists (as at $d_1 = 0$), the fallback selects the checkpoint with the lowest $\bar{q}^{\text{val}}$; the $d_1 = 0$ result therefore reflects training *toward* zero clarification, not a converged zero-question

policy. The $d_1 = 0.25$ run is excluded from the frontier: it satisfied the budget at validation but violated it at test time ($\bar{q}^{\text{test}} = 0.417 > 0.25$).

Pareto frontier and evaluation. All seven budget levels are studied; all runs use identical hyperparameters, varying only d_1 . All policies are evaluated on the 417-problem held-out test set (Section 4); statistical comparisons use paired t -tests per run. The baseline is untuned Qwen2.5-Coder-7B-Instruct under identical multi-turn conditions (prefix-scored routing, $T_{\text{max}} = 6$, greedy decoding). Budget compliance is assessed per run on the test set via one-sample t -test ($H_0: \mu_q \leq d_1$, $\alpha=0.05$). At $d_1=0.75$, neither run significantly exceeds the budget (run 1: $\bar{q}=0.782$, $p=0.15$; run 2: $\bar{q}=0.719$, $p=0.86$); both are statistically budget-compliant ($n=417$, $\text{SE} \approx 0.03$). At $d_1=1.0$, run 1 satisfies the budget ($\bar{q}=0.859$, well below budget), while run 2 significantly exceeds it ($\bar{q}=1.072$, $p=0.027$). The $d_1 = 0$ policy falls below the baseline and is dominated by $d_1 = 0.5$; it does not Pareto-dominate the baseline. The reported frontier ($d_1 \in \{0.5, 0.75, 1.0, 1.5, 2.0\}$) is anchored to the budget-satisfying run at each level; gains from budget-violating runs are reported separately for completeness.

$d_1 = 0.25$ run details. The single run reached $\bar{q}^{\text{val}} = 0.25$ at iter_0079 (val pass@1 = 0.481, budget-feasible at val selection), but test-time $\bar{q} = 0.417 > 0.25$; it is excluded from the Pareto frontier because $\bar{q}^{\text{test}} > d_1$.

B Training Curves

Figures 3–5 show training dynamics for representative policies at $d_1 \in \{0.5, 0.75, 1.0\}$. These three budgets span the efficient region of the frontier: $d_1=0$ never achieves budget compliance within 80 iterations and the single $d_1=0.25$ run violates the budget at test time, budgets strictly above $d_1 = 1.0$ show $\lambda_1 = 0$ throughout training (the constraint never binds), and $\{0.5, 0.75, 1.0\}$ are the only budgets trained with two independent runs. Each figure shows four panels across the 80 training iterations: (1) mean episode reward, (2) mid-training validation pass@1 evaluated every 10 iterations with the selected checkpoint marked, (3) mean questions per episode $\bar{q}$ with the budget d_1 as a dashed reference line, and (4) the Lagrange multiplier λ_1 .

The λ_1 panels directly confirm the budget compliance story from Section 5: for $d_1 = 1.0$, λ_1

activates at some iterations but does not persist throughout training, while for $d_1 \in \{0.5, 0.75\}$, λ_1 rises early and stabilises at a persistently positive value, reflecting the dual-ascent feedback loop enforcing the budget.

C Degradation Type Breakdown

At $d_1=1.0$, mean pass@1 rises on zero-question episodes (73.7% baseline $\rightarrow$ 81.1%) and on one-question episodes (67.8% $\rightarrow$ 77.4%), consistent with the model learning to write better code under ambiguity and to formulate more targeted questions when clarification is warranted.

Figures 7 and 8 show pass@1 broken down by all seven individual degradation types (rows) and question count: 0 questions, exactly 1 question, and 2+ questions (columns). For each policy, episodes are partitioned per run by that run’s question count and pooled across both runs before computing means. This finer-grained view reveals heterogeneous behaviour within the 1-deg group that is obscured in the aggregate Figures 2 and 6: ambiguity problems (1a) are near-ceiling for all models regardless of question count; contradiction problems (1c) show consistent gains for trained policies in both the 0Q and 1+Q columns, reflecting improved code-generation quality for contradictory specs; and truncation problems (1p) show that the 0Q performance gap between trained policies and the baseline is largely a selection effect i.e. trained policies have learned to skip asking on the easier truncation cases, while the baseline skips asking at random.

Training Curves: $d_1=0.5$ (run 2) (budget $d_1 = 0.5$, 80 iterations, best=iter_0059)

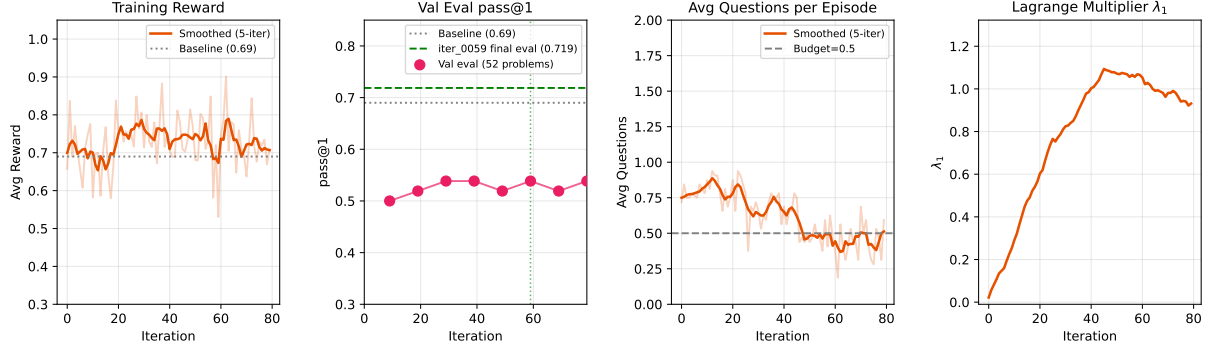


Figure 3: Training curves for $d_1 = 0.5$.

Training Curves: $d_1=0.75$ (run 1) (budget $d_1 = 0.75$, 80 iterations, best=iter_0039)

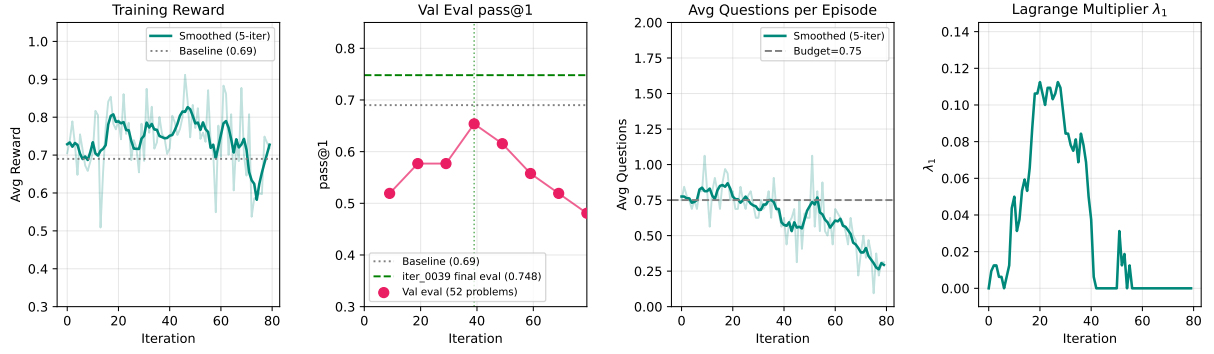


Figure 4: Training curves for $d_1 = 0.75$.

Training Curves: $d_1=1.0$ (run 1) (budget $d_1 = 1.0$, 80 iterations, best=iter_0039)

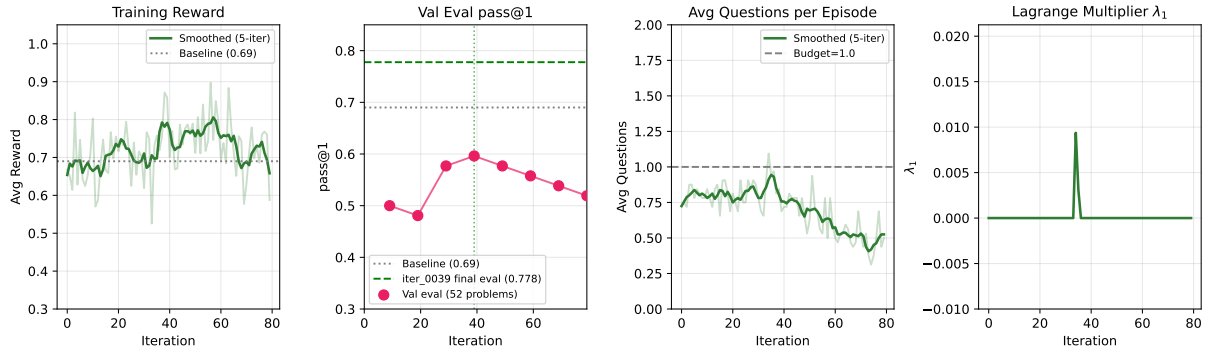


Figure 5: Training curves for $d_1 = 1.0$.

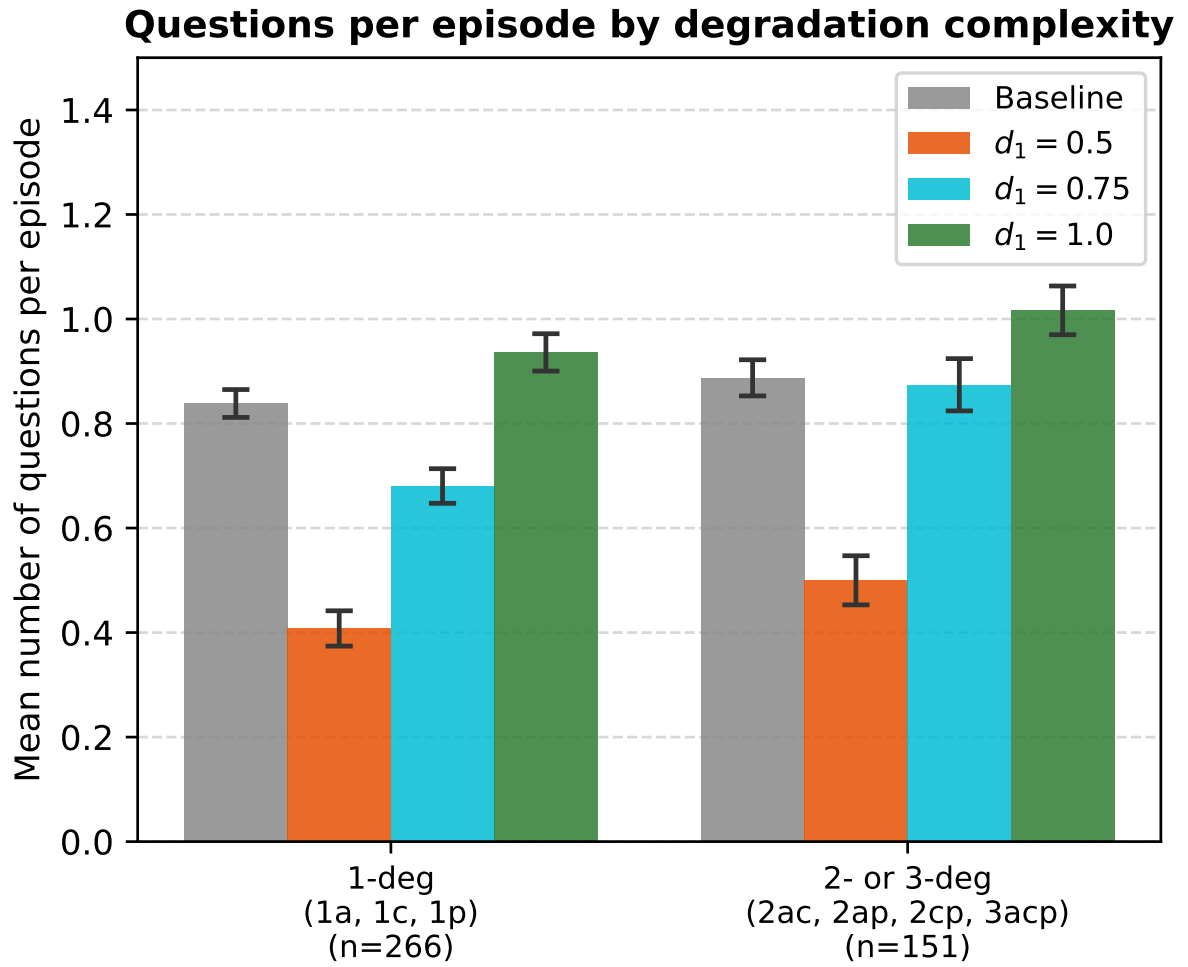


Figure 6: Mean number of questions per episode by degradation complexity group, evaluated on the 417-problem held-out set under greedy decoding. Settings match Figure 2.

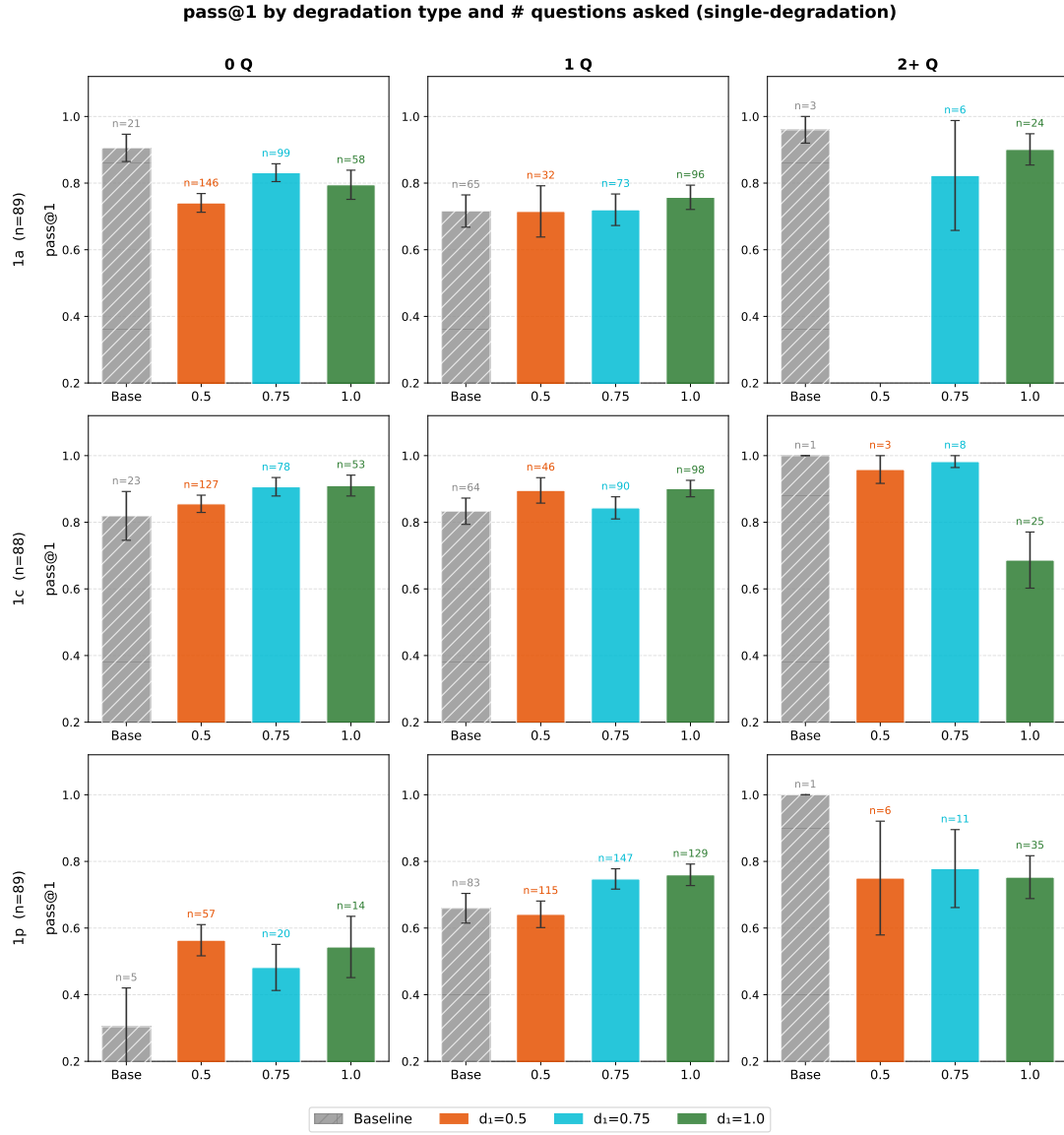


Figure 7: pass@1 by degradation type and number of questions asked, for single-degradation types (1a, 1c, 1p). Columns correspond to episodes where 0, exactly 1, or 2+ clarifying questions were asked. Per-run pooling: for policies with two independent runs, each run contributes its episodes independently, classified by that run’s question count, then pooled before computing means. Error bars are ± 1 SEM.

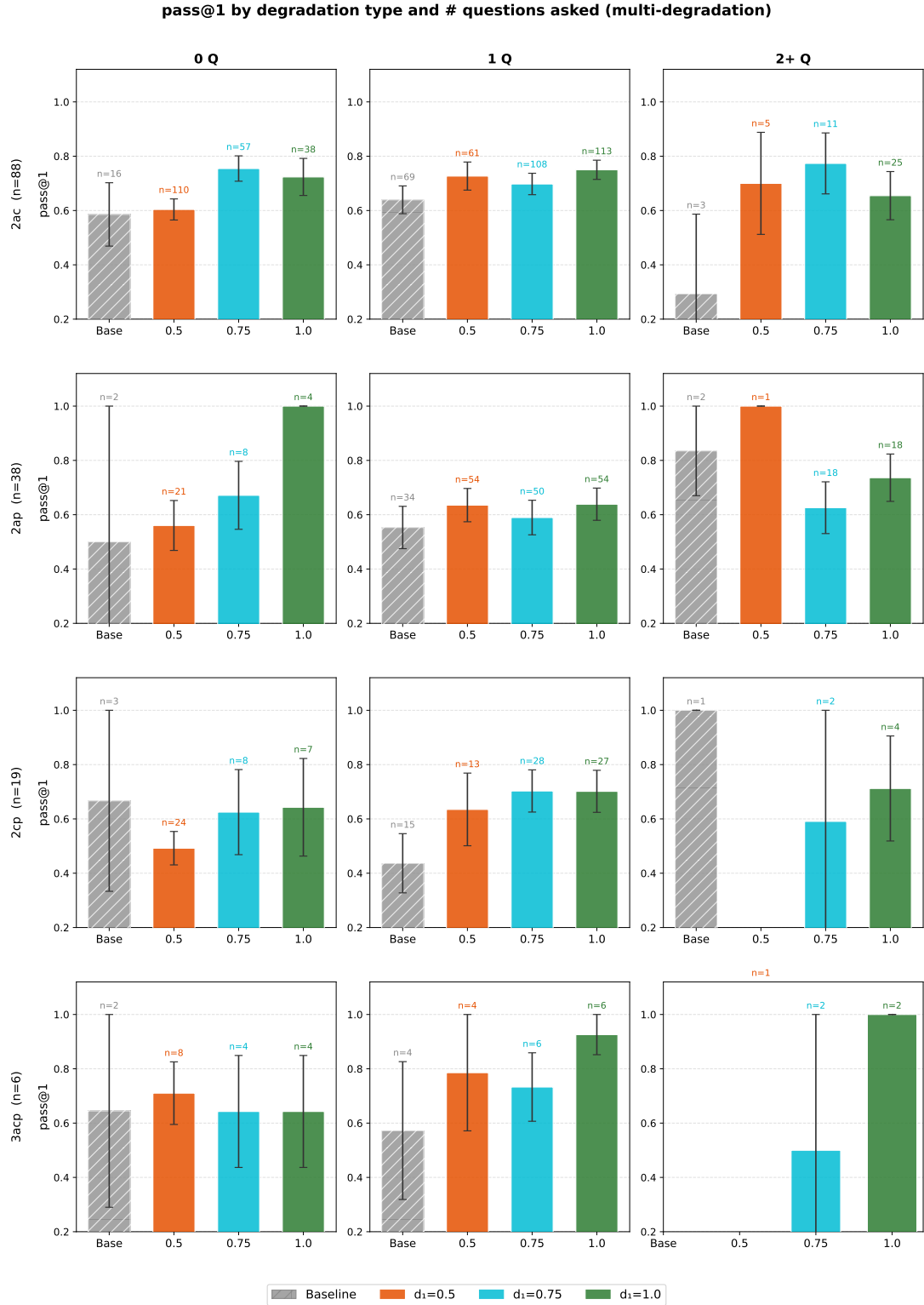


Figure 8: pass@1 by degradation type and number of questions asked, for multi-degradation types (2ac, 2ap, 2cp, 3acp). Same layout and pooling as Figure 7. Cells with very small n (notably 3acp and the 2+ Q column) should be interpreted cautiously.

D Representative Episode Comparisons

Each episode below shows the degraded specification presented to the agent, followed by the conversation. All episodes are from the 417-problem held-out evaluation set under greedy decoding (temperature = 0).

Episode 1. Interaction format; targeted vs. edge-case questioning (HumanEval/133/prompt1p, type: 1p). The spec body is truncated mid-sentence; the task description is entirely missing.

Degraded Spec

```
def candidate(lst):
    """You are given a list of numbers.
    You need to return the sum of squared
    """
```

Baseline (pass@1 = 0.00, 1 question):

Turn 1 [ASK]

What should be the behavior if the input list is empty?
Should it return 0 or raise an error?

Simulator

The behavior for an empty input list is not specified in the provided documentation.

Turn 2 [ANSWER]

0

Trained policy $d_1=1.0$ (pass@1 = 1.00, 1 question):

Turn 1 [ASK]

What is the task? Specifically, what should be done with the list of numbers?

Simulator

The task is to return the sum of the squared numbers in the given list after rounding each element to the upper integer (ceiling) first.

Turn 2 [ANSWER]

```
import math
def candidate(lst):
    return sum(math.ceil(x) ** 2 for x in lst)
```

The baseline does not yet know what the function is supposed to compute, yet asks about an edge case, a question that is only meaningful once the core task is understood. It receives an uninformative answer and submits a literal 0. The trained policy identifies that the *entire task description* is missing and recovers it in one targeted question.

Episode 2. Pareto efficiency: fewer questions, higher accuracy (HumanEval/45/prompt1p, type: 1p). The spec description is truncated to a single ambiguous word: the docstring reads only "Given length of a side and high, return.", omitting the return type and leaving high undefined. The baseline spends its question budget confirming the obvious (high = height), but its answer still fails; the $d_1=0.5$ policy answers immediately and correctly.

Degraded Spec

```
def triangle_area(a, h):
    """Given length of a side and high,
    return.
    """
```

Baseline (pass@1 = 0.00, 1 question):

Turn 1 [ASK]

Is the high meant to be the height of the triangle?

Simulator

Yes, in this context, 'high' refers to the height of the triangle.

Turn 2 [ANSWER]

$0.5 * a * h$

Trained policy $d_1=0.5$ (pass@1 = 1.00, 0 questions):

Turn 1 [ANSWER]

```
def triangle_area(a, h):
    return 0.5 * a * h
```

The baseline wastes its question budget confirming that high means height, information that does not change the solution, and then outputs a bare expression instead of a function definition, scoring 0. The trained policy infers the formula directly from the signature and returns a complete, correct function without asking anything. This is one concrete instance of the Pareto dominance shown in Figure 1.

Episode 3. Multi-degradation problems require more questions (HumanEval/126/prompt2ap, type: 2ap). The spec has two simultaneous degradations: the sort direction is ambiguous *and* the duplicate-handling rule is truncated. A single question can resolve at most one.

Degraded Spec

```
def candidate(lst):
    '''
    Given a list of numbers, return whether
    or not they are
    sorted in a specific order. If list has
    certain duplicate
    of the same number,
    '''
```

Baseline (pass@1 = 0.85, 1 question):

Turn 1 [ASK]

What is the specific order that the list should be sorted in?

Simulator

The list should be sorted in ascending order.

Turn 2 [ANSWER]

```
def candidate(lst):
    return lst == sorted(lst) # misses
    duplicate rule entirely
```

Trained policy $d_1=1.0$ (pass@1 = 1.00, 2 questions):

Turn 1 [ASK]

1. What is the specific order the list should be sorted in?
2. How should the function handle lists with duplicate numbers?

Simulator

1. The list should be sorted in ascending order.
2. The function should return False if the list has more than 1 duplicate of the same number.

Turn 2 [ANSWER]

```
def candidate(lst):
    if lst != sorted(lst):
        return False
    from collections import Counter
    counts = Counter(lst)
    return all(c <= 2 for c in counts.values())
```

The baseline resolves the ambiguous sort direction but never reaches the truncated duplicate rule, producing a partially correct solution. The trained policy recognises both signals of missing information: the unspecified sort direction *and* the visibly cut-off duplicate sentence (“If list has certain duplicate of the same number,”), and addresses both in a single two-part question, recovering the complete specification. This directly illustrates why Figure 6 shows higher mean question counts on harder multi-degradation problems.

E System Prompts

The policy receives a minimal zero-shot system prompt instructing it to respond with either [ASK] followed by a clarifying question or [ANSWER] followed by Python code, with one action permitted per turn. We deliberately avoided few-shot examples in the policy prompt: providing demonstrations would risk biasing the policy toward particular question styles or strategies, confounding the learned behaviour with the choice of examples. The GPT-4o-mini oracle simulator receives a separate system prompt that grants it access to the full ground-truth specification and instructs it to answer only what is asked, without volunteering test cases or the solution.

The placeholder {original_prompt} in the oracle simulator prompt is filled at runtime with the ground-truth un-degraded problem specification.

Policy system prompt

You are a coding assistant. Given a coding task below, you must either:

- Ask a clarifying question by responding with [ASK] followed by your question.
- Write your Python solution by responding with [ANSWER] followed by the code.

Important rules:

- Respond with ONLY one action per turn ([ASK] or [ANSWER]).
- When you have enough information, write the code.
- Do not explain your reasoning -- just output the action directly.

Oracle simulator system prompt (count mode)

You are a helpful assistant who holds the complete specification for a coding problem.

Full specification:
{original_prompt}

Note: The agent may refer to the function by a different name (e.g., "candidate"). Treat any function name the agent uses as referring to the function described above.

Rules:

1. Answer ONLY the specific question(s) the agent asks.
Do not volunteer extra information.
2. Do not reveal test cases or the full solution.
3. Keep your answer brief and factual.
4. After your answer, on a new line write EXACTLY:
QUESTION_COUNT: N
where N is the number of distinct atomic questions you identified in the agent's message.

Counting rules for N:

- Each distinct piece of information being requested = 1.
- "What is X and what is Y?" = 2.
- "What is X?" = 1.
- Conjunctions like "and", "also", "additionally" between separate requests = separate questions.
- "Describe the full behavior of X" where X is one concept = 1.

An Empirical Study on Improving an LLMs based Interviewer against Human Interviewees

Rikuto Tsuchida, Ryo Hasegawa, Tetsuya Sonoda, Takehito Utsuro

Graduate School of Science and Technology, University of Tsukuba

{s2520796, s2420791, s2630267}@u.tsukuba.ac.jp,
utsuro_iit.tsukuba.ac.jp

Abstract

This paper presents a semi-structured interview system composed of multiple LLM components, in which an LLM-based interviewer conducts dialogue and another LLM estimates interviewee persona attributes from the dialogue history and dynamically updated slot information. The system prioritizes questions related to important yet unconfirmed attributes to support efficient interviews. We evaluate persona attribute estimation performance when interviewing both human participants and LLM-based pseudo interviewees under zero-shot and few-shot prompting conditions. Experiments across three domains (IT engineer, store staff, and school teacher) with a total of 39 interviews show that persona attribute estimation is more challenging for human interviewees than for the LLM pseudo interviewees under zero-shot prompting. However, introducing few-shot prompting consistently improves estimation performance for human interviewees, primarily by increasing recall. On the other hand, its effect on the LLM pseudo interviewees is mixed across domains. A case study confirms that persona attribute estimation improves when illustrative examples are provided.

1 Introduction

With the advancement of Large Language Models (LLMs) (Brown et al., 2020), dialogue systems capable of natural conversation with humans are being increasingly utilized in scenarios such as interviews and customer service.

This paper builds on prior work (Hasegawa et al., 2025) that conducted semi-structured interviews (Fielding, 2003a,b; Wengraf, 2001) with an LLM-based user simulator as a pseudo interviewee. We explain semi-structured interviews in Section 3. As discussed in Section 2, that study showed that the proposed system can effectively carry out such interviews with the simulator.

However, the simulator answers from predefined persona settings and replies “I don’t know” to questions outside that scope, whereas human interviewees often use vague expressions, include multiple pieces of information in a single utterance, deviate from the intended topic, and rephrase or shift perspective rather than frankly admitting a lack of knowledge (Section 5). These more diverse dialogues are not identical to unconstrained real-world interviews, but they are closer to them than simulator dialogues and therefore provide a more informative test of interview systems. The objective of this paper is to improve the ability to accurately estimate a subject’s persona attributes based on the conversation history when conducting interviews with human participants.

In this paper, we use the semi-structured interview approach shown in Figure 1 and conduct interviews with both human interviewees and LLM-based pseudo interviewees.

Section 3 presents the proposed interview system, and Section 4 describes the systems used in this paper and how the LLMs estimate persona attributes and values. In the proposed system, an LLM responsible for estimating the interviewee’s persona attributes infers the interviewee’s attributes at each dialogue turn based on the dialogue history and information recorded in slots. This functionality enables the system to identify important information among the information obtained through the interview, allowing prioritized presentation of questions related to important persona attributes, thereby achieving more efficient interviews.

In Section 7, to verify the effectiveness of this system for human interviewees, we evaluate the performance of the LLM responsible for estimating persona attributes. Specifically, we evaluate the performance in interviews with both human interviewees and LLM-based pseudo interviewees, and assess the persona attribute estimation performance achieved by introducing few-shot prompting

with examples (Figure 2, Figure 3, Figure 5 in Section A of Appendix). Furthermore, we conduct case analysis on instances where results improved with the introduction of few-shot prompting. The contributions of this paper are as follows:

1. Based on the semi-structured interview framework, we repeated the experiments previously conducted with LLM-based pseudo interviewees, this time involving both human interviewees and LLM-based pseudo interviewees, and compared the system’s persona attribute estimation performance in both types of interviews.
2. We introduced zero-shot and few-shot prompts to the LLM performing persona attribute estimation and demonstrated that few-shot prompting improves persona attribute estimation performance in dialogues with human interviewees.
3. We performed case analysis on instances where persona attribute estimation output improved with the introduction of few-shot prompting, and demonstrated cases where few-shot prompting functions effectively.

2 Related Work

In the field of dialogue systems, extensive research has been conducted on interview-oriented dialogue systems. Interview dialogue encompasses diverse elements such as question generation, dialogue management, and dialogue strategies, and has been the subject of wide-ranging research (Zeng et al., 2023; DeVault et al., 2014; Johnston et al., 2013; Kobori et al., 2016; B et al., 2020; Nagasawa et al., 2024; Ge et al., 2023; Inoue et al., 2020). Among these studies, research on dialogue systems for job interviews is notable (Su et al., 2019, 2018). Inoue et al. (2020) developed a system in which an android robot serves as the interviewer. This system analyzes keywords and satisfaction levels contained in user utterances and generates follow-up questions based on four predefined topics. Schatzmann and Young (2009) proposed a statistical modeling approach that simulates user responses for the purpose of optimizing dialogue strategies.

In recent years, rapid advances in LLMs have led to improvements in dialogue system performance (Geiecke and Jaravel, 2024). In particular, combining LLMs with slot-filling architectures has

been shown to achieve both advanced language understanding capabilities and flexible dialogue control (Hudeček and Dusek, 2023; Jacqmin et al., 2022; Siddique et al., 2021; Coope et al., 2020). Furthermore, by leveraging the high contextual understanding capabilities of LLMs such as GPT, question generation based on dialogue flow and previous utterance content, as well as flexible interview dialogue according to the situation, have become possible (Sun et al., 2024; Feng et al., 2023; Heck et al., 2023). Additionally, Wagner and Ultes (2024) demonstrated that combining LLMs with rule-based or slot-based dialogue management enables finer control of dialogue flow. Research on dynamic slot generation has also been conducted in fields other than interview systems. For example, Komada et al. (2024) proposed a method for maintaining narrative consistency by automatically generating slots from dialogue history in tabletop role-playing games (TRPGs). In the medical field, Hashimoto et al. (2025) developed a dialogue system to support career interviews by nursing managers and reported improvements in both interview efficiency and quality. Zenimoto et al. (2025) proposed a dialogue system that collects health questionnaire information from elderly people through casual conversation, implementing topic-guided dialogue that acquires target information by guiding LLMs using Chain-of-Thought prompting.

Additionally, Hasegawa et al. (2025) proposed a dialogue control framework using LLMs that can efficiently ask questions by accumulating information obtained during dialogue as slots in semi-structured interviews with multiple interviewees. Based on these prior studies, this research focuses on semi-structured interviews that dynamically generate and update slot sets based on interviewee utterance content. While research on semi-structured interviews has been conducted previously (Parfenova, 2024; Hu et al., 2024), most rely on static or manually designed control strategies, and the flexible generation and control of questions according to subject utterances has not been sufficiently examined. Furthermore, the utilization of information obtained from past interviews in new interviews has not been adequately studied. This study evaluates the proposed framework not only through simulations with LLM-based pseudo interviewees but also through semi-structured interviews with human participants. It further investigates the impact of introducing few-shot prompting on persona attribute estimation based on dialogue history and

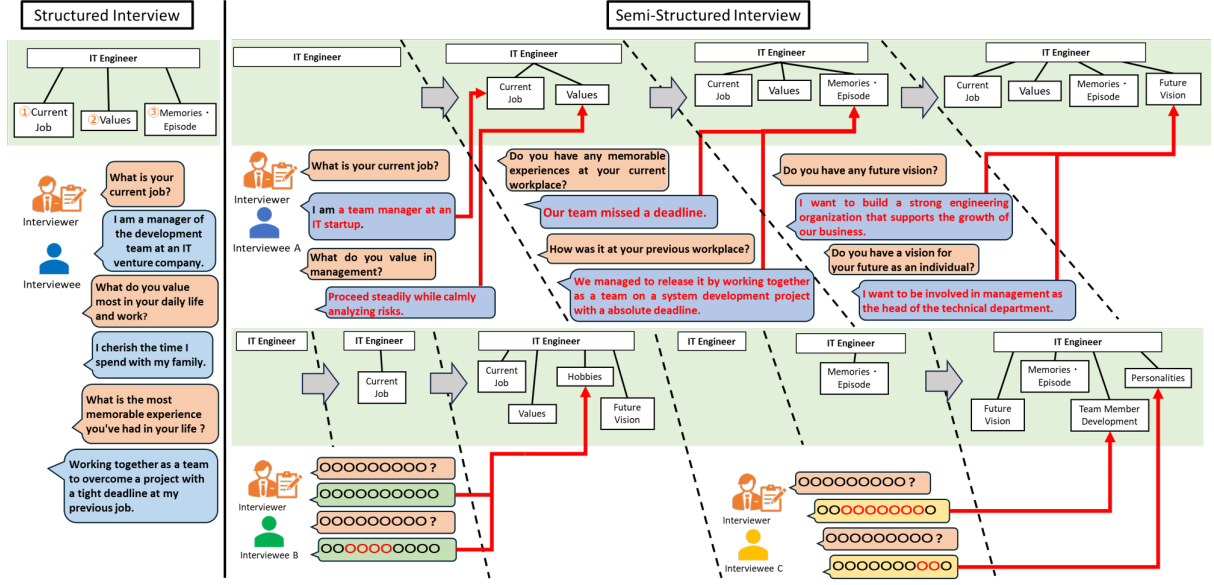


Figure 1: Diagram of Structured and Semi-Structured Interviews

slot information.

3 Semi-Structured Interviewer by LLMs

In this study, we implement a semi-structured interview system using LLMs based on the dialogue control framework (Hasegawa et al., 2025). Semi-structured interviews are an interview format in which the interviewer can flexibly ask additional questions according to the interviewee’s response content. Since questions can be adapted to the interviewee and situation, this approach has the advantage of eliciting more diverse responses compared to structured interviews based solely on pre-determined questions. The upper half of Figure 3 shows the overall processing flow of the interview system used in this study. The system consists of multiple LLMs that generate questions, create new slots, and record and update slot values from the interviewee’s utterances.

Figure 3 illustrates the processing flow of the proposed semi-structured interview system, including LLM modules for question generation and slot creation and update, and shows where each step takes place within the interactive system. In step 1, the system generates a new slot, “Current jobs”, and a new question is generated (step 2). Next, the interviewee answers the question (step 3), and the system fills the newly created slot with the answer (step 4). Finally, the system determines whether the saved response qualifies as a persona attribute. If it does, the response is saved with the corresponding slot values filled in (step 5). In other words, step

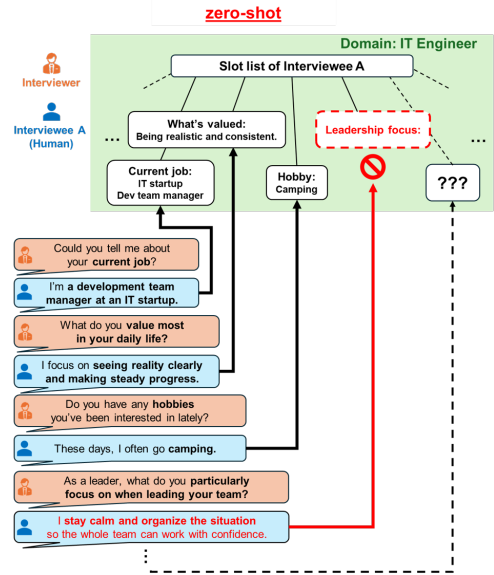


Figure 2: Diagram of Semi-Structured Interviews with Zero-Shot Prompts

3 is the interviewee’s answering of the question, whereas step 5 is the system’s inference, based on that answer, of whether the interviewee possesses the corresponding persona attributes. The system then determines whether to proceed to the next question; if it determines that the interview has been sufficiently conducted, the interview ends (step 6).

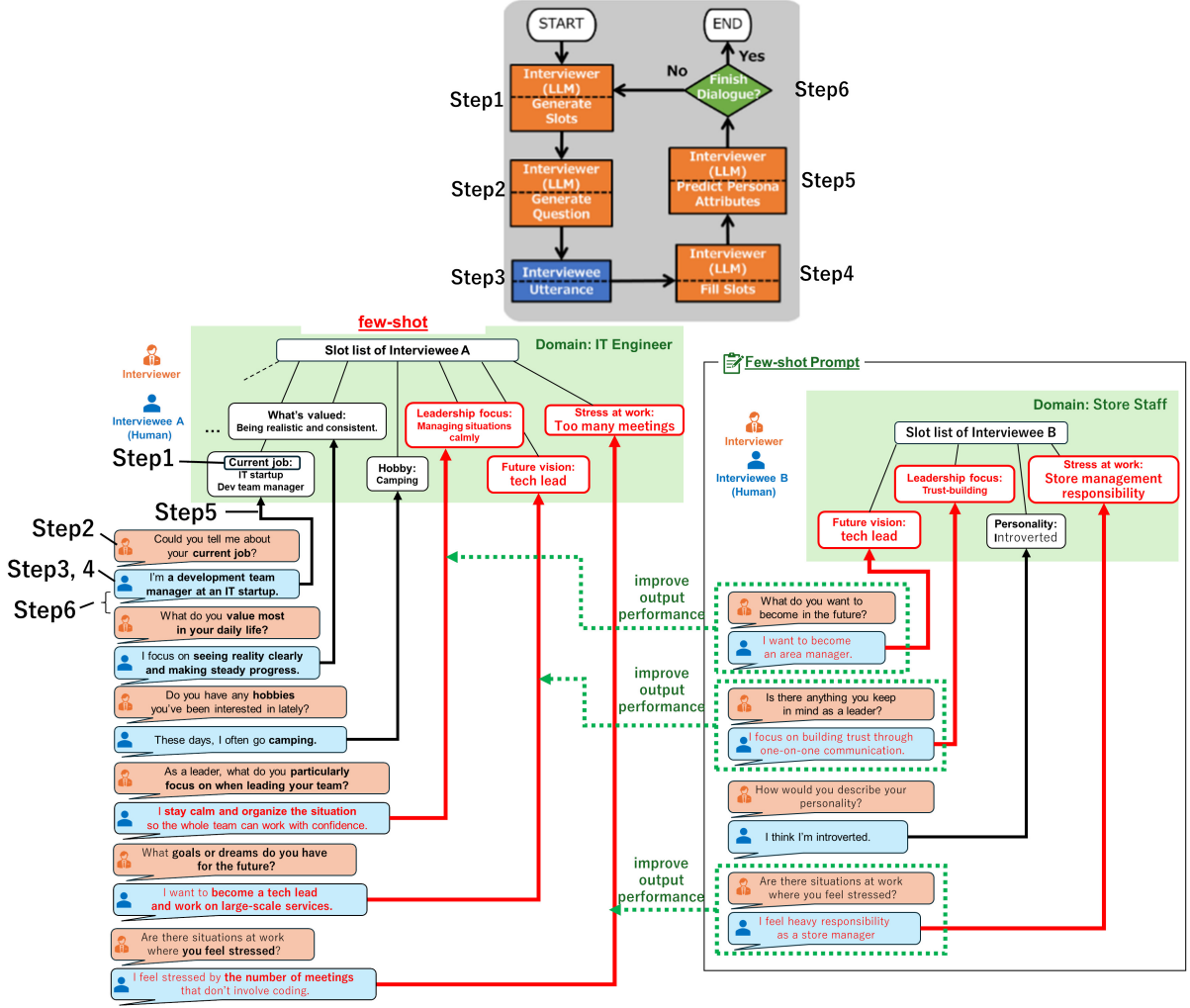


Figure 3: Diagram of Semi-Structured Interviews with Few-Shot Prompts

4 Persona Attributes and Values Estimation through Semi-Structured Interviews

A persona represents the character or role of an interviewee, referring to personal information and background information. Information collected through interviews is maintained as slots within the system. Each slot consists of a slot name and value pair. The slot name represents an abstract topic or attribute category (e.g., current career, hobbies, etc.), and the value stores the interviewee’s utterance content. At each turn of the interview, the system generates new slots and updates the values of existing slots based on the dialogue history.

New slot generation is performed through two types of methods. One is a method where the LLM generates slots for asking in-depth questions to the interviewee. The other is a method for generating new slots from a diverse candidate set of slot

names. As one component of this system, we introduce an LLM that estimates the interviewee’s persona attributes based on dialogue history and slots. This LLM estimates the interviewee’s persona attributes at each turn from the dialogue history and currently held slots. The estimation result, the estimated persona attributes, is labeled with whether the question regarding that attribute has not yet been asked or has already been asked. This enables prioritized generation of questions related to unasked estimated persona attributes, avoiding duplicate questions while improving information collection efficiency.

5 Human versus LLMs as Interviewees

There are differences in interview dialogues depending on whether the interviewee is human or an LLM (Figure 5 in Section A of Appendix).

Human interviewees typically provide diverse responses: vague or ambiguous expressions, an-

swers that do not fully align with the question, multiple pieces of information in a single utterance, topic deviation, or a shift in perspective rather than answering “I don’t know.” In contrast, the LLM-based pseudo interviewee used in this study, as well as that of Hasegawa et al. (2025), generates responses based on predefined persona settings, or replies “I don’t know” when the question is outside that scope. We give human participants the same instruction as the LLM interviewee: when a question is not covered by the assigned persona, they are told not to invent an answer. However, unlike the LLM, humans do not restrict themselves to the fixed response “I don’t know,” and this remaining diversity is a challenge addressed in this paper. As shown in Table 2 in Section A of Appendix, when the interviewee is an LLM-based pseudo interviewee, a specific response other than “I don’t know” about “leadership style” led the system to output that attribute. When the interviewee is human, the system did not treat the corresponding answer as important, and the attribute was not output.

6 Few-Shot Prompt for LLM-based Interviewer

In the proposed interview system, we apply few-shot prompting to the LLM responsible for estimating the interviewee’s persona attributes (Figure 3). Under the zero-shot condition, the persona attribute estimation LLM’s prompt is given no examples at all, and persona attributes are estimated based solely on the interviewee’s utterance content. In this case, since the LLM is not provided with explicit decision criteria through examples, it relies primarily on knowledge from its own pre-training and the interviewee’s response content to estimate persona attributes. As a result, estimation may become unstable, particularly for diverse and ambiguous utterances from human subjects.

In contrast, under the few-shot condition, a small number of examples for persona attribute estimation are provided as prompts to the persona attribute estimation LLM. This gives the estimation LLM explicit decision criteria for inferring persona attributes. As a result, the system can more appropriately estimate persona attributes from the subject’s utterance content, and improvements in estimation accuracy are expected. This comparison is important for clarifying the impact of few-shot prompting on persona attribute estimation performance for both LLM-based pseudo interviewees and human

interviewees. Specifically, we examine how few-shot prompting affects estimation for LLM-based pseudo interviewees, and how effectively it handles the diverse utterances of human interviewees.

7 Evaluation

7.1 The Procedure

In this study, we used the proposed interview system as the interviewer and set either an LLM-based pseudo interviewee or a human subject as the interviewee. For all LLMs in the system, we used GPT-4o¹ (gpt-4o-2024-11-20). Both LLM and human interviewees were provided with pre-created persona settings in advance to ensure reproducibility. The persona settings assigned individual values to each of 10 common persona attributes (Table 3 in Section A of Appendix). Personas were classified into three domains: “IT engineer,” “store staff,” and “school teacher.” We prepared 10 types of personas for each domain, and the LLM-based pseudo interviewee was designed to respond to questions related to content described in the persona settings based on the described content, while responding with “I don’t know” when asked about topics not described in the persona settings.

For dialogue control implementation, we used LangGraph².

First, using zero-shot prompting for the prompt of the LLM responsible for estimating persona attributes, we conducted interviews sequentially with both human and LLM interviewees. We conducted interviews with three LLM-based pseudo interviewees for comparison purposes and 13 human interviewees in each of the three domains. These 39 human participants were all different individuals; each was assigned a distinct persona profile and took part in only one domain, for a total of 9 LLM interviews and 39 human interviews.

Next, using few-shot prompting for the prompt of the LLM responsible for estimating persona attributes, we conducted interviews again sequentially with the same interviewees. In this case as well, we conducted 39 interviews similarly. We instructed participants to provide the same responses under the few-shot condition as they did under the zero-shot condition. Because this paper focuses on persona attribute estimation from the dialogue history, rather than on changes in the interviewees

¹<https://openai.com/index/hello-gpt-4o/>

²<https://github.com/langchain-ai/langgraph>

Table 1: Examples of Improvements in Persona Attribute Estimation Before and After Few-shot Introduction(Example 1: “School Teacher” domain)

	Utterance	Estimated Persona Attribute
System	What subjects were you particularly good at, when you were in school?	
Interviewee	I was particularly good at math when I was in school.	Favorite subjects in school
System	Could you tell me which subjects you were particularly good at in school?	
Interviewee	I was particularly good at computer science courses when I was in school.	Favorite subjects in school

(a) Excerpted Few-shot Prompt Examples Used in the Experiment(“IT engineer” domain)

Zero-shot		Few-shot	
Observed Dialogue			
What was your strongest subject in school?		What was your strongest subject in school?	
I was good at Japanese language and literature classes when I was in school.		I was good at Japanese language and literature classes when I was in school.	
Persona Attribute Estimation Results			
Human-annotated Persona Attribute (Ground Truth)	LLM Persona Attribute Estimation	Human-annotated Persona Attribute (Ground Truth)	LLM Persona Attribute Estimation
Future vision	Future vision	Future vision	Future vision
Family	Family	Values on Education	Values on Education
Promotion or job change	Hobbies	Promotion or job change	Promotion or job change
Hobbies	Career/Workplace	Hobbies	Hobbies
Past Career		Past Career	Past Career
Favorite subjects in school		Favorite subjects in school	Favorite subjects in school

(b) Model Estimations and Human-annotated Ground Truth (red colored persona attribute indicates the difference in persona attribute estimation between zero-shot and few-shot)

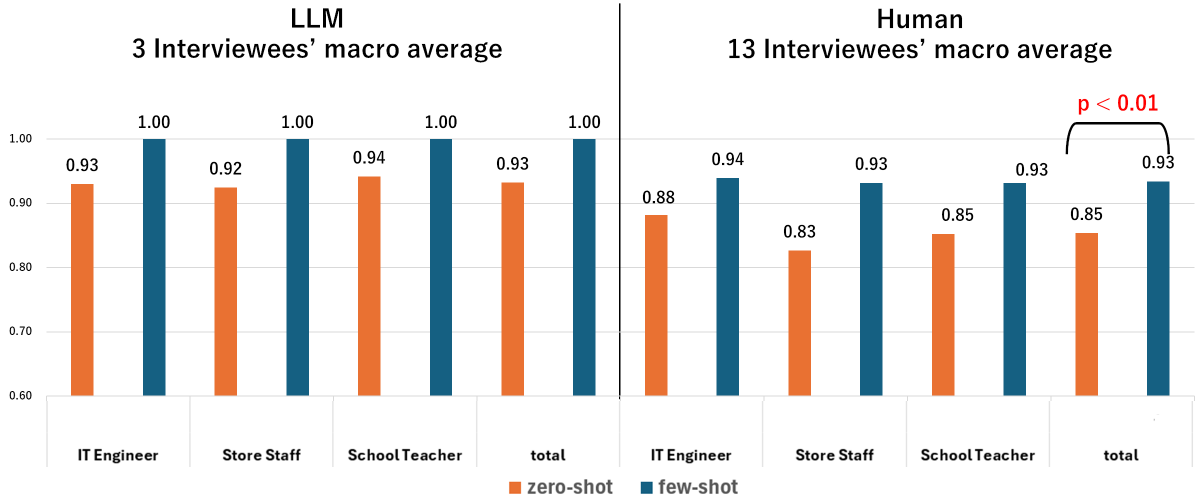


Figure 4: Evaluation Results of Estimating Persona Attributes (macro average of thirteen interviewees in each domain)

themselves, we consider the impact of interview order on the evaluation to be small.

To avoid the examples included in few-shot prompting being identical to the interviews being evaluated, the examples were created from domains different from the evaluation target domain. Specifically, for experiments in the “IT engineer” and “store staff” domains, we used few-shot examples from the “school teacher” domain, and for experi-

ments in the “school teacher” domain, we used few-shot examples from the “IT engineer” domain. This enabled evaluation of few-shot performance under settings where the evaluation target and few-shot domains were not identical. This design also indicates that the framework can be extended to a new domain simply by adding examples, which we consider far more scalable than traditional rule-based or domain-specific dialogue systems. The few-shot

examples were created based on preliminary experiments (3 people per domain) conducted before the main experiment. From dialogue logs obtained in the preliminary experiments, we distinguished items appropriate as persona attributes from items that should not be treated as attributes, and provided examples as positive and negative cases for persona attribute estimation. The negative examples are intended to illustrate typical responses that should not be treated as persona attributes, and they are designed to cover representative patterns of ambiguous, indirect, and non-attribute-based responses that frequently occur in actual interviews (e.g., generalizations, scene descriptions, impressions only, and abstract evaluations). Although the selection of these examples is not fully systematized, they reflect typical patterns of frequently occurring ambiguous responses.

To evaluate how accurately the proposed interview system could estimate interviewee persona attributes, we used recall, precision, and F-score as evaluation metrics. In this study, human evaluators reviewed the entire dialogue history and defined inferable persona attributes of the interviewee as the ground truth. One annotator was assigned to each domain to create the ground-truth data. Although this process involves subjective judgment, the interview questions are designed to elicit clear answers; therefore, we consider it straightforward to decide whether persona attributes can be inferred from the responses, and we regard the remaining ambiguity in ground-truth creation as extremely small. The objective of this evaluation is to measure how closely the LLM can approach such human annotation. The persona attributes output by the system after the interview conclusion were treated as estimation results. Recall was calculated with the number of persona attributes defined as ground truth as the denominator and the number of persona attributes matching the system’s estimation results as the numerator. A predicted persona attribute is not judged by exact string matching. Because human answers are diverse, we treat a prediction as correct when the interview history allows the judgment that the interviewee possesses the corresponding persona attribute. Therefore, recall represents a value indicating to what extent the system could identify persona attributes that were inferable through the interview. On the other hand, precision was calculated with the total number of persona attributes estimated by the system as the denominator and the number of persona attributes matching the ground

truth as the numerator. Therefore, precision is an indicator showing what proportion of the persona attributes estimated by the system were appropriate. F-score was calculated as the harmonic mean of Precision and Recall. Using these evaluation metrics, we evaluated which persona attributes the system could estimate and how accurately and comprehensively during the interview process.

Finally, based on a total of 39 interviews conducted by 39 interviewees (13 per domain), each with a distinct persona profile,³ across three domains, we constructed a distribution of statistics under the null hypothesis to examine whether there is a difference in the F-scores between zero-shot and few-shot conditions. Specifically, we grouped the 78 F-scores obtained under both conditions into a single set, randomly reassigned them into two groups, and calculated the mean difference between the two groups. By repeating this process 1,000 times, we estimated the distribution of the mean difference that would be obtained if the null hypothesis were true. We calculated the p-value by evaluating how extreme the observed mean difference between zero-shot and few-shot was within this estimated distribution. If the resulting p-value fell below the significance level of 0.05, we conclude that it was unlikely the observed difference arose by chance.

7.2 Evaluation Results

Figure 4 presents the evaluation results in terms of macro average of F-score for the “IT engineer,” “store staff,” and “school teacher” domains. Figure 4 gives the evaluation results for 13 human interviewees in each domain, while for comparison purposes, it also gives the evaluation results for three LLM-based pseudo interviewees in each domain.

First, from the comparison of human interviewees and LLM-based pseudo interviewees under the zero-shot condition, a tendency was observed for persona attribute estimation to become more difficult in dialogues with human subjects. For example, in the “IT engineer” domain, the macro average F-score under the zero-shot condition was 0.93 for the LLM pseudo interviewee but decreased to 0.88 for human interviewees. Similarly, in the “store staff” domain, the LLM pseudo interviewee achieved 0.92 while the human interviewee

³Details on the 39 interviewees are presented in section B of Appendix.

achieved 0.83. These results suggest that estimating persona attributes becomes more difficult in interviews with human subjects.

Next, for human interviewees, the few-shot condition surpassed the zero-shot condition in all domains. In fact, the p-value was under 0.01, indicating that there is a significant difference in F-scores between zero-shot and few-shot. The macro average F-score consistently improved across all domains for human interviewees, suggesting that introducing few-shot prompting contributes to improved persona attribute estimation performance for human utterances. On the other hand, for LLM-based pseudo interviewees, while improvement in macro average F-score was observed with the few-shot condition in the store staff domain, slight decreases in macro average F-score were confirmed in the “IT engineer” and “school teacher” domains.

Table 1 shows a case comparing persona attribute estimation results between zero-shot and few-shot conditions in the “school teacher” domain. In this case, under the few-shot condition, dialogue examples belonging to the “IT engineer” domain are presented as prompts in advance. In these few-shot examples, the correspondence relationship between the interviewer’s questions and persona attributes extracted from the corresponding utterance content is explicitly shown. In the conversation under the zero-shot condition, although the interviewee responds as “I was good at Japanese language and literature classes when I was in school.”, the LLM could not identify the persona attribute corresponding to “Favorite subjects in school” (in red colored character) from this utterance. On the other hand, under the few-shot condition, a similar utterance about favorite school subjects was obtained, and in this case, it can be seen that the LLM correctly estimated the corresponding persona attribute “Favorite subjects in school” (in red colored character).

This result suggests that the few-shot condition enabled appropriate attribute estimation by providing the LLM in advance with decision criteria for mapping utterances to corresponding persona attributes.

Table 4 and Table 5 in Section A of Appendix show that, in the “store staff” and “IT engineer” domains, few-shot prompts enable the LLM to more consistently associate the utterances with the appropriate persona attribute which is not reliably identified under the zero-shot condition. In Table 4 and Table 5, again, red colored persona attributes “Workplace relationships” in Table 4 and “Basic

Personal Information” in Table 5 respectively are among those correctly estimated only with the few-shot condition but not with the zero-shot condition.

8 Conclusion

In this study, we implemented an LLM interviewer based on semi-structured interviews and evaluated persona attribute estimation from dialogue history and slot information, for both LLM-based pseudo interviewees and human interviewees, with and without few-shot prompting. Estimation was more difficult with human interviewees. Few-shot prompting improved performance for humans in all domains, mainly in recall, but had mixed effects for LLM-based pseudo interviewees. Case analysis confirmed that examples can improve estimation.

As future work, as discussed in section D of Appendix, in order to make the semi-structured interview a more practical one, we will introduce more structured persona design. By this, it becomes easier for the interviewer LLM to efficiently organize the questions in the dialogue, where, out of those 10 initially set persona attributes, it is expected that the number of unexplored ones will decrease to almost zero.

Limitations

In this paper, interviews were conducted with 13 interviewees for each of the three domains (39 interviewees in total). Moving forward, it will be necessary to increase the number of interviewees and conduct experiments on a larger scale. A potential risk of this work is the impact on system stability resulting from the domain change. While we are currently conducting experiments using three domains, we should conduct further experiments with additional domains to account for a wider range of interviewees.

References

- Pooja Rao S B, Manish Agnihotri, and Dinesh Babu Jayagopi. 2020. Automatic follow-up question generation for asynchronous interviews. In *Proc. Intel-LanG*, pages 10–20.
- Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are

- few-shot learners. In *Proc. 33rd NeurIPS*, pages 1877–1901.
- Samuel Coope, Tyler Farghly, Daniela Gerz, Ivan Vulić, and Matthew Henderson. 2020. Span-ConveRT: Few-shot span extraction for dialog with pretrained conversational representations. In *Proc. 58th ACL*, pages 107–121.
- David DeVault, Ron Artstein, Grace Benn, Teresa Dey, Ed Fast, Alesia Gainer, Kallirroi Georgila, Jon Gratch, Arno Hartholt, Margaux Lhommet, Gale Lucas, Stacy Marsella, Fabrizio Morbini, Angela Nazarian, Stefan Scherer, Giota Stratou, Apar Suri, David Traum, Rachel Wood, and 3 others. 2014. Simsen-sei kiosk: a virtual human interviewer for health-care decision support. In *Proc. 13th AAMAS*, page 1061–1068.
- Yujie Feng, Zexin Lu, Bo Liu, Liming Zhan, and Xiaoming Wu. 2023. [Towards LLM-driven dialogue state tracking](#). In *Proc. EMNLP*, pages 739–755.
- NG Fielding, editor. 2003a. *Interviewing*, volume I. Sage Publications.
- NG Fielding, editor. 2003b. *Interviewing*, volume II. Sage Publications.
- Yubin Ge, Ziang Xiao, Jana Diesner, Heng Ji, Karrie Karahalios, and Hari Sundaram. 2023. What should I ask: A knowledge-driven approach for follow-up questions generation in conversational surveys. In *Proc. 37th PACLIC*, pages 113–124.
- Friedrich Geiecke and Xavier Jaravel. 2024. [Conversations at scale: Robust ai-led interviews with a simple open-source platform](#). *Social Science Research Network*, pages 1–73.
- Ryo Hasegawa, Yijie Hua, Takehito Utsuro, Ekai Hashimoto, Mikio Nakano, and Shun Shiramatsu. 2025. [A dialogue system for semi-structured interviews by LLMs and its evaluation on persona information collection](#). In *Proc. IWSDS 2025*, pages 39–59.
- Ekai Hashimoto, Mikio Nakano, Takayoshi Sakurai, Shun Shiramatsu, Toshitake Komazaki, and Shiho Tsuchiya. 2025. A career interview dialogue system using large language model-based dynamic slot generation. In *Proc. 31st COLING*, pages 1562–1584.
- Michael Heck, Nurul Lubis, Benjamin Ruppik, Renato Vukovic, Shutong Feng, Christian Geishauser, Hsien-chin Lin, Carel van Niekerk, and Milica Gasic. 2023. [ChatGPT for zero-shot dialogue state tracking: A solution or an opportunity?](#) In *Proc. 61st ACL*, pages 936–950.
- Jiaxiong Hu, Jingya Guo, Ningjing Tang, Xiaojuan Ma, Yuan Yao, Changyuan Yang, and Yingqing Xu. 2024. [Designing the conversational agent: Asking follow-up questions for information elicitation](#). *Proc. ACM Hum.-Comput. Interact.*, 8(CSCW1).
- Vojtěch Hudeček and Ondrej Dusek. 2023. [Are large language models all you need for task-oriented dialogue?](#) In *Proc. 24th SIGDIAL*, pages 216–228.
- Koji Inoue, Kohei Hara, Divesh Lala, Kenta Yamamoto, Shizuka Nakamura, Katsuya Takanashi, and Tatsuya Kawahara. 2020. Job interviewer android with elaborate follow-up question generation. In *Proc. ICMI*, page 324–332.
- Léo Jacqmin, Lina M. Rojas Barahona, and Benoit Favre. 2022. [“do you follow me?”: A survey of recent approaches in dialogue state tracking](#). In *Proc. 23rd SIGDIAL*, pages 336–350.
- Michael Johnston, Patrick Ehlen, Frederick G. Conrad, Michael F. Schober, Christopher Antoun, Stefanie Fail, Andrew Hupp, Lucas Vickers, Huiying Yan, and Chan Zhang. 2013. [Spoken dialog systems for automated survey interviewing](#). In *Proc. 14th SIGDIAL*, pages 329–333.
- Takahiro Kobori, Mikio Nakano, and Tomoaki Nakamura. 2016. [Small talk improves user impressions of interview dialogue systems](#). In *Proc. 17th SIGDIAL*, pages 370–380.
- Keigo Komada, Kaori Abe, Shoji Moriya, and Jun Suzuki. 2024. Dynamic slot-making-and-filling method for improving long-term dialogue consistency. In *Proc. JSAI*, page JSAI2024(0):4Xin2102–4Xin2102 (in Japanese).
- Fuminori Nagasawa, Shogo Okada, Takuya Ishihara, and Katsumi Nitta. 2024. [Adaptive interview strategy based on interviewees’ speaking willingness recognition for interview robots](#). *IEEE Transactions on Affective Computing*, 15(3):942–957.
- Angelina Parfenova. 2024. [Automating the information extraction from semi-structured interview transcripts](#). In *Proc. WWW*, page 983–986.
- Jost Schatzmann and Steve Young. 2009. [The hidden agenda user simulation model](#). *IEEE Transactions on Audio, Speech, and Language Processing*, 17(4):733–747.
- A.B. Siddique, Fuad Jamour, and Vagelis Hristidis. 2021. [Linguistically-enriched and context-aware zero-shot slot filling](#). In *Proc. WWW*, page 3279–3290.
- Ming-Hsiang Su, Chung-Hsien Wu, and Yi Chang. 2019. [Follow-up question generation using neural tensor network-based domain ontology population in an interview coaching system](#). In *Interspeech*, pages 4185–4189.
- Ming-Hsiang Su, Chung-Hsien Wu, Kun-Yi Huang, Qian-Bei Hong, and Huai-Hung Huang. 2018. [Follow-up question generation using pattern-based seq2seq with a small corpus for interview coaching](#). In *Interspeech*, pages 1006–1010.

Guangzhi Sun, Shutong Feng, Dongcheng Jiang, Chao Zhang, Milica Gasic, and Phil Woodland. 2024. [Speech-based slot filling using large language models](#). In *Findings of ACL*, pages 6351–6362.

Nicolas Wagner and Stefan Ultes. 2024. [On the controllability of large language models for dialogue interaction](#). In *Proc. 25th SIGDIAL*, pages 216–221.

Tom Wengraf. 2001. *Qualitative Research Interviewing*. Sage Publications.

Jie Zeng, Yukiko Nakano, and Tatsuya Sakato. 2023. [Question generation to elicit users’ food preferences by considering the semantic content](#). In *Proc. 24th SIGDIAL*, pages 190–196.

Yuki Zenimoto, Mariko Yoshida, Ryo Hori, Mayu Urata, Aiko Inoue, Takahiro Hayashi, and Ryuichiro Higashinaka. 2025. [Automated administration of questionnaires during casual conversation using question-guiding dialogue system](#). In *Proc. SEMDIAL*, pages 115–124.

A Additional Figures and Tables

Figure 5 contrasts typical response characteristics of human interviewees and an LLM-based pseudo interviewee in the semi-structured interview setting. Table 2 provides example dialogue excerpts comparing responses from a human interviewee and an LLM-based pseudo interviewee. As a complement for Figure 4 presenting F-score only but not recall nor precision, Figure 6 presents the evaluation results in terms of macro average of recall / precision / F-score over 13 human interviewees in each of “IT engineer,” “store staff,” and “school teacher” domains. As shown in the figure, in all the domains, few-shot achieved improvement over zero-shot in F-scores, where their improvements in recall are more dominant than those in precision in each of the three domains. Table 3 summarizes an example predefined persona used in the experiments, listing persona attributes and their values for an IT engineer (“Shota Suzuki”). Table 4 and Table 5 present additional case studies from the store staff and IT engineer domains, showing how few-shot examples help map utterances to the persona attributes “Workplace relationships” and “Basic Personal Information,” respectively.

B Details on Participants

39 human participants took part in the experiments described in this paper (13 participants in each of the three domains). Each participant was assigned a distinct persona profile and did not participate

in more than one domain. All participants were students and received no financial compensation. Prior to the experiments, we explained the purpose and importance of this study, and all participants agreed to take part in the evaluation on a voluntary basis. The following instructions were given to the participants before the interviews. Participants were asked (i) to respond to the system’s questions as interviewees, (ii) to answer by role-playing the pre-defined persona assigned to them, and (iii) not to disclose any personal information beyond what was specified in the persona description.

C Few-shots versus CoT Prompts

What we have given to LLMs as few-shots are far beyond conventional few-shots but they are nearly equivalent to CoT studied in Zenimoto et al. (2025) both in their quantity and quality. Our few-shots consist of cross domain dialogue history of three interview sessions as well as all the pairs of slot names and slot values automatically extracted by a zero-shot LLM from the given dialogue history. Here, each interview session corresponds to an interview with an interviewee. All the pairs of slot names and slot values extracted from the given dialogue history, on the other hand, are divided into positive ones corresponding to persona attributes of each interviewee and the remaining negative ones not corresponding to persona attributes. (For example, few-shots collected from the school teachers domain consist of 31 persona attributes and 43 of those other than persona attributes, while those collected from the store staff domain consist of 31 persona attributes and 44 of those other than persona attributes.) Those plenty of few-shots are valuable information source that represent criteria when judging whether a pair of a slot name and a slot value is appropriate as a persona attribute, which can be regarded as a CoT.

D Persona Attributes Employed in Experiments

The 10 persona attributes are manually set in advance before we start interview experiments. The subset of those 10 persona attributes manually set in advance and appearing in the actual dialogue are only a part of the whole ground truth persona attributes. The whole ground truth persona attributes include the subset of those 10 and also the additional ones that are manually selected through careful examination of the actual dialogue history.

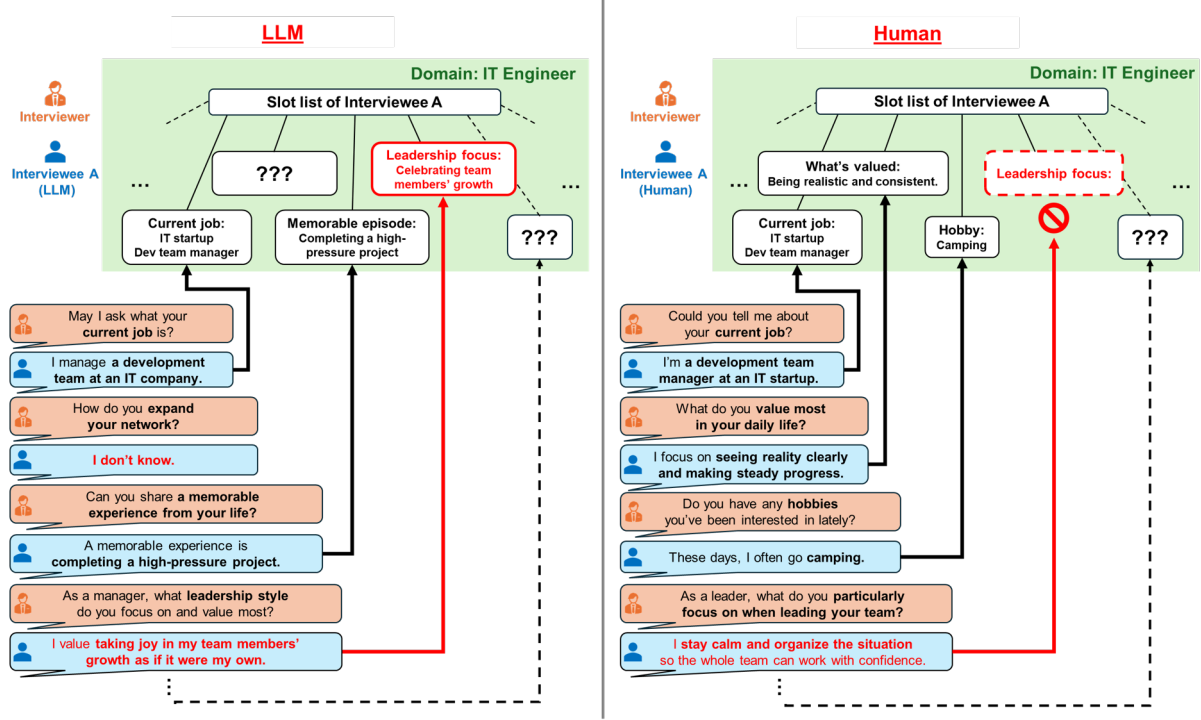


Figure 5: Diagram of Semi-Structured Interviews with Human and LLM Interviewees by Zero-shot Prompts

Speaker	Utterance	Estimated Persona Attribute
System:	As a manager of the development team, what leadership style do you value most?	
Interviewee:	I value a style that is attentive to others and takes genuine delight in the growth of team members as if it were my own.	[[Correct]]: Leadership Style

(c) Dialogue between the System and the LLM Interviewee

Speaker	Utterance	Estimated Persona Attribute
System:	What do you particularly focus on or keep in mind when leading your team as a leader?	
Interviewee:	I will maintain a level head and organize the situation so the entire team can move forward with confidence.	[[Incorrect]]: "None"

(d) Dialogue between the System and the Human Interviewee

Table 2: The differences of responses between Human-interviewee and LLM-interviewee

(shown in Table 1, Table 4, and Table 5).

As clearly shown in those tables, regarding the comparison between zero-shot and few-shot, we are successful in achieving higher recall and hence higher F-score in the task of persona estimation through semi-structured interview. However, in order to make the semi-structured interview a more practical one, we will introduce more structured persona design. By this, it becomes easier for the interviewer LLM to efficiently organize the questions in the dialogue, where, out of those 10 initially set persona attributes, it is expected that the number of unexplored ones will decrease to almost zero.

E Topic Selection Strategy in Dialogues

In the dialogue session of the proposed method, the following topic selection strategy is employed.

- (1) First, it is randomly selected whether the follow-up question strategy is to be selected or not.
- (2a) when the follow-up question strategy is selected, a new slot that follows the current thread of the ongoing dialogue is generated by the slot generator LLM, and then a new question for filling the newly generated slot is composed by the question composer LLM.
- (2b) even when the follow-up question strategy is

ID	Persona Attributes (given in advance)	Values of “Shota Suzuki” (given in advance)
1	Basic Personal Information (Name, Age, Home-town, Gender)	Shota Suzuki, 25 years old, Tokyo, Male
2	Personality	Strong in logical thinking and highly intellectually curious. Once focused, tends to become deeply absorbed in tasks and persistently works through problems until they are solved.
3	Past Career	After graduating from a university with a major in information science, joined a current major tech venture as a new graduate and is now in the third year. In the university laboratory, engaged in research on image recognition technology and presented findings at academic conferences.
4	Current Career	Working as a backend engineer at a company that operates one of Japan’s largest social networking services. Mainly responsible for new API development and maintenance/operation of microservices.
5	Vision for the current career	Wants to be involved in designing the overall architecture of large-scale services. Aims to become a tech lead who can participate from the stage of technology selection.
6	Concerns and Dissatisfaction	Because the organization is large and vertically divided, it’s difficult to see how one’s own work contributes to the overall product. Feels stressed by the large number of coordination tasks such as meetings and documentation, beyond writing code.
7	Thoughts on Promotion or Career Change	Has a professional orientation toward pursuing technical expertise rather than management. In the future, is also considering working at an early-stage startup with greater autonomy.
8	Hobbies and Personal Life	Hobbies include intellectually challenging board games and weekend bouldering. Often spends days off reading trending technical books or discussing interesting technologies with friends.
9	Family	Currently living alone near the company, having moved out from the family home in Tokyo. Family includes parents and a younger sister who is a university student.
10	Memorable Episodes	In his first year at the company, a feature he developed was released, and he saw users on social media saying things like, “This feature is really useful.” It made him realize that the code he wrote had a real impact on people’s lives, and it gave him a strong sense of purpose and fulfillment.

Table 3: List of persona attributes for interviewees and values for the first interviewee, IT engineer “Shota Suzuki”

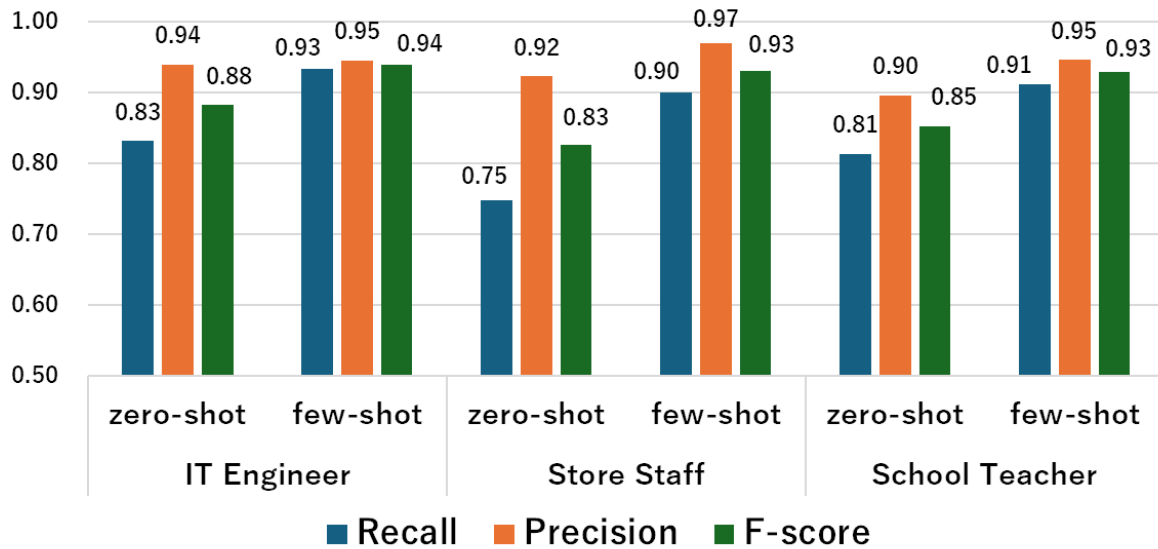


Figure 6: Evaluation Results of Estimating Persona Attributes (macro average of Recall / Precision / F-scores over interviewees)

NOT selected, among the pool of topics for shifting from current topic to another one, considering the history of topics already covered in this dialogue so far, one remaining topic is randomly selected. A new slot corresponding to the selected topic is then generated by the slot generator LLM, and a question for filling that slot is composed by the question composer LLM.

Table 4: Examples of Improvements in Persona Attribute Estimation Before and After Few-shot Introduction(Example 2: “Store Staff” domain)

	Utterance	Estimated Persona Attribute
System	Could you share any particularly memorable experiences or thoughts you have regarding interpersonal relationships in the educational setting?	
Interviewee	One of my former students, now an adult, went out of their way to come and say hello, telling me, “You were scary, but I am who I am today because you taught me so strictly back then.	Workplace Relationships
System	How do you feel about your relationships with colleagues and supervisors in the educational setting?	
Interviewee	I don’t really have any particular thoughts about my relationship with my boss.	None

(a) Excerpted Few-shot Prompt Examples Used in the Experiment(“school teacher” domain)

Zero-shot		Few-shot	
Observed Dialogue			
How do you feel about relationships in the workplace? For example, please tell us about any memorable experiences you’ve had with colleagues or supervisors.		How do you feel about relationships in the workplace? For example, I’d love to hear about your relationships with colleagues and supervisors.	
We have many regular customers, and I enjoy working in this friendly, relaxed atmosphere.		We have many regular customers, and I enjoy working in this friendly, relaxed atmosphere.	
Persona Attribute Estimation Results			
Human-annotated Persona Attribute (Ground Truth)	LLM Persona Attribute Estimation	Human-annotated Persona Attribute (Ground Truth)	LLM Persona Attribute Estimation
Current career Workplace relationships Personality Past Career Basic Personal Information	Career/Workplace Personality Basic Personal Information	Current career Workplace relationships Personality Past Career Basic Personal Information	Current career Workplace relationships Personality Past Career Basic Personal Information

(b) Model Estimations and Human-annotated Ground Truth(**red colored persona attribute** indicates the difference in persona attribute estimation between zero-shot and few-shot)

Table 5: Examples of Improvements in Persona Attribute Estimation Before and After Few-shot Introduction(Example 3: “IT engineer” domain)

	Utterance	Estimated Persona Attribute
System	Could you tell me what kind of neighborhood you live in?	
Interviewee	I live in Kanagawa Prefecture	Basic Personal Information
System	Could you tell me about where you’re from and the environment you grew up in?	
Interviewee	I’m from Chiba Prefecture	Basic Personal Information

(a) Excerpted Few-shot Prompt Examples Used in the Experiment(“school teacher” domain)

Zero-shot		Few-shot	
Observed Dialogue			
First, I'd like to ask you a basic question: Where do you live?		Could you start by briefly introducing yourself?	
I'm from Saitama Prefecture		My name is Takuya Nakamura, I'm a 38-year-old man. I'm from Saitama Prefecture.	
Persona Attribute Estimation Results			
Human-annotated Persona Attribute (Ground Truth)	LLM Persona Attribute Estimation	Human-annotated Persona Attribute (Ground Truth)	LLM Persona Attribute Estimation
Personality	Personality	Personality	Personality
Side job	Career/Workplace	Decision-making process	Decision-making process
Basic Personal Information	Hobbies	Basic Personal Information	Basic Personal Information
Promotion or job change	Family	Promotion or job change	Promotion or job change
Hobbies		Hobbies	Hobbies
Family		Family	Family

(b) Model Estimations and Human-annotated Ground Truth (**red colored persona attribute** indicates the difference in persona attribute estimation between zero-shot and few-shot)

The LM Challenge 2026

Sherzod Hakimov[⊗], Dimitrije Ristic[⊗], Philipp Sadler[⊗], Davide Mazzaccara[‡],
Sabrina McCallum[◊], Filippo Momentè[‡], Maria Manina[⊗], Karl Osswald[⊗], Dieu-Thu Le[◊],
Oliver Lemon[◊], Alexander Koller[§], Raffaella Bernardi^{*}, Raquel Fernández[△],
Mario Giulianelli[×], Alessandro Suglia[◊], David Schlangen^{⊗, ⊕}

[⊗]University of Potsdam, [‡]University of Trento, [◊]University of Edinburgh,

[◊]Amazon AGI, Germany, [◊]Heriot-Watt University, [§]Saarland University,

^{*}Free University of Bozen-Bolzano, [△]University of Amsterdam, [×]UCL, [⊕]DFKI

lm-playschool@googlegroups.com

Abstract

Children acquire language from far less input than language models, and social interaction is widely thought to be part of the reason. Dialogue games cannot fully replicate human interaction, but they offer a controllable environment for simulating situated language use and for studying how models learn from and through interaction. The LM Playschool Challenge builds on this idea, shifting language model training away from observational data towards multi-turn language use. Participants were free to post-train a language model of any size, using any data or method, with one objective in mind — to improve its performance on an unseen set of dialogue games. Our findings suggest that post-training on dialogue games effectively teaches game rules and structure and can support generalisation, especially for small models — the best system achieved gains of roughly seven and three times its baseline score on held-out instances of seen games and on unseen games, respectively. However, gains on unseen games were concentrated in a few games, and larger models improved mainly by achieving better completion rates. General ability was largely retained across submitted systems, owing to the fact that fine-tuning on dialogue games produces shorter responses — and in turn, trigger fewer formatting errors on static benchmarks. Nearly all submissions relied on supervised fine-tuning, often followed by preference optimisation, and none trained on the dense turn-level feedback offered by Playpen, leaving this an open direction for future iterations of the challenge.

1 Introduction

The **LM Playschool Workshop** is concerned with studying and developing language agents that learn or improve through social interaction in conversational, collaborative, goal-oriented, multi-turn environments. The **LM Playschool Challenge** gave

this broad research programme a concrete empirical realisation by asking whether post-training large language models through *dialogue games*—i.e., goal-oriented tasks governed by multi-turn language interaction—can improve their interactive capabilities. This question is relevant to the development of more useful language technologies because real-world interactions *are* conversational, collaborative, goal-oriented, and multi-turn. It is also relevant to the development of models that learn and use language in more human-like ways, both with respect to the formal and functional skills (Mahowald et al., 2024) they acquire and to the efficiency of that learning.

Dialogue games provide a useful setting for evaluating language agents and their ability to use language in interaction. In collaborative dialogue games, a language agent works with a partner towards a joint goal (Schlangen, 2023). Success therefore depends on its ability to understand the task, coordinate with its partner through language, select appropriate actions, and adapt to the changing game state across multiple turns. Previous work has shown that LLMs can be prompted to play a wide range of interactive language-mediated games (Chalamalasetti et al., 2023; Li et al., 2023; Qiao et al., 2023; Xu et al., 2023). These games test verbal interaction skills that are not adequately covered by single-turn benchmarks. They can discriminate between models, correlate with cognitive tests targeting core executive functions (Momentè et al., 2025), and reveal that current LLMs still fall short of human performance (Beyer et al., 2024).

Dialogue games are also promising environments for training language agents because they provide an intrinsic learning signal derived directly from success or failure in the game (Wang et al., 2024a; Cheng et al., 2024; Gul and Artzi, 2024; Horst et al., 2025). First, this signal evaluates utter-

ances by how effectively they advance a shared goal rather than by their local appropriateness alone. It is also an objective signal, in the sense that it can be computed programmatically from the game state without relying on a learned judgment model, and it is grounded in interaction because task success depends on successful communication across turns. These properties support several approaches to post-training, including offline learning from recorded interactions and online learning through interaction with the game environment.

The study of how language agents learn through interaction is also informed by work on human language acquisition. Human children learn from far less linguistic input than current neural language models require (Hart and Risley, 1995; Cristia et al., 2019; Linzen, 2020), and social interaction is widely hypothesised to contribute to this data efficiency (Clark, 2016; Bloom, 2000; Bisk et al., 2020; Bender and Koller, 2020). Dialogue games do not reproduce the conditions of human language acquisition, but they provide a controlled setting in which to investigate whether learning through interaction can make model training more effective and data-efficient.

The LM Playschool Challenge brought together these three lines of work: language-based games as a means of evaluating large language models and their ability to use language in interaction; game-based and interactive environments for training large language models; and connections between human language acquisition and machine language learning, with particular attention to the role of social interaction in data-efficient learning. The challenge asked participants to improve language agents through model post-training. Participants were free to choose their base models, data, and training methods, but were particularly encouraged to use interaction data. The public Playpen environment (Horst et al., 2025) provided example games, learning scripts, and a development evaluation setup. Submitted systems were then evaluated on unseen dialogue games drawn from a private version of the clembench framework (Chalamalasetti et al., 2023) as well as on static evaluation benchmarks, with improvement measured relative to each system’s base model. The challenge received eleven systems from eight teams, all of which were built on Qwen3.5. Most submissions relied on offline supervised fine-tuning on Playpen interactions, often followed by DPO (Rafailov et al., 2023), while two submissions ex-

plored inference-time methods and only one used online reinforcement learning.

Across the submitted systems, post-training on dialogue games teaches models the rules and structure of the games more effectively than optimal strategies. The most significant gains were observed in the smallest (2B) models, where the best system achieved a 7-fold improvement on unseen instances of training games and a 3-fold on entirely unseen games relative to the baseline model, through gains in both game completion and game quality. These results demonstrate that post-training can support generalisation, especially for smaller models, although gains on unseen games were smaller than those in-domain and were concentrated on a small subset of games. For larger models, improvements were mainly in-domain and driven by higher completion rates on a wide range of games. On static benchmarks, despite a slight overall decline, post-training produced more concise outputs and fewer formatting errors.

These findings are based on the approaches explored by the submitted systems and therefore leave several promising directions open. In particular, no submission trained directly on the dense feedback that the Playpen provides within an episode, leaving learning from feedback during interaction as an interesting direction for future iterations of the challenge.

2 The Playpen Environment

Playpen is an interactive environment for learning from and in dialogue games. It supports the use of dialogue games both as evaluation devices and as environments for model training by structuring interaction around the standard observe-act loop and providing access to turn-level and episode-level rewards. It can be considered as a learning-oriented wrapper around the clembench package, which in itself focussed on evaluation and not learning. For the challenge, we re-wrote the environment used in (Horst et al., 2025) with a focus on ease-of-use and extensibility.¹

A dialogue game in clembench (and hence, in Playpen) involves one or more player agents whose interactions with the environment and with one another are mediated by a programmatic GameMaster. At each turn, an agent receives an observation containing the dialogue history together

¹The package is being developed further; the current state can be found at <https://github.com/lm-playpen/playpen>.

with any game-specific state that is available to its role, and produces an action. Depending on the game, the action may take the form of a move command, a clue, a guess, or a message to another player. The GameMaster processes the action, checks that it satisfies the rules of the game, updates the game state, and returns the next observation together with any associated reward. This process continues until the game reaches a terminal state.

Beyond standard game-specific reward signals, Playpen can also incorporate additional sources of feedback, such as responses from judgment models or verifiers, and provide agents with access to tools. Integration with the PettingZoo (Terry et al., 2021) and Gymnasium (Towers et al., 2025) provides interfaces for multi-agent and single-agent learning, respectively, while integration with OpenEnv sandbox (OpenEnv Technical Committee, 2025) supports isolated execution of game environments. Playpen is designed around the clembench suite of games (Chalamalasetti et al., 2023; Schlangen et al., 2026). Users can customise existing games, define task-specific scores and reward functions, generate new game instances, or design new games altogether.

Specifically for the challenge, we provided a simple command-line interface through which a candidate model could be validated on a specific set of game-instances held back from the training set, as well on some static benchmarks; as described in more detail in the next section. We also packaged up transcripts from previous runs of the benchmark (as documented at <https://clembench.github.io>) into an easy to process format, for experiments with learning from demonstrations.

3 Evaluation

The LM Playschool Challenge asked participants to post-train a baseline model and submit the resulting model checkpoint. (Note: What we call *baseline model* here are publicly released, instruction-tuned and post-trained models that form the basis for our *continued* post-training; not to be confused with base models that are only pre-trained.) We evaluate these checkpoints following Horst et al. (2025), comparing each post-trained model with the corresponding base model across three main dimensions. **In-domain performance** measures improvement on unseen game instances of the clembench games that were available during training. **Out-of-domain performance** is evaluated on previously

unpublished clembench games, providing a measure of how well improvements transfer to tasks that were not available during post-training. Finally, we evaluate whether models **retain their general capabilities** after post-training using a set of static benchmarks (e.g., MMLU-Pro, Wang et al., 2024b).

In addition to this main evaluation, we conduct separate validation runs on the game instances included in the official Playpen validation set and on the static benchmark tasks released to participants as part of the challenge package. All evaluation runs were carried out after the submission deadline using a fixed evaluation pipeline and the same configuration across submissions.²

3.1 In-Domain Evaluation

In-domain evaluation measures whether post-training improved a model’s gameplay on the distribution of games it was trained on, but using unseen game instances. The in-domain set consists of 16 games and game variants featured in the Playpen training set and available as part of clembench. The instances were produced following the standard recipe for the benchmark. The instances, together with the material described in the next section, are now available at <https://github.com/lm-playpen/lm-playschool-2026-closed-set>.

3.2 Static Benchmark Evaluation

Static benchmark evaluation assesses whether post-training improves or deteriorates general capabilities. We evaluate each model on 100 instances from each of five static benchmarks: BigBenchHard (BBH; Suzgun et al., 2023; Srivastava et al., 2023), CLadder (Jin et al., 2024), EQ-Bench (Paech, 2023), IFEval (Zhou et al., 2023), and MMLU-Pro (Wang et al., 2024b).

3.3 Out-of-Domain Evaluation

Out-of-domain evaluation measures whether improvements acquired through post-training transfer to games that are beyond the training distribution, or whether the model has overspecialised to the games available during training. For this purpose, we introduce a set of 13 new, previously unpublished games and game variants; hence, participants didn’t have access to them when they

²The full runs, including all transcripts of gameplay, are available at <https://github.com/lm-playpen/lm-playschool-2026-final-results>.

submitted their final checkpoints. Five are new games developed for the challenge (CHRONICLE, CLOCKWORK COURIER, CRYPTOLECT, TOWER OF HANOI, and GET TO THE POINT), two are new variants of games already available during training (CLEAN UP — SEMANTIC TEXT and WORDLE CRAZY), and four are adaptations of games from TextArena (Guertler et al., 2025) (SOKOBAN, BLACKJACK, FROZEN LAKE, and MASTERMIND).

CHRONICLE is a two-player historical deduction game. The game features a Narrator who describes a secret historical event (such as a battle, scientific discovery, or a political milestone) to a Detective over 10 turns, one sentence at a time, without naming the event or using forbidden event-specific nouns. The game evaluates a model’s world knowledge as well as the ability to describe and recognise an event from partial descriptions.

CLOCKWORK COURIER is a single-player planning and navigation game played on a dynamic map that changes according to a pre-defined timetable. A courier must collect every parcel on the map and deliver it to its designated destination before time runs out. Map navigation is further complicated by gates that open and close on a fixed timer and guards that patrol predefined routes. The full map and timetable are available to the player from the beginning, making the task one of planning under known, deterministic dynamics. The game assesses the ability to plan and timely execute actions over a dynamically changing environment.

CRYPTOLECT is a single-player grammar induction game in which the model assumes the role of a field linguist learning an unknown constructed language through interaction with a programmatic informant. The final task is to translate a set of previously unseen sentences into the new language. The game evaluates the ability to acquire information through interaction, infer latent rules from evidence, and generalise these rules to new examples.

GET TO THE POINT is a two-player word-association game where one player, the Helper, attempts to guide the other, the Seeker, from a fixed starting word towards a hidden target word through indirect clues. The game resembles Taboo, which is included in the in-domain set, but differs in its communication constraints. Rather than avoiding a list of forbidden words, the Helper may use the starting clue word and up to three additional words on each turn. The game therefore evaluates semantic reasoning as well as, given its similarity to

Taboo, a model’s ability to transfer skills acquired on a related training game to a task with different interaction rules.

TOWER OF HANOI is a text-based implementation of the classic puzzle, featuring three pegs and disks of varying sizes. The model must move all disks to a target peg, moving only one disk per turn and never placing a larger disk on top of a smaller one. The game evaluates state tracking and multi-step planning.

We additionally include two previously unseen variants of in-domain games. In CLEAN UP – SEMANTIC TEXT, two players are given a 7×7 ASCII grid containing letter objects at different positions. The goal is to rearrange the objects on their private grids so that identical objects occupy matching coordinates. Unlike in the original game, explicit coordinates and row or column instructions are forbidden, and players are required to describe locations relative to objects in the surrounding environment. In WORDLE CRAZY, the core game mechanics remain the same as in WORDLE, but the mapping between guesses and colour-coded feedback is altered. Models must therefore infer and adapt to the modified feedback convention rather than relying on the standard WORDLE rules.

Finally, SOKOBAN, BLACKJACK, FROZEN LAKE, and MASTERMIND are well-known games integrated from TextArena. They test several forms of general-purpose planning and reasoning: spatial planning, decision-making under uncertainty, navigation under stochastic dynamics, and logical deduction from constrained feedback, respectively.

3.4 Evaluation Pipeline

For evaluation, each author submitted a merged model checkpoint via HuggingFace, along with information about the baseline model it was fine-tuned from. Authors also had to submit an entry for the clemcore model registry, specifying the configuration to use during evaluation.³

In total, six teams submitted fine-tuned checkpoints, while two teams took part with inference-time approaches guided by process reward models.⁴ All fine-tuned submissions were evaluated with temperature set to 0 and a max token limit of 5000. For the two special submissions, we used the default token limits and temperatures provided in

³<https://github.com/clp-research/clemcore>

⁴The two inference-time approaches decided not to submit a technical report.

their code, and set the search budget to best-of-4 sampled responses due to resource constraints.

Evaluation was run on a cluster of 4 x A100 80GB GPUs provided by the University of Potsdam. We evaluated the submitted checkpoints, along with 5 baseline models from the Qwen 3.5 (Qwen Team, 2026) and Llama 3.1 (Grattafiori et al., 2024) families, consuming 415 GPU hours.

3.5 Metrics and Scoring

The main score featured in Playpen (following clembench) is the **Clemscore**. For each game, it combines *% Played*, the percentage of instances completed, with a game-specific *Quality Score*:

$$\text{Clemscore} = \text{Quality Score} \times \frac{\% \text{ Played}}{100}. \quad (1)$$

The resulting scores are macro-averaged across games. Performance on the five static benchmarks, instead, is averaged into the **Statscore**. We refer to performance on the clembench games and static benchmarks from the official Playpen validation subset as Validation Clemscore and Validation Statscore, respectively. Following Horst et al. (2025), we report the Clemscore separately for in-domain (ID) and out-of-domain (OOD) games and calculate their changes relative to the corresponding baseline models. We use these scores and their relative changes to derive three further metrics.

To combine our two objectives of improving gameplay while not regressing on general benchmarks, the **adjusted delta** (Δ_{adj}) corrects the change in Clemscore for any decline in static benchmark performance:

$$\Delta_{\text{adj}} = \Delta + \min(\Delta_{\text{Statscore}}, 0). \quad (2)$$

Here, Δ is the change in the ID or OOD game score relative to the baseline, and $\Delta_{\text{Statscore}}$ is the change in Statscore. A decline in Statscore reduces the adjusted delta, whereas an increase leaves it unchanged, such that Δ_{adj} penalises models whose performance improves at the expense of degraded general ability.

The **generalisation gap** (GG) measures how far OOD performance lags behind ID performance:

$$\text{GG} = S_{\text{OOD}} - S_{\text{ID}}, \quad (3)$$

where S_{OOD} and S_{ID} are the model’s respective scores. Large negative gaps indicate poor generalisation.

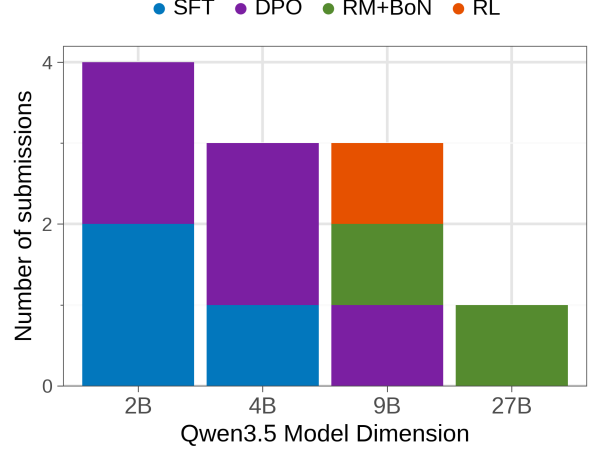


Figure 1: Number of submission by baseline model and improvement approach. Submission blending approaches are subsumed under what came last in their pipeline, e.g., SFT+DPO is counted as DPO.

The **proportion of error reduction** (PER) allows improvements from models with different baseline scores to be directly compared. Larger models generally start from stronger baselines and therefore have less room to improve, so an equal gain in score does not represent an equal reduction in the remaining error. PER divides the gain, Δ , by the difference between the baseline score, B , and the maximum score of 100:

$$\text{PER} = \frac{\Delta}{100 - B}. \quad (4)$$

It therefore expresses the gain as a proportion of the improvement still possible from that baseline.

We excluded Hot Air Balloon from the in-domain set of games because baseline models consistently hit the time wall.

4 Submissions

Challenge participants were asked to submit a paper through OpenReview describing their methodology, experimental setup, training data, and training and validation results. They were also required to submit a model to the Hugging Face Hub, with a model card containing the checkpoint, configuration files, and training card.

We received 11 model submissions from eight teams, six of which submitted papers to OpenReview: CityUoL, DAIR, playornotplay, Bentel rockers, Dialogue Architects, Large Language Problems. Although the challenge placed no restrictions on model choice, all submissions used models from the Qwen3.5 family. As reported in

Model	Team	ID Δ	OOD Δ	External Benchmarks	Adjusted ID Δ	Adjusted OOD Δ	Generalization Gap	Error Reduction (ID)	Error Reduction (OOD)
qwen3.5-2b		6.70	3.72	41.92	—	—	—	—	—
Qwen-GuidePlay-2B-v1	CityUoL 🐼	40.62	6.53	-2.06	38.56	4.47	-37.07	43.54%	6.78%
sft-dpo-v2	DAIR ★	46.42	10.84	-2.40	44.02	8.44	-38.56	49.75%	11.26%
sft-v1	DAIR	41.54	11.90	-4.55	36.99	7.35	-32.62	44.52%	12.36%
playornotplay-v1.0	playornotplay ✂	33.54	4.16	1.01	33.54	4.16	-32.36	35.95%	4.32%
qwen3.5-4b		29.28	17.99	53.53	—	—	—	—	—
Bentel_iter	Bentel rockers 🦖	4.02	-1.39	-3.21	0.81	-4.60	-16.70	5.68%	-1.69%
Bentel_iter_2	Bentel rockers	-0.81	-7.80	-9.89	-10.70	-17.69	-18.28	-1.15%	-9.51%
Bentel_iter_3	Bentel rockers	4.46	-1.39	-3.21	1.25	-4.60	-17.14	6.31%	-1.69%
qwen3.5-9b		37.37	24.91	53.85	—	—	—	—	—
prm-search-best_of_n	BSU-SLIM ▼	0.17	-9.73	-7.09	-6.92	-16.82	-22.36	0.27%	-12.96%
SCoRe_Qwen3.5-9B	Dialogue Architects ■	2.72	-1.54	0.48	2.72	-1.54	-16.72	4.34%	-2.05%
llp-final	Large Language Problems ▲	22.82	-2.82	0.64	22.82	-2.82	-38.10	36.44%	-3.76%
qwen3.5-27b		64.40	43.51	63.35	—	—	—	—	—
prm-guided-best_of_n	SLED-BSU ▼	-10.43	-13.49	0.86	-10.43	-13.49	-23.95	-29.30%	-23.88%

Table 1: Challenge results. Rows with a grey background report the absolute scores of the baseline models; for submitted models all columns report differences from the corresponding baseline model. External Benchmarks reports the static-game ClemScore. Icons in the Team column are the team markers used in the result figures. Bold values with a pale highlight mark the category leaders highlighted in Section 5.

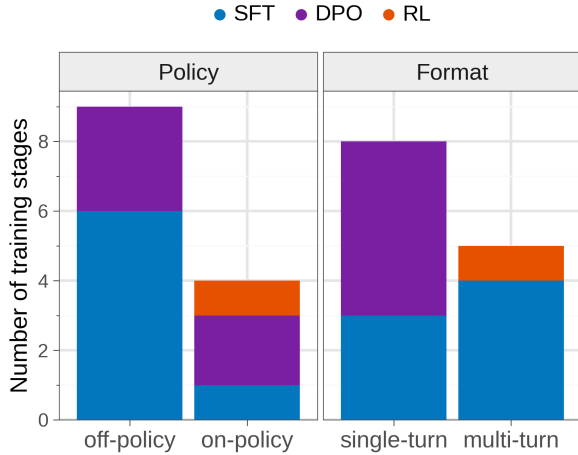


Figure 2: Policy and format of training strategies. Submissions blending approaches (e.g., SFT followed by DPO) are counted twice.

Fig. 1, the majority of submissions focused on improving smaller models, particularly the 2B version of Qwen3.5. The approaches included supervised fine-tuning (SFT), direct preference optimisation (DPO), on-policy reinforcement learning (RL), and an inference-time strategy combining a reward model with best-of- N sampling (RM+BoN). SFT was the most common approach, used by five of the six teams that submitted papers; Dialogue Architects was the exception. Four teams subsequently applied DPO. On-policy RL was used only in the final system submitted by Dialogue Architects; DAIR and playornotplay also reported experimenting with online RL but did not include it in their final submissions.

Figure 2 breaks down the training strategies

by data provenance (on-policy or off-policy) and interaction format (single-turn or multi-turn), following the categorisation in Horst et al. (2025). Most SFT approaches used off-policy data from Playpen, while DPO approaches were more evenly distributed across the two categories. All teams that used DPO experimented with single-turn data, whereas SFT was applied to both single-turn and multi-turn data. The Dialogue Architects submission was the only one to use online, on-policy training on multi-turn data.

5 Competition Results

Table 1 reports the challenge results for all submitted systems. DAIR’s 2B SFT+DPO system achieved the largest adjusted in-domain improvement (+44.02), in-domain error reduction (49.75%), and adjusted OOD improvement (+8.44). DAIR’s SFT-only variant achieved the largest OOD error reduction (12.36%). The largest improvement on the static benchmarks was achieved by playornotplay (+1.01).

6 Findings

Gains over the baseline are largest for the smallest model category. As shown in Figure 3, the largest absolute gains occur among the 2B systems, which also have the most room to improve relative to their baseline model. Results at larger model sizes are more mixed. The 4B and 9B systems preserve in-domain performance or achieve moderate to sizeable improvements, but often lose OOD performance. The 27B system, built on the

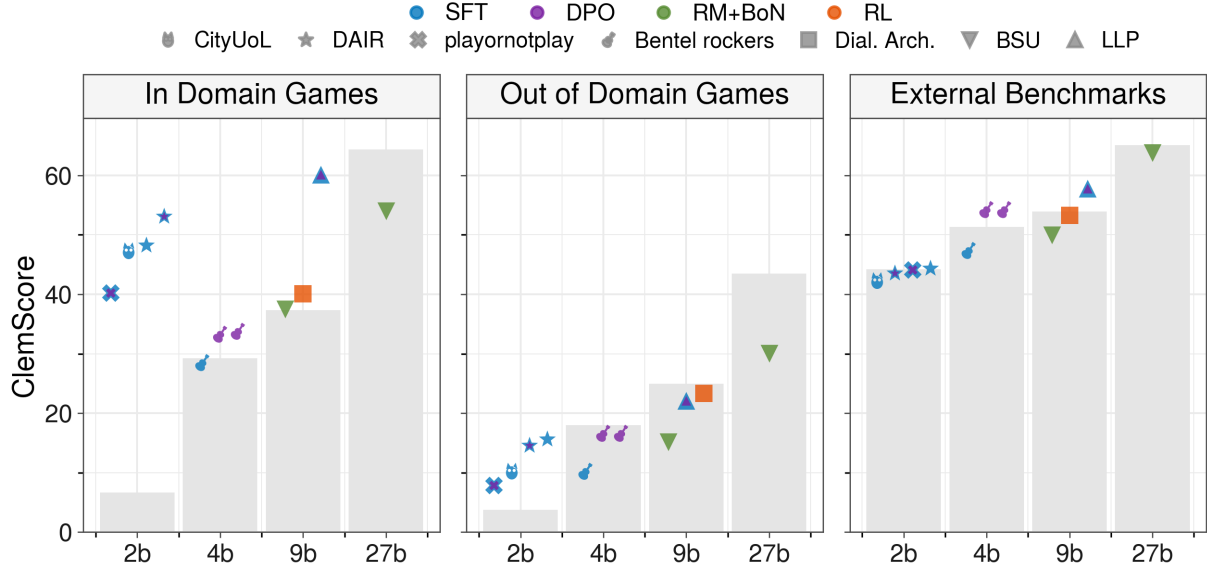


Figure 3: Results for training strategies: grey bars are the baseline model results, the coloured points are the results for each submission – divided in In Domain games, Out of Domain games, and External Benchmarks. Each submission is again categorised according to the training strategy. Teams identity is identifiable through marker icons.

strongest base model, regresses both with respect to in-domain and OOD performance.

Post-training changes the pattern of performance across games. Figure 6 shows that post-training does not simply raise or lower performance uniformly across the benchmark. At 2B, the SFT traces move outward most clearly on the TextMap-World games and Deal or No Deal, while retaining the baseline model’s already high ImageGame score. Other game-specific weaknesses remain. At larger model sizes, post-trained variants likewise differ from their baseline models in which games improve, remain difficult, or decline.

OOD and static-benchmark gains are narrow and do not reliably track in-domain trends. Figure 3 shows that, at the aggregate level, in-domain improvements do not translate consistently into proportionate OOD gains. At the game level, OOD gains are concentrated in a subset of OOD games (particularly in Wordle variants) rather than spread across the full suite, as illustrated in Figure 6. Likewise, both aggregate and benchmark-level changes in static benchmark performance do not consistently track changes in interactive-game performance (see Figure 3 and Figure 6). Together, these results indicate a disconnect between aggregate improvements and a broader, game-independent ability gain.

Training with Dialogue Games Teaches to Follow Game Rules. The two metrics composing ClemScore measure different aspects of game play: % Played games is a proxy for the models’ instruction-following abilities in the game domain, while the Quality Score for models’ abilities in game playing. The aggregate results in Figure 4 show that the strongest 2B systems improve both these aspects. At larger model sizes, however, changes are less consistently aligned. In Domain for 4B, 9B and 27B submissions, game-level changes are mixed, with % Played games often improving where Quality Score tend to decrease with the exception of the LLP submission. Fig. 10, for the ID Taboo game, shows the mixed deltas in terms of quality score for a game that baseline Qwen models are able to play. Out of Domain, submissions tend to also complete more episodes, while their quality is lower. An example of an OOD game is Wordle Crazy in Fig. 11, where the improvements are mostly in terms of playability. In conclusion, training with dialogue games, with no emerging distinction among training strategies, makes models follow more strictly the structure and rules of the games. This is further confirmed by Fig. 9, showing that the most OOD quality improvements are in games that are structurally similar to ID ones (Wordle variants).

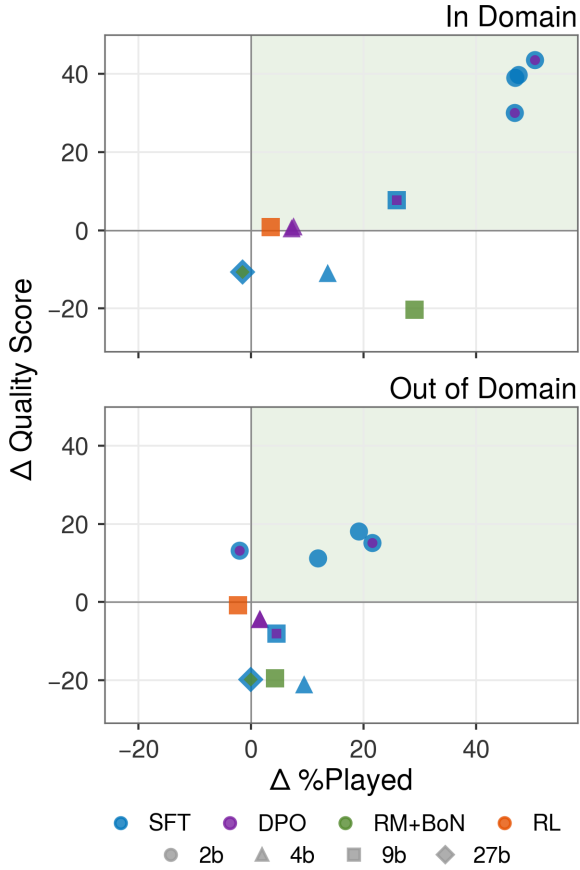


Figure 4: Submissions’ Δ % Played instances and Δ Quality Score for In Domain and Out of Domain Games, classified according training method and baseline size. 2B submissions tend to improve both these aspects, with the exception of playornotplay OOD, where unexpectedly it only improves quality score. In general, SFT + DPO seems to produce consistent results for ID games. For larger baselines, models tend to only increase the % of played game, with mixed performances ID and decreased in OD.

Dialogue Games Training Makes Language More Concise. Post-training with Dialogue Game improves following games rules and structure, does this benefit performances on external benchmarks? Table 2 shows that gains are obtained for BBH, with decreasing performances for IFEval in particular (replicating Horst et al., 2025). We observe that trained models tend to generate more concise answers, an interesting exception for this is the RL/SCoRE method, for which the output length is not significantly shorter on average. In BBH more concise outputs improve performances, since baseline models’ outputs suffer from formatting errors (i.e., unexpected chain-of-thought and complete repetition of the answer), that are avoided

by training with game language.⁵ As shown in Fig. 5, however, these improvements are balanced by decreases in terms of corrected answers on the remaining questions.

Policy and Dialogue Format. Following the distinction in Horst et al. (2025), Fig. 7 reports the influence of a series of factors on the final deltas. In particular, we consider: 1) the training regime: token vs turn loss, online vs offline training; and 2) the type of data used for training: off-policy vs on-policy, single-turn vs multi-turn. These patterns should be interpreted as trends rather than conclusive results. Nevertheless, off-policy data and multi-turn formats appear to yield consistent improvements in games, both in and out of domain. Learning off-policy from multi-turn dialogues means learning from complete demonstrations generated by stronger models, which may provide information about the overall game structure rather than a meaningful turn-level learning signal. Regarding online versus offline learning, the only online approach, SCoRE/RL, is notably conservative, as already shown in Fig. 4 and Table 2.

7 Conclusions

In this technical report, we discuss the outcome of the first LM Playschool 2026 challenge. The main goal of this challenge is to provide a controllable environment based on dialogue games for simulating situated language use and for studying how models learn from and through interaction, instead of passively observing dialogue data. To evaluate whether the resulting models are well-rounded language users, we first use unseen instances of dialogue games that the model has seen at training time. However, to make sure that models develop language abilities that do not exclusively apply to training data, our experimental setup values different forms of generalisation by evaluating: 1) performance on out-of-domain dialogue games, and 2) performance on reference-based NLP benchmarks.

The challenge received great interest with 11 submissions across a variety of different model sizes ranging from 2B up to 27B. Based on the categorisation of learning-from-interaction approaches proposed by Horst et al. (2025), most approaches focused on off-policy training regimes, while only

⁵This finding is accentuated by the BBH evaluation prompt, which does not specify the answer format and results in more errors by the baselines models, due to verbosity.

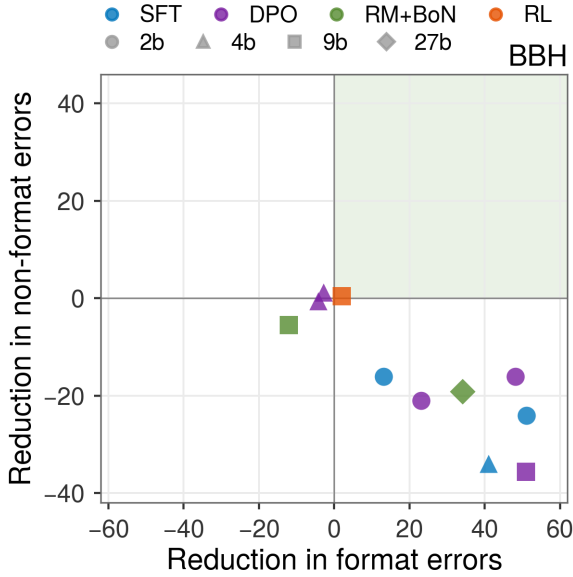


Figure 5: BBH error composition relative to the base model. For BBH questions ($n = 104$), every incorrect episode is classified as: a “format error”, where the error is due to a format issues (the answer is reached but buried in reasoning or prefixed, e.g. A: (C)), or “non-format error” for the remaining cases. Both axes are differences in instances counts, computed as reduction in errors (base – submission), so a positive value means the submission made fewer errors of that class than its baseline model; the shaded upper-right quadrant therefore marks a reduction in both classes at once. Marker shape denotes the base model, colour the terminal stage of the fine-tuning pipeline.

one participant used an on-policy training regime with Reinforcement Learning. Based on our results, current learning methods provide substantial gains, particularly for small language models, with their performance matching or sometimes exceeding base model performance on in-domain games. However, generalisation remains limited, suggesting that models tend to learn from the surface form of the language outputs they observe.

While the submissions demonstrate that post-training on recorded Playpen interactions can produce substantial improvements, these improvements are uneven across interactive games. The feedback taxonomy also identifies three promising directions that the challenge did not test: training directly on within-episode Game Master corrections, refreshing preference data as the policy improves, and learning from feedback supplied—implicitly or explicitly—by a co-player. Evaluating these regimes would extend the challenge from learning game protocols from recorded trajectories towards learning to adapt to feedback during interaction.

Challenge submissions did not experiment with chain-of-thought reasoning, and training made the models’ language more concise rather than more strategic. Related works have demonstrated that training with chain of thought on competitive (Liu et al., 2026) and cooperative (Yuan et al., 2026) games can teach models more strategic chain-of-thought reasoning, leading to improvements on general benchmarks. Given the number of games and high-level abilities captured by the Playpen game environment, incorporating reasoning traces into training could lead to valuable improvement.

Modelling the multi-turn nature of dialogue games is essential to perform correct state-tracking across turns. We believe that the Playpen environment will facilitate the implementation of multi-turn training regimes that can leverage signals that can be derived across multiple turns of interaction.

Finally, building on previous work on multi-modal dialogue games (Hakimov et al., 2025), studying the intersection between communicative grounding and visual grounding represents a promising research direction for future multimodal conversational agents (Suglia et al., 2024).

Acknowledgements

We thank Uday Bhaskar, Kranti Chalamalasetti, Shahrukh Mohiuddin, Karl Osswald, and Ruilin Yang for their contributions to the out-of-distribution games. We gratefully acknowledge the support of the Deutsche Forschungsgemeinschaft (DFG), the Autonomous Province of Bozen–Bolzano, and Amazon. This work was conducted as part of the LLADIGA project under the Joint Projects Italia–Germania 2025 programme. Finally, we thank ELLIS for bringing together the organisers and facilitating this collaboration.

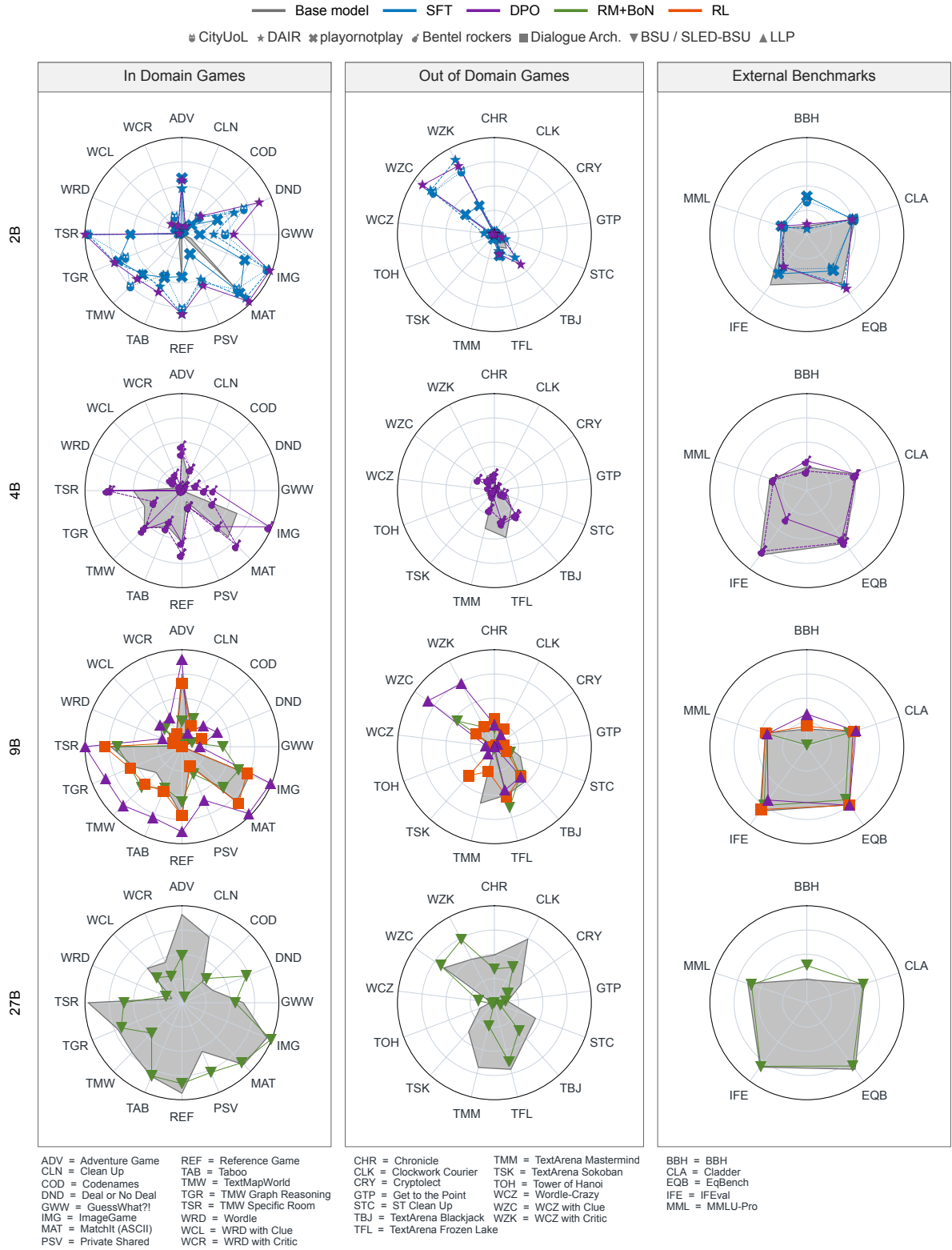


Figure 6: Per-game ClemScore profiles for every base-model size (2B, 4B, 9B, 27B) and its submitted post-trained variants. Each spoke denotes a game or benchmark, and radial distance gives the score (0–100). The grey trajectory is the corresponding base model; coloured trajectories are its post-trained variants. Marker icons identify the submitting team, matching Table 1; colours identify the final-stage method. Differences from the grey line show which game-specific strengths and weaknesses post-training preserved or changed.

A Detailed results

The following plots complement the aggregate and profile analyses in Section 6. Table 2 illustrates how individual benchmark-level changes contribute to an aggregate increase or decrease in external benchmark performance. The training-dimension map in Figure 7 situates submissions in the feedback taxonomy, providing descriptive diagnostics rather than controlled estimates of a training-design effect. The radar plots in Figures 8 and 9 unpack the ClemScore profiles presented in Figure 6 into their two component parts: outward movement against the base-model in the Quality Score profiles indicates better play on that game, whereas outward movement in the percentage-played profiles indicates that more episodes were completed. Comparing the two identifies whether a game-level change reflects quality, completion, or both. Finally, the plots in Figures 10 and 11 isolate concrete example cases where post-training flipped a base model’s behaviour on an instance-level.

Model	Quality Score					Mean words generated				
	BBH	CLad.	EQ-B.	IFE.	MMLU	BBH	CLad.	EQ-B.	IFE.	MMLU
qwen3.5-2b	8.65	52.00	63.20	64.00	23.00	196.1	483.3	8.0	611.3	6.4
GuidePlay-2B	25.97	0.00	-15.00	-20.00	2.00	1.5	1.2	8.0	346.4	4.0
sft-dpo-v2	1.93	-3.00	5.84	-23.00	5.00	68.2	73.7	8.0	228.0	4.0
sft-v1	-2.88	-2.00	2.23	-23.00	2.00	77.5	141.8	8.0	577.5	4.0
playornotplay	30.77	-2.00	-11.11	-14.00	5.00	84.6	220.6	8.0	433.7	4.0
qwen3.5-4b	24.04	54.00	68.68	82.00	40.00	150.2	612.8	26.2	193.7	4.0
Bentel_iter	-3.85	-2.00	-1.28	-5.00	-4.00	220.4	618.9	8.9	370.7	4.0
Bentel_iter_2	6.73	0.00	-6.79	-46.00	-4.00	25.9	93.2	9.5	261.5	4.0
Bentel_iter_3	-3.85	-2.00	-1.28	-5.00	-4.00	220.4	618.9	8.9	370.7	4.0
qwen3.5-9b	18.27	50.00	73.99	82.00	45.00	179.7	702.7	8.0	191.0	4.0
prm-search-bon	-16.35	-4.00	-5.59	-7.00	-1.57	224.7	360.3	20.0	196.6	5.2
SCoRe-9B	2.88	1.00	0.51	-2.00	0.00	172.0	718.3	8.0	182.1	4.0
llp-final	15.38	3.00	1.47	-14.00	-1.57	15.4	180.3	12.4	168.5	4.0
qwen3.5-27b	24.04	62.00	84.72	82.00	64.00	191.7	755.3	8.0	193.4	4.0
prm-guided-bon	14.42	-1.00	-4.14	-1.00	-4.00	61.3	299.9	8.0	178.4	4.0

Table 2: External benchmark results deltas (Quality Score), with the mean number of words generated per question (grey columns). Score columns show absolute values for base models and differences to the base for submissions; length columns are absolute in both cases. **Bold** marks a significant reduction in output length relative to the base model (Wilcoxon signed-rank over paired questions, $p < 0.05$).

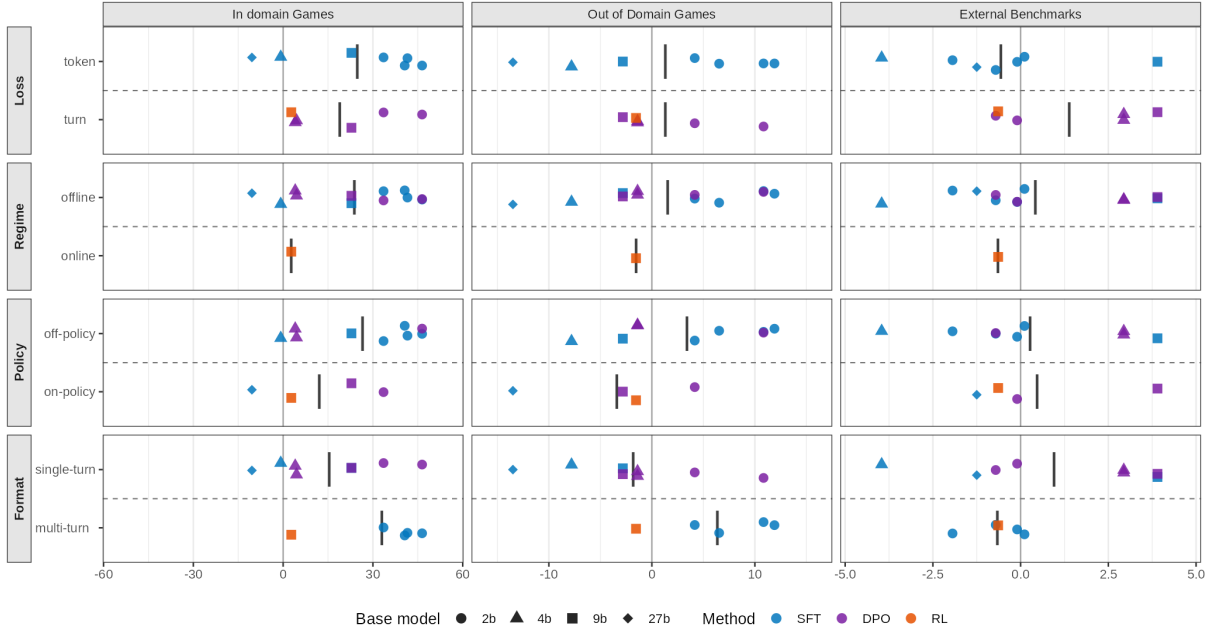


Figure 7: Submission-level map of score changes by four annotated training dimensions: loss granularity, training regime, policy provenance, and dialogue format. Each coloured point is a post-trained submission’s change from its base model; colours denote method and shapes denote base-model size. Black markers show the corresponding base-model results. A submission can appear in more than one category when its pipeline combines training stages. This figure is descriptive: the categories overlap and are not controlled comparisons of the effects of individual design choices.

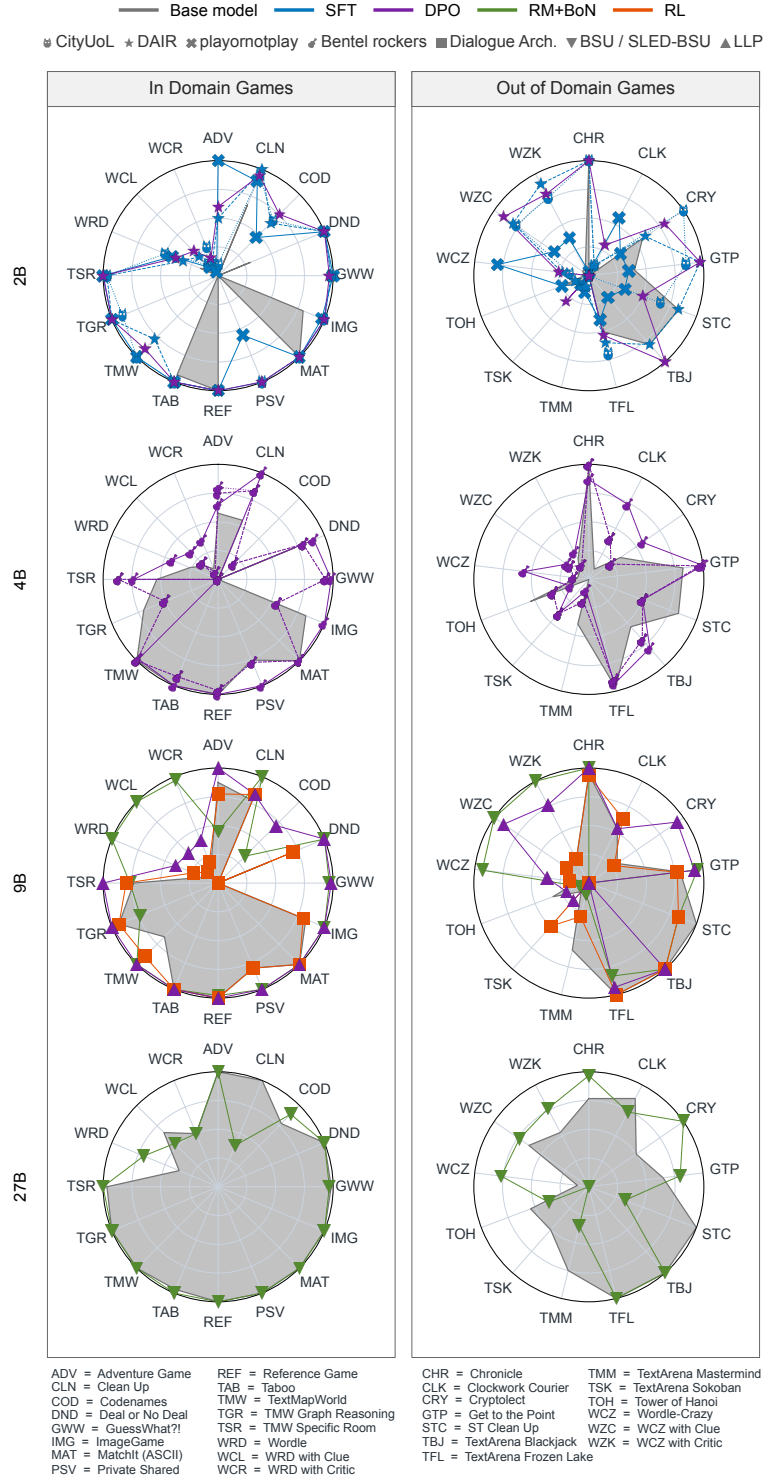


Figure 8: Per-game percentage-played profiles for every base-model size (2B, 4B, 9B, 27B) and its submitted post-trained variants. This decomposes the ClemScore profiles shown in Figure 6. Each spoke denotes a dynamic game, and radial distance gives the percentage of episodes completed (0–100). The grey trajectory is the corresponding base model; coloured trajectories are its post-trained variants. Marker icons identify the submitting team, matching Table 1; colours identify the final-stage method. Differences from the grey line show which game-specific completion strengths and weaknesses post-training preserved or changed. Static benchmarks are omitted because completion is near-saturated and does not materially aid interpretation.

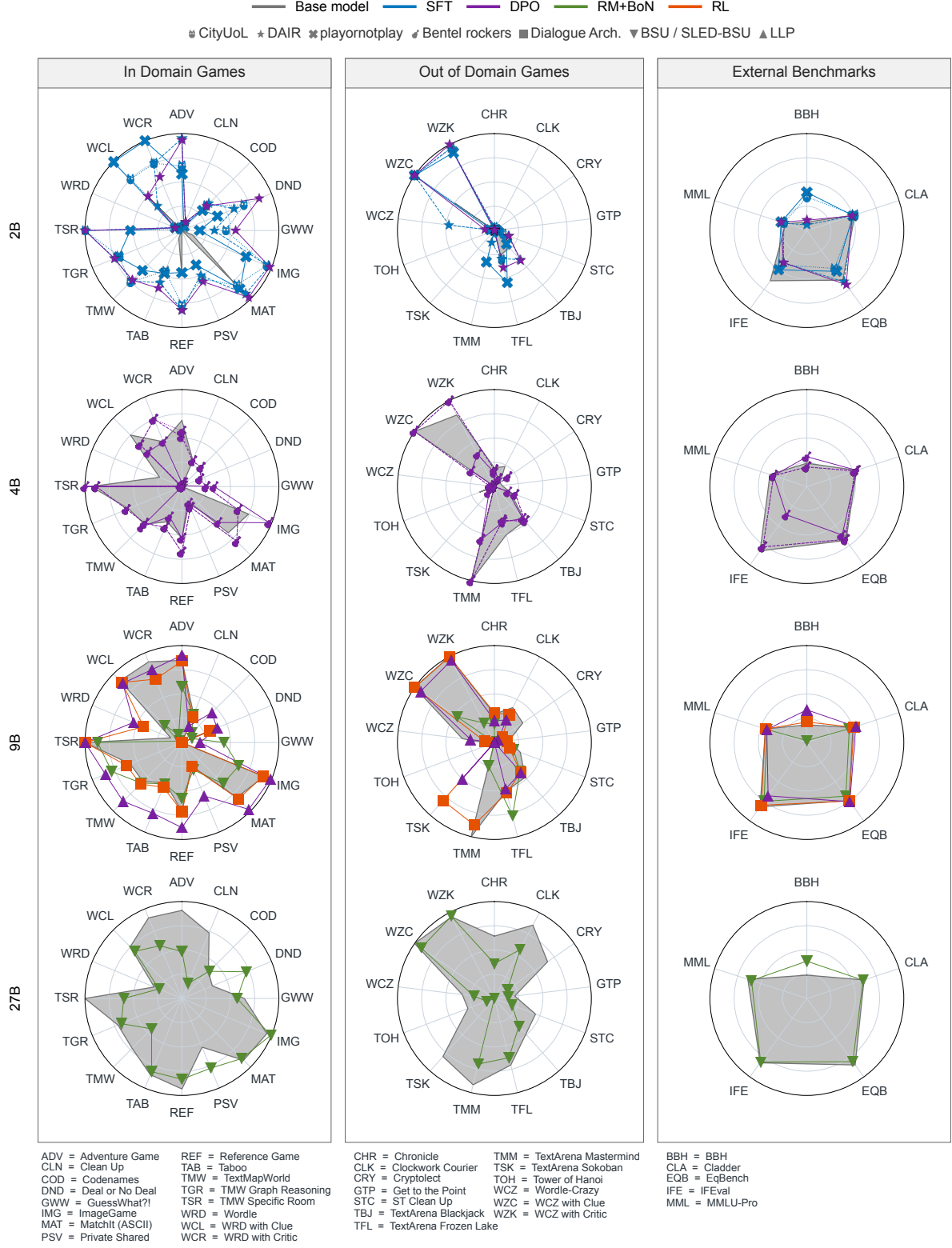


Figure 9: Per-game Quality Score profiles for every base-model size (2B, 4B, 9B, 27B) and its submitted post-trained variants. This decomposes the ClemScore profiles shown in Figure 6. Each spoke denotes a game or benchmark, and radial distance gives mean Quality Score (0–100). The grey trajectory is the corresponding base model; coloured trajectories are its post-trained variants. Marker icons identify the submitting team, matching Table 1; colours identify the final-stage method. Differences from the grey line show which game-specific strengths and weaknesses post-training preserved or changed.

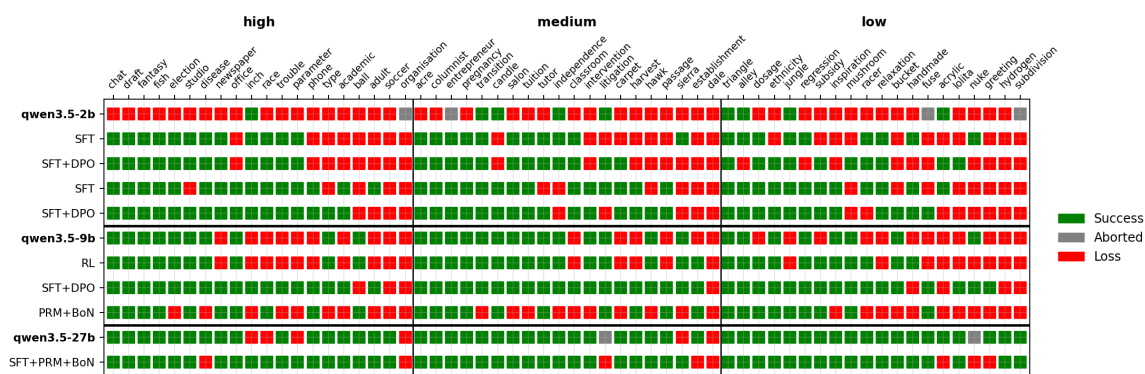


Figure 10: ID Taboo instances flipped from the base models to the trained models. The Qwen baselines are able to play the game, with very few aborted instances even for the smallest 2B baseline. Trained submissions lead to improvements in terms of quality scores (flipped instances).

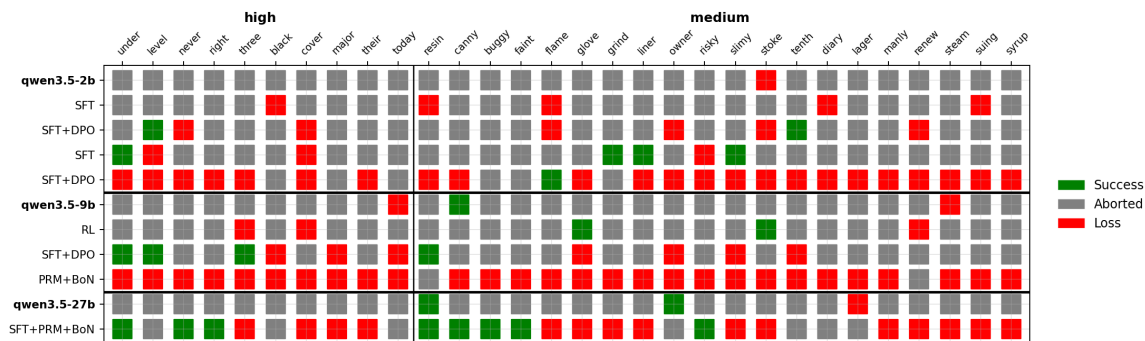


Figure 11: OOD Wordle Crazy instances flipped from the base models to the trained models. Qwen baselines tend to abort all episodes, trained submissions are able to follow the rules of the game and play, but mostly losing it (reflected in the quality score).

References

- Emily M Bender and Alexander Koller. 2020. Climbing towards NLU: On Meaning, Form, and Understanding in the Age of Data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 5185–5198. Issue: 2.
- Anne Beyer, Kranti Chalamalasetti, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2024. [clembench-2024: A Challenging, Dynamic, Complementary, Multilingual Benchmark and Underlying Flexible Framework for LLMs as Multi-Action Agents](#). *arXiv preprint*. ArXiv:2405.20859 [cs].
- Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. 2020. [Experience grounds language](#). *EMNLP 2020 - 2020 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, pages 8718–8735. ArXiv: 2004.10151 ISBN: 9781952148606.
- Paul Bloom. 2000. *How children learn the meanings of words*. Learning, development, and conceptual change. MIT Press, Cambridge.
- Kranti Chalamalasetti, Jana Götze, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2023. [clembench: Using game play to evaluate chat-optimized language models as conversational agents](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11174–11219, Singapore. Association for Computational Linguistics.
- Pengyu Cheng, Tianhao Hu, Han Xu, Zhisong Zhang, Yong Dai, Lei Han, Nan Du, and Xiaolong Li. 2024. [Self-playing Adversarial Language Game Enhances LLM Reasoning](#).
- Eve V. Clark. 2016. *First language acquisition*, 3rd ed edition. Cambridge University Press, Cambridge.
- Alejandrina Cristia, Emmanuel Dupoux, Michael Gurven, and Jonathan Stieglitz. 2019. Child-directed speech is infrequent in a forager-farmer population: A time allocation study. *Child development*, 90(3):759–773.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Leon Guertler, Bobby Cheng, Simon Yu, Bo Liu, Leshem Choshen, and Cheston Tan. 2025. [Textarena](#). *Preprint*, arXiv:2504.11442.
- Mustafa Omer Gul and Yoav Artzi. 2024. [CoGen: Learning from Feedback with Coupled Comprehension and Generation](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12966–12982, Miami, Florida, USA. Association for Computational Linguistics.
- Sherzod Hakimov, Yerkezhan Abdullayeva, Kushal Koshti, Antonia Schmidt, Yan Weiser, Anne Beyer, and David Schlangen. 2025. [Using game play to investigate multimodal and conversational grounding in large multimodal models](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5686–5718, Abu Dhabi, UAE. Association for Computational Linguistics.
- Betty Hart and Todd R. Risley. 1995. *Meaningful differences in the everyday experience of young American children*. Paul H. Brookes Publishing Co.
- Nicola Horst, Davide Mazzaccara, Antonia Schmidt, Michael Sullivan, Filippo Momentè, Luca Franceschetti, Philipp Sadler, Sherzod Hakimov, Alberto Testoni, Raffaella Bernardi, Raquel Fernández, Alexander Koller, Oliver Lemon, David Schlangen, Mario Giulianelli, and Alessandro Suglia. 2025. [Playpen: An environment for exploring learning from dialogue game feedback](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29854–29891, Suzhou, China. Association for Computational Linguistics.
- Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, and Bernhard Schölkopf. 2024. [Cladder: Assessing causal reasoning in language models](#). *Preprint*, arXiv:2312.04350.
- Jiatong Li, Rui Li, and Qi Liu. 2023. Beyond Static Datasets: A Deep Interaction Approach to LLM Evaluation. ArXiv: 2309.04369v1.
- Tal Linzen. 2020. [How can we accelerate progress towards human-like linguistic generalization?](#) In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5210–5217, Online. Association for Computational Linguistics.
- Bo Liu, Simon Yu, Zichen Liu, Leon Guertler, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, Wee Sun Lee, and Natasha Jaques. 2026. [SPIRAL: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning](#). In *The Fourteenth International Conference on Learning Representations*.
- Kyle Mahowald, Anna A Ivanova, Idan A Blank, Nancy Kanwisher, Joshua B Tenenbaum, and Evelina Fedorenko. 2024. Dissociating language and thought in large language models. *Trends in cognitive sciences*.
- Filippo Momentè, Alessandro Suglia, Mario Giulianelli, Ambra Ferrari, Alexander Koller, Oliver Lemon,

- David Schlangen, Raquel Fernández, and Raffaella Bernardi. 2025. [Triangulating LLM progress through benchmarks, games, and cognitive tests](#). *arXiv preprint arXiv:2502.14359*.
- OpenEnv Technical Committee. 2025. OpenEnv: Agentic Execution Environments for Reinforcement Learning and Post-Training. <https://github.com/huggingface/OpenEnv>.
- Samuel J. Paech. 2023. [Eq-bench: An emotional intelligence benchmark for large language models](#). *Preprint*, arXiv:2312.06281.
- Dan Qiao, Chenfei Wu, Yaobo Liang, Juntao Li, and Nan Duan. 2023. [GameEval: Evaluating LLMs on Conversational Games](#). ArXiv: 2308.10032.
- Qwen Team. 2026. [Qwen3.5: Towards native multi-modal agents](#).
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: your language model is secretly a reward model. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, pages 53728–53741, Red Hook, NY, USA. Curran Associates Inc.
- David Schlangen. 2023. [Dialogue games for benchmarking language understanding: Motivation, taxonomy, strategy](#). *CoRR*, abs/2304.07007.
- David Schlangen, Sherzod Hakimov, Chalamalasetti Kranti, Jonathan Jordan, and Philipp Sadler. 2026. [A third paradigm for llm evaluation: Dialogue game-based evaluation using clembench](#). *Preprint*, arXiv:2507.08491.
- Aarohi Srivastava, Abhinav Rastogi, Abhishek Rao, Abu Awal Md Shoeb, Abubakar Abid, Adam Fisch, Adam R. Brown, Adam Santoro, Aditya Gupta, Adrià Garriga-Alonso, Agnieszka Kluska, Aitor Lewkowycz, Akshat Agarwal, Alethea Power, Alex Ray, Alex Warstadt, Alexander W. Kocurek, Ali Safaya, Ali Tazary, and 431 others. 2023. [Beyond the imitation game: Quantifying and extrapolating the capabilities of language models](#). *Transactions on Machine Learning Research*.
- Alessandro Suglia, Ioannis Konostas, and Oliver Lemon. 2024. [Visually Grounded Language Learning: a review of language games, datasets, tasks, and models](#). *Journal of Artificial Intelligence Research*, 79:173–239. ArXiv: 2312.02431.
- Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. [Challenging BIG-bench tasks and whether chain-of-thought can solve them](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.
- J Terry, Benjamin Black, Nathaniel Grammel, Mario Jayakumar, Ananth Hari, Ryan Sullivan, Luis S Santos, Clemens Dieffendahl, Caroline Horsch, Rodrigo Perez-Vicente, et al. 2021. Pettingzoo: Gym for multi-agent reinforcement learning. *Advances in Neural Information Processing Systems*, 34:15032–15043.
- Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U. Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulão, Andreas Kallinteris, Markus Krimmel, Arjun KG, Rodrigo Perez-Vicente, Andrea Pierré, Sander Schulhoff, Jun Jet Tai, Hannah Tan, and Omar G. Younis. 2025. [Gymnasium: A standard interface for reinforcement learning environments](#). *Preprint*, arXiv:2407.17032.
- Ruiyi Wang, Haoifei Yu, Wenxin Zhang, Zhengyang Qi, Maarten Sap, Yonatan Bisk, Graham Neubig, and Hao Zhu. 2024a. [SOTOPIA-pi: Interactive Learning of Socially Intelligent Language Agents](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12912–12940, Bangkok, Thailand. Association for Computational Linguistics.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhui Chen. 2024b. Mmlu-pro: a more robust and challenging multi-task language understanding benchmark. In *Proceedings of the 38th International Conference on Neural Information Processing Systems, NIPS '24*, Red Hook, NY, USA. Curran Associates Inc.
- Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. 2023. An Empirical Study on Werewolf. ArXiv: 2309.04658v1.
- Huining Yuan, Zelai Xu, Zheyue Tan, Xiangmin Yi, Mo Guang, Kaiwen Long, Haojia Hui, Boxun Li, Xinlei Chen, Bo Zhao, Xiao-Ping Zhang, Chao Yu, and Yu Wang. 2026. [MARSHAL: Incentivizing multi-agent reasoning via self-play with strategic LLMs](#). In *The Fourteenth International Conference on Learning Representations*.
- Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. [Instruction-following evaluation for large language models](#). *Preprint*, arXiv:2311.07911.

Enhancing LLM Performance via LoRA SFT and DPO for the LM Playschool Challenge

Vasamsetti Nihal Tej

IIT Bhubaneswar

23cs01071@iitbbs.ac.in

Velineni Harshavaishnav

IIT Bhubaneswar

23cs01067@iitbbs.ac.in

Shreya Ghosh

IIT Bhubaneswar

shreya@iitbbs.ac.in

Abstract

Large Language Models (LLMs) have achieved SOTA results in many natural language processing tasks. However, they are primarily trained for static text generation and often struggle in interactive, goal-oriented environments that require multi-turn reasoning, state tracking, and need to give structured output formats. This paper presents our team’s submission to the LM Playschool Workshop Shared Task at EMNLP 2026. The LM Playschool Workshop Shared Task at EMNLP 2026 addresses this challenge by evaluating language models as interactive agents in structured dialogue games. We hereby explored different fine-tuning strategies for Qwen 4B and Qwen 7B models to improve their interactive capabilities and evaluated them. We perform a comparison of Supervised Fine-Tuning (SFT) and Direct Preference Optimization (DPO) using multi-turn training data. Our study analyzes the effectiveness of each optimization strategy for learning robust dialogue behaviors while maintaining models stability towards interactive games and also not to forget its language linguistics by different evaluating metrics. Experimental results show that although supervised fine-tuning provides consistent interactive performance, directly optimizing the foundation model using DPO reduces optimization errors and leads to improved agent behavior. Our best-performing model achieved a **clmscore of 38.23** on the official LM Playschool leaderboard, evaluated using the Playpen workbench.

1 Introduction

Most Large Language Models (LLMs) are pre-trained on static text corpora and are not designed to learn or operate effectively in interactive environments. Although instruction tuning aligns LLMs for single-turn, open-ended tasks, they still struggle to function as goal-oriented interactive agents. As, Multi-turn dialogue games require models to maintain dialogue history, accurately track game states,

follow structural output formats, and also need to satisfy environment-specific constraints throughout long interactions to maintain correctness.

One major challenge is that pretrained LLMs often fail to consistently follow the required response format specified by the user or the environment, we need to also ensure model may lose its language knowledge when training for interactive games. Dialogue games provide an effective benchmark for evaluating and improving these interactive capabilities, these workbenches expect a strict output format in the interactions else they may go into abort/lose state.

The LM Playschool Shared Task, built on the Playpen-2025 benchmark, evaluates agentic capabilities through conversational board games and dialogue games using the `clmbench` workbench. Models are required to correctly interpret user feedback, maintain implicit game states across multiple turns, and generate concise, structured responses that strictly follow the required output format, such as `Instruction: [Action]` followed by state markers like `DONE`.

In this work, we present our complete training pipeline and the techniques used to improve interactive dialogue capabilities. During development, we encountered two major challenges: handling multi-turn dialogue data and designing effective data processing pipelines, and optimizing fine-tuning strategies. Our main contributions are summarized as follows:

1. We design a data processing pipeline to handle multi-turn dialogue interactions and generate training data for SFT and DPO. Since each training method requires a different data format, we develop separate preprocessing and data generation pipelines for each approach.
2. We further analyze the effect of different hyperparameter settings on the interactive performance of these models by evaluating with

the playpen workbench, which contains two primary scores: clemscore and statscore.

2 Data Preprocessing and Engineering Pipeline

Our primary training corpus is derived from the raw interaction traces of the Playpen-2025 benchmark. The benchmark contains interaction logs collected across four different versions of the Playpen environment. Each game session is stored as a complete multi-turn conversation under the `chat` field, represented as a list of nested dictionaries containing user and assistant messages. The complete dataset consists of **114,479 interaction traces**.

Since each optimization technique requires a different data format, we design separate preprocessing pipelines for SFT and DPO. Additionally, we perform a warm-up fine-tuning stage to familiarize the model with interactive dialogue generation. During this phase, we train the model using warmup data made up of strict, rule-based questions and answers. This teaches the model the basic rules of chatting, making sure it knows how to follow exact formats and instructions before it moves on to harder, more open-ended conversations. During this stage, the model is trained using a small learning rate to learn the required response format and follow the output structure specified by the input before proceeding to the main fine-tuning process.

2.1 Supervised Fine-Tuning (SFT)

We first filter the dataset to retain only successful interactions, resulting in **31,779 successful game sessions**. Rather than training only on the final successful responses, we generate intermediate training examples using a cumulative dialogue prefix extraction algorithm.

Let a dialogue session be represented as

$$S = \{u_1, a_1, u_2, a_2, \dots, u_T, a_T\}, \quad (1)$$

where u_t denotes the user (or environment) message and a_t denotes the assistant response.

For every assistant turn, we generate one training example by retaining the dialogue history up to and including the corresponding assistant response:

$$x_t = [u_1, a_1, \dots, u_t, a_t]. \quad (2)$$

During supervised fine-tuning, only the assistant tokens in a_t contribute to the causal language modeling loss. This cumulative dialogue prefix

strategy converts each successful dialogue into multiple training examples while preserving the complete conversational context. After using this strategy, we got **60,011 interactions**. The warm-up data is taken up from the [playpen-paper-2025](#) which contains **20,266 interaction traces**. Both of these datasets are combined and preprocessed using Qwen3.5 tokenizer. Upon removing the duplicates, our dataset has been reduced to **58,088 interactions**.

2.2 Direct Preference Optimization (DPO)

We first filter the dataset to retain only successful interactions, resulting in **31,779 successful game sessions**. To prepare the data for Direct Preference Optimization (DPO), the interaction logs are then grouped by the same `game`, `experiment` and `episode` identifiers from the **114,479 interactions**, ensuring that each preference pair is constructed from identical dialogue contexts. Successful responses are treated as preferred responses (y^+), whereas responses from *Lose* or *Aborted* episodes are treated as rejected responses (y^-). Each preference example is represented as

$$(x, y^+, y^-), \quad (3)$$

where x denotes the shared prompt, y^+ is the preferred response, and y^- is the rejected response. These prompt-preference triplets are then used to optimize the model using the DPO objective. Here, we set the positive-to-negative ratio for the DPO preference pairs to **1:2** which makes it to a total of **63,558 interactions**. This dataset is preprocessed with Qwen3.5 tokenizer. Upon removing the duplicates, our dataset has been reduced to **59,154 interactions**. DPO mathematically embeds the reward function directly within the model’s token log-probabilities, eliminating the need for an external reward network. Utilizing this metadata bypasses costly online sampling and multi-model hosting, ensuring highly efficient and stable optimization during multi-response comparison.

3 Methodology

Our model adaptation strategy is strictly divided into two phases, ensuring that the model first learns the desired output format before optimizing for preference alignment.

3.1 Low-Rank Adaptation (LoRA)

To reduce the memory overhead of updating billions of parameters, we used LoRA. LoRA freezes the pre-trained model weights $W_0 \in \mathbb{R}^{d \times k}$ and adds trainable rank decomposition matrices to every layer of Transformer architecture. The forward pass is modified as follows for a certain layer:

$$h = W_0 x + \Delta W x = W_0 x + B A x \quad (4)$$

where $B \in \mathbb{R}^{d \times r}$ and $A \in \mathbb{R}^{r \times k}$, and the rank $r \ll \min(d, k)$. We applied LoRA adapters mostly to the query and value projection matrices within the self-attention modules.

3.2 Supervised Fine-Tuning (SFT)

In the first phase, we trained the base model using standard autoregressive language modeling. Given a dataset of instruction-response pairs $\mathcal{D}_{\text{SFT}} = \{(x_i, y_i)\}_{i=1}^N$, we optimize the negative log-likelihood:

$$\mathcal{L}_{\text{SFT}} = -\mathbb{E}_{(x,y) \sim \mathcal{D}} \left[\sum_{t=1}^{|y|} \log P_{\theta}(y_t \mid x, y_{<t}) \right] \quad (5)$$

Only the LoRA parameters θ were updated during this phase.

3.3 Direct Preference Optimization (DPO)

As an alternative adaptation strategy, we fine-tuned the base model using DPO. Unlike Reinforcement Learning from Human Feedback (RLHF), which relies on training a separate reward model followed by reinforcement learning, DPO directly learns from pairwise preference data. For each prompt x , the training dataset contains a preferred response y_w and a rejected response y_l . This technique encourages the model to assign a higher probability to the preferred response while reducing the likelihood of the rejected response, thereby improving alignment with human preferences. Additionally, DPO regularizes the optimized policy to remain close to the reference model, resulting in an optimized approach that is both computationally efficient and stable. The DPO loss is defined as:

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E}_{(x,y_w,y_l)} \left[\log \sigma \left(\beta \log \frac{\pi_{\theta}(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_{\theta}(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right] \quad (6)$$

Here, π_{ref} denotes the frozen reference model, π_{θ} is the DPO model being trained, and β controls the deviation from the reference policy. The DPO model is fine-tuned using preference pairs consisting of chosen and rejected responses.

4 Experimental Setup

4.1 Base Model Architecture

- **Base Model:** [Qwen3.5-4B](#)
- **Parameters:** 4 Billion

4.2 Optimization Details

The optimization hyperparameters for both phases were carefully tuned to prevent catastrophic forgetting. We also used statscore to diagnose the catastrophic forgetting.

- **Context Length:** We set the maximum context length to 2048 for SFT and 1024 for DPO.
- **LoRA Configuration:** Rank (r) = 32 for SFT and 16 for DPO, Alpha (α) = 64 for SFT, 32 for DPO, Dropout = 0.05.
- **Batch Size:** [e.g., Global batch size of 16] (per device batch size of 2 with 8 gradient accumulation steps).
- **Learning Rate:** 2e-4 for SFT, 5e-5 for DPO (Cosine scheduler with linear warmup).
- **Epochs:** 1 for SFT, 1 for DPO.
- **Optimizer:** AdamW ($\beta = 0.1$, weight decay=0.01).
- **Compute Budget:** 1x NVIDIA RTX 5000 Ada Generation 32GB GPU for approximately 13 hours total.
- **Random Seed:** 42.

Mixed-precision training (bfloat16) was utilized to accelerate training and reduce VRAM usage.

4.3 Inference and Decoding Strategy

During final evaluation on the LM Playschool framework, we employed the following deterministic decoding parameters to ensure reproducible generation:

- **Temperature:** 0.0
- **Max New Tokens:** 300

Model	Clem	Δ	Stat	Δ
<i>Qwen3.5-4B</i>				
Base Pre-trained	37.33	0.00	52.07	0.00
LoRA SFT	38.03	+0.70	48.07	-4.00
LoRA DPO (Final)	38.23	+0.90	53.56	+1.49
<i>Qwen2.5-7B-Instruct</i>				
Base Pre-trained	26.52	0.00	48.35	0.00
LoRA SFT	34.23	+7.52	49.67	+1.32

Table 1: Validation performance across training stages for Qwen3.5 and Qwen2.5 architectures.

5 Results and Evaluation

We evaluated our models strictly according to the LM Playschool Challenge guidelines, relying on the *clemscore* and *statscore* as our primary metrics for conversational capability and task adherence. Briefly, the **clemscore** measures the model’s interactive conversational capabilities and its ability to follow fine-grained rules within simulated, multi-turn dialogue games (derived from the clem-bench framework). In contrast, the **statscore** acts as a baseline for general knowledge and reasoning by quantifying the model’s aggregate performance across a suite of standard and static benchmarks. All results are evaluated only in a single run.

As shown in Table 1, the SFT phase successfully aligned the base model to follow instructions, achieving a solid baseline clemscore of 38.03. The DPO phase successfully refined the model’s outputs by penalizing undesirable generation patterns, yielding an improved final score of 38.23.

6 Reproducibility Information

To ensure full transparency and allow reviewers to reproduce our submission, we have open-sourced all necessary models. The models are hosted on the Hugging Face Hub.

- **Final DPO Checkpoint (Adapters):** <https://huggingface.co/harshavaishnav/DPO>
- **SFT Checkpoint / Full Submission:** https://huggingface.co/harshavaishnav/LMPlayschool_Submission

The exact final merged models that corresponds to the LMPlayschool_Submission repository are

linked above. We have done inference on the final adapters but we are instructed to submit the merged models (base model + LoRA adapters). We have changed the partitioning mechanism from auto to cuda to do inference on a single GPU. Except this, we did not deviate from the public evaluation setup provided by the organizers.

7 Conclusion and Future Work

This technical report details our system development for the LM Playschool Workshop Shared Task at EMNLP 2026. Through rigorous data parsing and a systematic exploration of post-training paradigms across Qwen models, we demonstrated that direct preference tuning provides a powerful mechanism for stabilizing agentic behavior in multi-turn games. The stability of the statscore indicates that our approach effectively prevents catastrophic forgetting of the model. Our best run yielded a top leaderboard score of 38.23. Future work will investigate online RL loops (such as PPO or ReST) to integrate real-time game state engine rewards directly into the token optimization process.

Limitations

Despite the improvements, the model still exhibits limitations common to models of this scale, including occasional hallucinations in multi-turn reasoning tasks. The chosen rank ($r = 16$) for LoRA may also bottleneck the model’s capacity to internalize entirely new factual knowledge, restricting it primarily to style and format alignment.

References

- [1] K. Chalamalasetti, J. Götze, S. Hakimov, et al. 2023. **clembench: Using Game Play to Evaluate Chat-Optimized Language Models as Conversational Agents**. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.
- [2] Horst, Nicola, et al. 2025. Playpen: An Environment for Exploring Learning Through Conversational Interaction. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.
- [3] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen. 2021. **LoRA: Low-Rank Adaptation of Large Language Models**. *arXiv preprint arXiv:2106.09685*.
- [4] L. Ouyang, J. Wu, X. Jiang, et al. 2022. **Training language models to follow instructions with human feedback**. *Advances in Neural Information Processing Systems*, 35:27730–27744.

- [5] R. Rafailov, A. Sharma, E. Mitchell, C. D. Manning, S. Ermon, and C. Finn. 2023. [Direct Preference Optimization: Your Language Model is Secretly a Reward Model](#). *Advances in Neural Information Processing Systems*, 36.
- [6] A. Yang, A. Li, B. Yang, et al. 2025. [Qwen3 Technical Report](#). *arXiv preprint arXiv:2505.09388*.

Adapting SCoRe to Single-Player Dialogue Games: Initial Experiments on PLAYPEN

Neda Foroutan^{1,2*}, Valhari Meshram^{1*}

¹Potsdam Universität

²Technische Universität Berlin

{neda.foroutan, valhari.meshram}@uni-potsdam.de

*These authors contributed equally as first authors.

Abstract

Large Language Models (LLMs) are increasingly deployed as autonomous agents that solve tasks through multi-step interactions. However, most post-training methods are designed for single-turn reasoning or instruction-following tasks and provide limited supervision for long-horizon conversational decision making. In this work, we investigate whether Self-Correction via Reinforcement Learning (SCoRe), originally proposed for single-step mathematical reasoning and code generation, can improve agentic performance in multi-turn dialogue games. We adapt SCoRe to the PLAYPEN benchmark by fine-tuning Qwen3.5-9B on five single-player dialogue games using interaction-level rewards and iterative self-correction. Experimental results show that the resulting model substantially improves performance on the trained single-player games and further generalizes to unseen multi-player dialogue games, increasing the overall ClemScore from 41.18 to 43.31. Although a slight decrease is observed on the aggregate StatScore, the model largely retains its general language understanding and reasoning capabilities. These findings demonstrate that self-correction reinforcement learning is an effective approach for improving long-horizon conversational agent behavior beyond single-step reasoning tasks.

1 Introduction

LLMs have achieved remarkable performance across a wide range of natural language processing tasks, including question answering (Lucas et al., 2024; Borroto Santana et al., 2025), code generation (Dolcetti et al., 2026; Dong et al., 2024), and instruction following (Dong et al., 2025; Wu et al., 2024). Most existing LLMs are trained to generate a single response given an input prompt (Marvin et al., 2024), making them highly effective for both traditional single-turn applications (Nadi et al., 2024) and instruction-following (Heo et al.,

2025) tasks. More recently, LLMs have also been deployed as autonomous agents that interact with external environments by planning, reasoning, and taking actions over multiple steps to accomplish complex tasks (Kannan et al., 2024; Janakiraman, 2026).

Unlike single-turn settings, agentic applications require LLMs to maintain context, adapt to feedback, and make decisions throughout an interaction (Ouyang et al., 2026). However, most foundation models are pre-trained on static text corpora and instruction-tuned on single prompt-response pairs, providing limited supervision for multi-step conversational decision making. Reinforcement learning has therefore become a common approach for improving agentic behavior by optimizing models based on interaction outcomes rather than individual responses (Horst et al., 2025; Zhou and Zanette, 2024).

The LM Playschool Challenge¹ investigates whether training on multi-step conversational interactions, referred to as *dialogue games*, can improve the agentic capabilities of LLMs. In these games, an LLM interacts with a game environment over multiple turns, where each action influences subsequent observations and the final task outcome. This setting provides a realistic benchmark for evaluating planning, reasoning, and decision-making in interactive environments beyond conventional static benchmarks.

In this work, we, as the Dialogue Architects team, adapt Self-Correction via Reinforcement Learning (SCoRe) (Kumar et al., 2025) to multi-turn dialogue games by fine-tuning Qwen3.5-9B on five single-player games from the PLAYPEN benchmark. Unlike the original SCoRe framework, which focuses on single-step reasoning and code generation, our approach applies iterative self-correction to long-horizon conversational interac-

¹<https://lm-playschool.github.io/challenge/>

tions. Experimental results show that SCoRe improves performance on both trained single-player games and unseen multi-player games, increasing the overall ClemScore (+2.13), while maintaining competitive performance on general language understanding and reasoning benchmarks. These findings demonstrate that self-correction reinforcement learning is an effective approach for improving long-horizon conversational agent behavior.

2 Related Work

The PLAYPEN framework (Horst et al., 2025) introduces a benchmark for training and evaluating LLMs as conversational agents in multi-turn dialogue games. Unlike traditional instruction-following benchmarks, PLAYPEN evaluates models on long-horizon interactions where each action influences future observations and the final task outcome. Performance is measured using *ClemScore* for interactive gameplay and *StatScore* for general language understanding and reasoning. To improve agentic performance, the PLAYPEN authors investigated Supervised Fine-Tuning (SFT), Direct Preference Optimization (DPO), and Group Relative Policy Optimization (GRPO). While SFT and DPO improved performance on games seen during training, they generalized poorly to unseen games and often reduced StatScore. GRPO achieved the best trade-off by improving interactive gameplay while largely preserving general language capabilities.

Self-correction has recently emerged as an effective paradigm for reinforcement learning across a variety of sequential decision-making tasks, including autonomous driving (Guo et al., 2026), image captioning (Zhang et al., 2025), mathematical reasoning (Xiong et al., 2025), and search agents (Li et al., 2026). These methods enable models to iteratively revise intermediate decisions instead of relying solely on final outcome rewards.

Another self-correction approach is SCoRe (Self-Correction via Reinforcement Learning) (Kumar et al., 2025), which introduces a multi-turn reinforcement learning framework that explicitly trains language models to improve their own responses through iterative self-correction. Rather than relying on supervised correction traces or external teacher models, SCoRe performs online reinforcement learning on self-generated trajectories and employs reward shaping to encourage successful self-correction while preventing behavior collapse. The method achieved state-of-the-art performance

on the MATH and HumanEval benchmarks, substantially improving mathematical reasoning and code generation through a single self-correction step.

Our work extends SCoRe to a substantially different setting. While the original framework was developed for single-step reasoning and coding tasks, we apply it to multi-turn dialogue games in the PLAYPEN benchmark, where the model must maintain conversational context, adapt its decisions over multiple interaction steps, and optimize for episode-level success. Our results show that SCoRe improves interactive dialogue-game performance and is effective beyond single-step reasoning tasks and can enhance long-horizon conversational agent behavior.

3 Methodology

3.1 SCoRe Learning Algorithm

SCoRe (Kumar et al., 2025) is a two-stage reinforcement learning framework designed to improve a language model’s ability to self-correct. During training, the model first generates an initial attempt (y_1) and then generates a second attempt (y_2) conditioned on the original task, the complete trajectory of the first attempt, and a self-correction prompt. The objective of SCoRe is to encourage the second attempt to correct mistakes made in the first attempt while maintaining stable optimization throughout training. Figure 1 provides an overview of the two-stage SCoRe training framework.

Stage I – Constrained Initialization using KL-Divergence: The policy model is initialized using reinforcement learning to improve the accuracy of the second attempt while keeping the first attempt close to the original base model. A KL-divergence regularization constrains the first attempt to remain similar to the base model, allowing the second attempt to improve independently. This stage reduces the tendency to couple the two attempts and provides a stable initialization for subsequent self-correction training.

Stage II – Reward-Shaped Multi-Turn RL: Starting from the Stage I policy, the model is further optimized using a reward-shaping objective that jointly encourages the correctness of the second attempt and meaningful improvement over the first attempt. A reward bonus is assigned when the second attempt successfully corrects an incorrect first attempt, while a large negative penalty is applied when a correct first attempt is changed into an

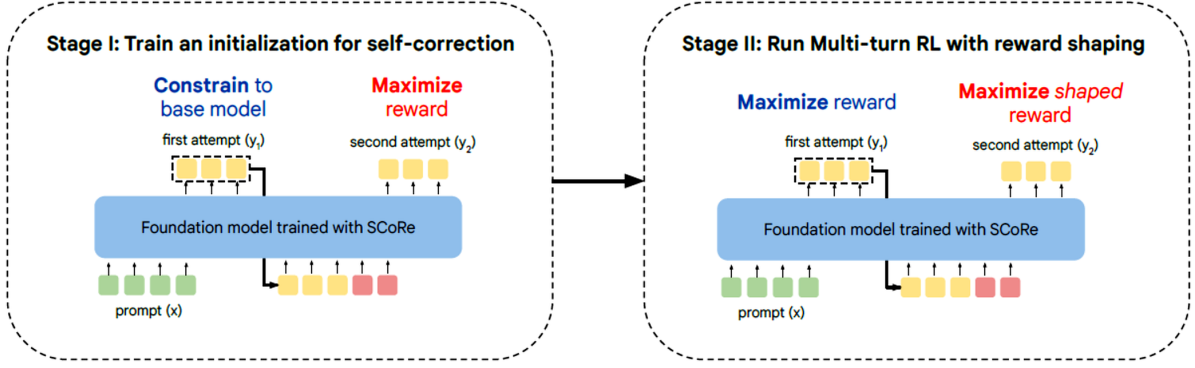


Figure 1: Overview of the two-stage SCoRe training framework: Source: (Kumar et al., 2025). Stage I trains a stable initialization by constraining the first attempt to remain close to the base model while optimizing the second attempt. Stage II jointly optimizes both attempts using reward-shaped reinforcement learning to encourage successful self-correction.

incorrect one. This reward design explicitly encourages effective self-correction while discouraging unnecessary revisions.

3.2 Implementation

We used Qwen3.5-9B² as the base policy model and fine-tuned it using the two-stage SCoRe framework (Kumar et al., 2025) on the single-player dialogue games from the PLAYPEN benchmark³. We employed 4-bit QLoRA for parameter-efficient fine-tuning and merged the learned LoRA weights into the base model after training. Unlike the original SCoRe work, which focused on single-step reasoning and code generation, we adapt SCoRe to multi-turn dialogue games, where each self-correction attempt corresponds to an entire gameplay episode rather than a single generated response. The training data consists of five single-player PLAYPEN games: *adventuregame*, *textmap-world*, *textmapworld_graphreasoning*, *textmap-world_specificroom*, and *wordle*. The total number of training instances is 161, with the breakdown by game reported in Table 4 in Appendix A.

For each training episode, the model first generates an initial attempt (y_1) by completing an entire dialogue-game episode. The complete interaction trajectory together with the episode outcome (successful, unsuccessful, or aborted) is then provided through a self-correction prompt, enabling the model to reflect on its previous decisions before generating a second attempt (y_2). Figure 2 shows the self-correction prompt template used to transition from the first attempt to the second attempt.

²<https://huggingface.co/Qwen/Qwen3.5-9B>

³<https://huggingface.co/datasets/colab-potsdam/playpen-data>

Episode-level rewards were computed using the success signals returned by each CLEMcore game evaluator. A reward of +1 was assigned when the policy model successfully completed the game, 0 when it failed to solve the game, and -1 when the episode was aborted due to a rule violation or invalid interaction.

To facilitate reproducibility, the training scripts are publicly available at <https://github.com/NedaForoutan/SCoRe-to-Single-Player-Dialogue-Game>, and the fine-tuned model is released at <https://huggingface.co/Valhari14/score-qwen3.5-9b-playpen>.

3.2.1 Experimental Setup

Training on the five single-player dialogue games took approximately 48 hours using three NVIDIA A30 GPUs on a single compute node. The GPUs were used to parallelize on-policy rollout generation, as SCoRe requires two complete game playthroughs per training episode, with an initial attempt followed by a self-correction attempt, resulting in approximately twice the generation cost of standard single-attempt reinforcement learning fine-tuning. To enable memory-efficient training, the model was loaded using 4-bit QLoRA through the Hugging Face Transformers, PEFT, and bitsandbytes libraries. Unless otherwise specified, we followed the training configuration of the original SCoRe implementation.

The Stage II reward-shaping coefficient was set to $\alpha = 10.0$, while the KL-divergence regularization weights were $\beta_1 = 0.01$ for Stage II and $\beta_2 = 0.1$ for Stage I. The learning rate was 1×10^{-5} . During training, the sampling tempera-

```

if y1.success : "You SUCCEEDED in your previous attempt."
if y1.aborted : "Your previous attempt was ABORTED due to a rule violation."
if y1.lost : "Your previous attempt was UNSUCCESSFUL."

"Below is a summary of what you did last time. Carefully review it, identify what
went wrong or could be improved, and play better this time. A fresh attempt at the
same task begins now."

```

Figure 2: Self-correction prompt template used during SCoRe training. After completing the first attempt, the model receives feedback indicating whether the episode was successful, unsuccessful, or aborted, together with an instruction to review its previous decisions before generating a second attempt.

Model	ClemScore	StatScore
Qwen3.5-9B	41.18	54.35
SCoRe_Qwen3.5-9B	43.31 ■	52.71 ■

Table 1: Performance comparison between the base **Qwen3.5-9B** model and the **SCoRe**-fine-tuned variant on the PLAYPEN benchmark. ClemScore evaluates performance on all interactive dialogue games, while StatScore is the average across BBH, CLadder, EQBench, IFEval, and MMLU-Pro.

ture was set to 0.8, the maximum number of generated tokens per attempt was 256, and Stage I and Stage II were trained for 3 and 5 epochs, respectively.

4 Results

Table 1 compares the overall performance of the base Qwen3.5-9B model and the SCoRe fine-tuned model, SCoRe_Qwen3.5-9B, on the PLAYPEN benchmark. SCoRe_Qwen3.5-9B improves the overall ClemScore from 41.18 to 43.31 (+2.13), indicating better performance on interactive dialogue games. The aggregate StatScore decreases only slightly from 54.35 to 52.71, suggesting that the reinforcement learning fine-tuning largely preserves the model’s general language understanding and reasoning capabilities despite being optimized for interactive dialogue tasks.

Performance on Single-Player Dialogue Games:

Table 2 presents the performance of the SCoRe fine-tuned model on the five single-player PLAYPEN dialogue games. Compared with the baseline model, SCoRe substantially improves the overall ClemScore from 39.38 to 47.31 (+7.93). Although the completion rate (% Played) remains unchanged across all games, the quality score (QS) consistently improves on every game except *Wordle*, where both models aborted or timed out on several instances. The largest improvement is observed

on *AdventureGame*, where the quality score increases from 11.11 to 33.33, followed by *TextMap-World_SpecificRoom* (39.66 to 56.45) and *TextMap-World* (68.01 to 73.07). These results indicate that SCoRe primarily improves the quality of decision making within successful gameplay episodes on the single-player dialogue games used for training, leading to a substantial improvement in overall interactive performance.

Generalization to Multi-Player Dialogue Games:

To evaluate whether the learned self-correction behaviour generalizes beyond the training games, we evaluated the model on unseen multi-player PLAYPEN dialogue games. As shown in Table 3, SCoRe improves the overall ClemScore from 41.18 to 43.31 (+2.13) and increases the average completion rate from 70.21% to 74.37%. The largest improvement is observed on *Wordle_withClue*, where the completion rate doubles from 33.33% to 66.67%, although the average quality score decreases because the additional successful-played episodes receive lower-quality solutions. SCoRe also improves the quality score on *Taboo* from 41.67 to 50.00, indicating that the learned self-correction strategy transfers effectively to this collaborative word-based game.

Performance decreases slightly on *Private-Shared*, where the quality score drops by 4.26 points, and the overall average quality score decreases modestly from 66.20 to 64.44. Moreover, the baseline and SCoRe models achieve identical performance on *ImageGame*, *MatchIt_ASCII*, *ReferenceGame*, and *Wordle_withCritic*. And both models consistently aborted or timed out on *Wordle*, *GuessWhat*, and *Codenames*.

These results suggest that training with SCoRe on single-player dialogue games improves the model’s ability to generalize to unseen multi-player dialogue games, particularly by increasing task

Model	Overall	All,Average		AdventureGame		TextMapWorld		TMW-GR		TMW-SR	
	ClemScore	%	QS	%	QS	%	QS	%	QS	%	QS
Qwen3.5-9B	39.38	72	54.7	60	11.11	100	68.01	100	39.66	100	100
SCoRe-Qwen3.5-9B	47.31 ^{+7.93}	72	65.71 ^{+11.01}	60	33.33 ^{+22.22}	100	73.07 ^{+5.06}	100	56.45 ^{+16.79}	100	100

Table 2: Generalization performance of the SCoRe-Qwen3.5-9B (SCoRe-Qwen) model on **unseen single-player PLAYPEN dialogue games** after training on five single-player dialogue games. “%” denotes the percentage of successfully played games, and QS denotes the average quality score for completed games. TMW-GR and TMW-SR refer to *TextMapWorld_GraphReasoning* and *TextMapWorld_SpecificRoom*, respectively.

completion rates. While improvements in interaction quality are not uniform across all game types, the gains in overall ClemScore demonstrate that self-correction reinforcement learning transfers beyond the training games and improves long-horizon conversational decision making.

Overall, the experimental results demonstrate that SCoRe is an effective reinforcement learning strategy for improving LLMs in interactive dialogue-game environments. Fine-tuning on a small set of single-player dialogue games substantially improves in-domain performance while also transferring to unseen multi-player games. Importantly, these gains are achieved with only a modest reduction in StatScore, indicating that the improvements in agentic behaviour come without significant degradation of the model’s general language understanding and reasoning capabilities.

5 Conclusion

This work adapts Self-Correction via Reinforcement Learning (SCoRe) to multi-turn dialogue games in the PLAYPEN benchmark. Unlike the original SCoRe framework, which was developed for single-step mathematical reasoning and code generation, our approach applies iterative self-correction to long-horizon conversational interactions. The model is trained on five single-player dialogue games, where each training episode consists of an initial gameplay attempt followed by a self-corrected replay of the same game instance. Experimental results show that SCoRe substantially improves performance on the trained single-player dialogue games and also generalizes to unseen multi-player games, demonstrating that self-correction learned from single-player interactions transfers to more complex collaborative settings. Overall, our findings demonstrate that self-correction reinforcement learning is a promising training paradigm for enhancing the agentic capabilities of large language models in long-horizon conversational en-

vironments. Beyond improving performance on individual dialogue games, the proposed adaptation shows that SCoRe can be successfully extended from single-step reasoning tasks to sequential decision-making settings involving multi-turn interactions.

6 Limitations

This work has several limitations. First, due to the lack of time for training SCoRe, our experiments are limited to the five single-player dialogue games provided in the PLAYPEN benchmark. Extending the approach to multi-player dialogue games requires substantially longer training times, and is therefore left for future work.

Second, we evaluate SCoRe using a single base model, Qwen3.5-9B. Because of the limited time and computational resources required for reinforcement learning fine-tuning, we were unable to investigate the effectiveness of SCoRe on additional LLMs. Evaluating the proposed approach across a broader range of open-source models with different architectures and scales would provide a better understanding of its generalizability.

References

- Manuel Alejandro Borroto Santana, Katie Gallagher, Antonio Ielo, Irfan Kareem, Francesco Ricca, and Alessandra Russo. 2025. [Question answering with llms and learning from answer sets](#). *Theory and Practice of Logic Programming*, page 1–25.
- Greta Dolcetti, Vincenzo Arceri, Eleonora Iotti, Sergio Maffei, Agostino Cortesi, and Enea Zaffanella. 2026. [Helping llms improve code generation using feedback from testing and static analysis](#). *Discover Artificial Intelligence*, 6(1):314.
- Guanting Dong, Keming Lu, Chengpeng Li, Tingyu Xia, Bowen Yu, Chang Zhou, and Jingren Zhou. 2025. [Self-play with execution feedback: Improving instruction-following capabilities of large language models](#). In *International Conference on Learning Representations*, volume 2025, pages 39286–39313.

Model	Overall		All Avg.		PrivateShared		Taboo		Wordle+Clue	
	ClemScore	%	QS		%	QS	%	QS	%	QS
Qwen3.5-9B	41.18	70.21	66.20		80.00	26.52	100.00	41.67	33.33	100.00
SCoRe-Qwen3.5-9B	43.31 ^{+2.13}	74.37 ^{+4.16}	64.44 ^{-1.76}		80.00	22.26 ^{-4.26}	100.00	50.00 ^{+8.33}	66.67 ^{+33.34}	66.66 ^{-33.34}

Table 3: Generalization performance of the SCoRe-Qwen3.5-9B (SCoRe-Qwen) model on **unseen multi-player PLAYPEN dialogue games** after training on five single-player dialogue games. “%” denotes the percentage of successfully completed games, and QS denotes the average quality score. For brevity, only games exhibiting differences between the baseline Qwen3.5-9B and SCoRe-Qwen are reported.

- Yihong Dong, Xue Jiang, Zhi Jin, and Ge Li. 2024. [Self-collaboration code generation via chatgpt](#). *ACM Trans. Softw. Eng. Methodol.*, 33(7).
- Yihong Guo, Dongqiangzi Ye, Sijia Chen, Anqi Liu, and Xianming Liu. 2026. [Correctionplanner: Self-correction planner with reinforcement learning in autonomous driving](#). *Preprint*, arXiv:2603.15771.
- Juyeon Heo, Christina Heinze-Deml, Oussama Elachqar, Kwan Ho Ryan Chan, Shirley Ren, Andrew Miller, Udhyakumar Nallasamy, and Jaya Narain. 2025. [Do llms ``know''internally when they follow instructions?](#) In *International Conference on Learning Representations*, volume 2025, pages 81339–81357.
- Nicola Horst, Davide Mazzaccara, Antonia Schmidt, Michael Sullivan, Filippo Momentè, Luca Franceschetti, Philipp Sadler, Sherzod Hakimov, Alberto Testoni, Raffaella Bernardi, Raquel Fernández, Alexander Koller, Oliver Lemon, David Schlangen, Mario Giulianelli, and Alessandro Suglia. 2025. [Playpen: An environment for exploring learning from dialogue game feedback](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29854–29891, Suzhou, China. Association for Computational Linguistics.
- Anantharaman Janakiraman. 2026. [From generative intelligence to agentic autonomy: Leveraging large language models for multi-agent reasoning, planning, and execution](#). *Journal of Integrated Science, AI and Engineering*, 2(1).
- Shyam Sundar Kannan, Vishnunandan L. N. Venkatesh, and Byung-Cheol Min. 2024. [Smart-llm: Smart multi-agent robot task planning using large language models](#). In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 12140–12147.
- Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, JD Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. 2025. [Training language models to self-correct via reinforcement learning](#). In *International Conference on Learning Representations*, volume 2025, pages 54523–54549.
- Shiyu Li, Yang Tang, Yifan Wang, Peiming Li, and Xi Chen. 2026. [Reseek: A self-correcting framework for search agents with instructive rewards](#). *Preprint*, arXiv:2510.00568.
- Mary M Lucas, Justin Yang, Jon K Pomeroy, and Christopher C Yang. 2024. [Reasoning with large language models for medical question answering](#). *Journal of the American Medical Informatics Association*, 31(9):1964–1975.
- Ggaliwango Marvin, Nakayiza Hellen, Daudi Jjingo, and Joyce Nakatumba-Nabende. 2024. Prompt engineering in large language models. In *Data Intelligence and Cognitive Informatics*, pages 387–402, Singapore. Springer Nature Singapore.
- Farhad Nadi, Hadi Naghavipour, Tahir Mehmood, Al-liesya Binti Azman, Jeetha A/P Nagantheran, Kezia Sim Kui Ting, Nor Muhammad Ilman Bin Nor Adnan, Roshene A/P Sivarajan, Suita A/P Veerah, and Romi Fadillah Rahmat. 2024. Sentiment analysis using large language models: A case study of gpt-3.5. In *Data Science and Emerging Technologies*, pages 161–168, Singapore. Springer Nature Singapore.
- Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. 2026. [Reasoning-bank: Scaling agent self-evolving with reasoning memory](#). In *International Conference on Learning Representations*, volume 2026, pages 94327–94354.
- Tianhao Wu, Janice Lan, Weizhe Yuan, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. 2024. [Thinking llms: General instruction following with thought generation](#). *Preprint*, arXiv:2410.10630.
- Wei Xiong, Hanning Zhang, Chenlu Ye, Lichang Chen, Nan Jiang, and Tong Zhang. 2025. [Self-rewarding correction for mathematical reasoning](#). *Preprint*, arXiv:2502.19613.
- Lin Zhang, Xianfang Zeng, Kangcong Li, Gang Yu, and Tao Chen. 2025. Sc-captioner: Improving image captioning with self-correction by reinforcement learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 23145–23155.

Yifei Zhou and Andrea Zanette. 2024. Archer: training language model agents via hierarchical multi-turn rl. In *Proceedings of the 41st International Conference on Machine Learning, ICML’24*. JMLR.org.

A Training Instances

Table 4 reports the number of training instances for each of the single-player games from the PLAYPEN benchmark. The training set comprises 161 instances in total, distributed across AdventureGame, TextMapWorld, TextMapWorld_GraphReasoning, TextMapWorld_SpecificRoom, and Wordle. These instances were used to fine-tune Qwen3.5-9B with Self-Correction via Reinforcement Learning (SCoRe) for multi-turn dialogue-game interactions.

Game	# of Instances
AdventureGame	35
TextMapWorld	45
TextMapWorld_GraphReasoning	27
TextMapWorld_SpecificRoom	27
Wordle	27
Total	161

Table 4: Number of training instances for each single-player PLAYPEN game used for fine-tuning of SCoRe_Qwen3.5-9B.

B Dialogue Game Transcripts

To qualitatively illustrate the effect of SCoRe, Figure 3 presents a representative self-correction episode from the *textmapworld* game. In the first attempt (y_1), the model terminates the game after revisiting previously explored rooms, failing to discover all reachable locations. After receiving the self-correction prompt shown in Figure 2, the model generates a second attempt (y_2), avoids the earlier navigation error, successfully explores the remaining room, and completes the task. This example demonstrates how SCoRe enables the model to reflect on its previous attempt and improve its decision-making over multiple interaction steps.

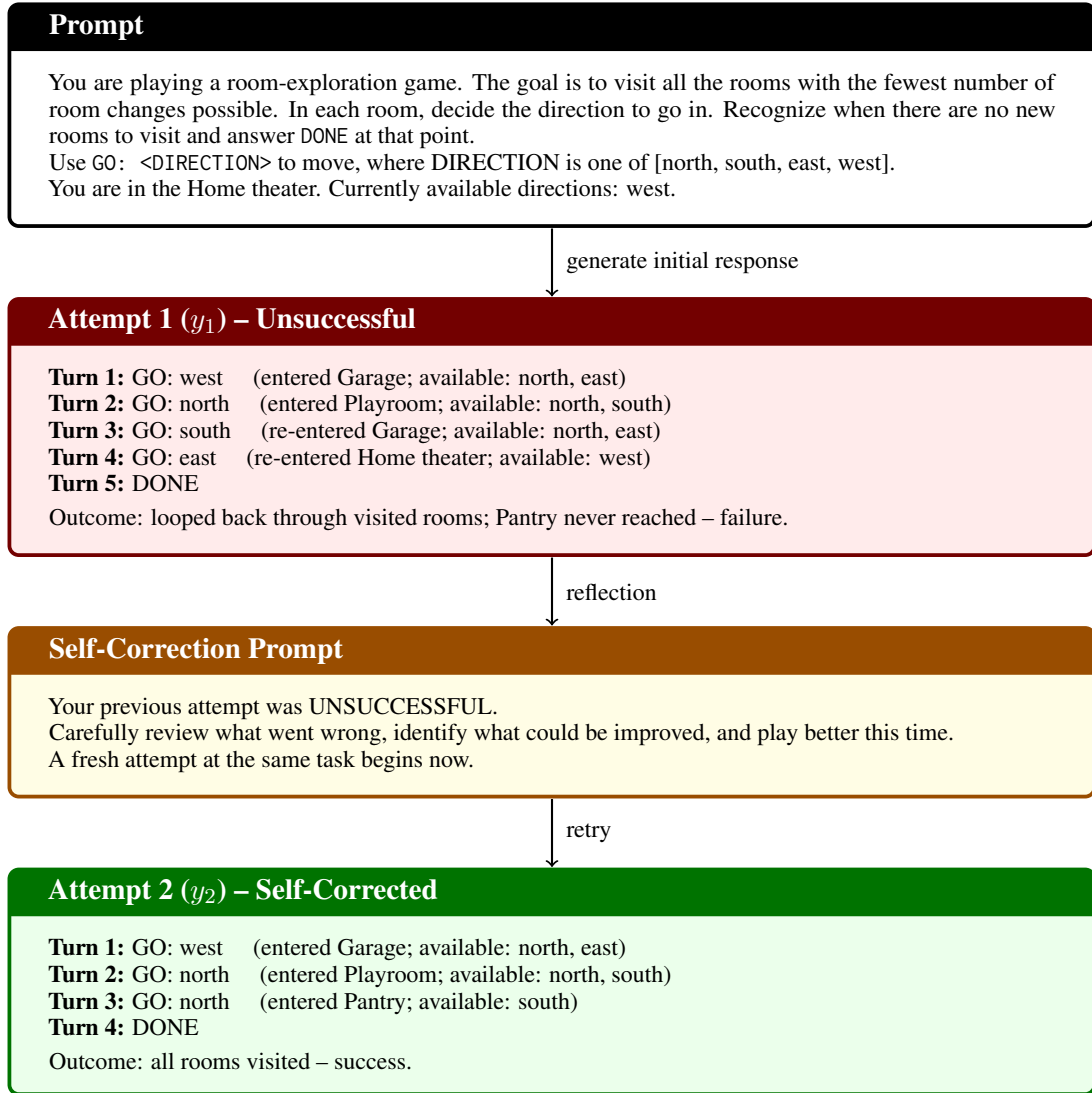


Figure 3: SCoRe self-correction example on textmapworld. The model first generates an unsuccessful attempt (y_1), receives the self-correction prompt, and then generates a corrected attempt (y_2) that solves the task.

First Make It Playable, Then Make It Good: Staged Interaction Learning for Small Dialogue-Game Agents

Syed Mahbubul Huq¹ & Pranava Madhyastha^{1,2}

¹ City, University of London

² The Alan Turing Institute

{syed-mahbubul.huq,2,pranava.madhyastha}@city.ac.uk

Abstract

We present **Qwen-GuidePlay-2B**, a 2B-parameter language model for dialogue-game interaction. We fine-tune Qwen3.5-2B using three steps: a) SFT on only successful game trajectories from Playpen, b) weighted turn-level SFT, and c) teacher-guided SFT. The teacher model (which is a larger model) is only used to fix formatting and evaluate examples, but does not create new gold actions. Our final model scores 57.12 clemscore and 42.68 statscore on the public Playpen validation. In the officially released challenge results, our model obtains the second-highest Playpen clemscore delta among submitted systems (which is $\approx +36$ over its base model)¹. Our findings suggest that imitating full trajectories helps with playability, while turn-level and teacher-guided training usually improve decision-making and increase the overall score. Alternative procedurally heavy approaches like replay-repair and hard-example mining did not help, which suggests that small models are performant simply by using careful curation strategies rather than aggressive changes. We make available both the model and the code for reproducibility.²

1 Introduction

Large language models (LLMs) have gained significant popularity as agents in multi-step interactive environments, even though they are traditionally being trained on isolated instruction-response pairs. Dialogue games, however, demand a distinct form of competence. An effective agent must first observe a state, then produce a valid action, receive subsequent feedback, and interact according to a defined protocol (Chalamalasetti et al., 2023; Beyer et al., 2024; Schlangen et al., 2025). In this context,

language serves as both a representational medium and the very mechanism through which the agent acts. As a result, a generated string may constitute a legal move, an illegal move, a premature termination, or a failure to adhere to the rules of the game.

Within such setups, a model may fail in various ways, such as by providing unnecessary explanations, employing incorrect action formats, failing to halt when required, executing multiple actions simultaneously, or breaking the game’s established role structure. To address these challenges within the Playpen framework, our approach focuses on equipping the language model with a robust understanding of interaction dynamics and legal rules, rather than merely encouraging the memorisation of specific game states. In the context of this challenge, playability measures the extent to which the model has internalised interactional rules well enough to complete episodes without aborting. Quality, conversely, evaluates how well the model performs the task once valid participation is established.

Our aim for this submission is to develop a model that explicitly captures this distinction. In our setup, a successful Playpen transcript comprises both interactional signals, which demonstrate adherence to participation rules, and decision signals, which guide the appropriate action to take given a specific dialogue or game state. To leverage these effectively, we employ a staged and incremental training strategy. First, we fine-tune the model on complete, successful trajectories, truncated to a maximum of 1,024 tokens, enabling it to learn the global rhythm of valid participation. Second, we factorise these successful trajectories into local state-action examples (mapping dialogue/game history to next assistant action), thereby aligning the training objective more closely with the inference-time decision problem. Third, we utilise a more capable model as a constrained teacher, and this

¹<https://github.com/lm-playpen/lm-playschool-2026-final-results>

²Model: <https://huggingface.co/syedhuq/qwen-guideplay-2b>. Code: <https://github.com/SyedHuq28/qwen-guideplay-2b>.

teacher evaluates selected examples and generates invalid 'near-misses' for repair, without inventing new gold-standard actions.

Our architectural design is grounded in recent literature on successful and guided learning. Success-filtered training leverages high-quality trajectories as supervision (Zelikman et al., 2022; Gulcehre et al., 2023, *inter alia*), whilst value-weighted policy learning applies stronger updates to higher-value actions (Peng et al., 2019). Furthermore, cognitive accounts of learning emphasise guided participation, scaffolding, and the gradual internalisation of rule-governed practices (Vygotsky, 1978; Bruner, 1985). Recent developments in text-based game agents also demonstrate that complex action spaces can be rendered more tractable by representing states and actions as text, decomposing extensive action choices into smaller, manageable decisions (Xu et al., 2025). We synthesise these principles to formulate a robust training recipe for Playpen dialogue games.

Our primary contributions are: (a) We propose a three-stage interaction-learning recipe for a 2B-parameter Playpen agent; (b) We provide an analysis demonstrating that successful trajectories and factorised state-action examples serve complementary roles in improving both playability and decision quality; (c) We introduce a teacher-guided augmentation stage that evaluates and repairs examples without generating new gold actions; and (d) We finally present compact ablation studies revealing that our lightweight, teacher-guided training recipe performs highly effectively, precluding the need for computationally heavier replay-repair mechanisms, outcome weighting, or hard-example mining.

2 Method

2.1 Stage 1: Success SFT for valid participation

We fine-tune Qwen3.5-2B (Qwen Team, 2026) using only successful Playpen trajectories. We export the Playpen interactions train split to JSONL and filter for episodes where Outcome equals "success". We focus on successful episodes because this stage aims to teach the model what a valid interaction looks like. The model, before doing anything, requires understanding and learning the structure of a successful dialogue-game episode before improving its decision-making. This is aligned with the Playpen framing of Dialogue Games (Horst et al., 2025). It is also related to

success-filtered post-training methods, which train on model outputs or traces that satisfy an external success criterion (Zelikman et al., 2022).

We format each successful episode with the Qwen chat template and train the model using standard causal language model loss, truncating sequences longer than 1,024 tokens. We chose this limit to balance coverage and computing cost. In our dataset, the median episode length is 462 tokens. Setting the limit at 1,024 tokens keeps about 82% of episodes complete. We decided against a higher limit because it would include more episodes but would also need much more memory. We use Qwen3.5 in its default non-thinking mode throughout training.

2.2 Stage 2: Value-weighted turn-level SFT

Stage 2 converts successful trajectories into local state-action examples. For each assistant turn, we create:

prompt = dialogue/game history before the assistant action

completion = next assistant action.

A successful transcript becomes many small local decisions, and the model is trained on the inference-time problem directly (given the current interaction state, produce the next legal assistant action). Decision-level factorisation of complex action spaces has recently been studied and applied for LLM game agents (Xu et al., 2025); in our setting, the "moves" are themselves conversational contributions.

Each row receives a lightweight heuristic value computed from the completion text. Clean, concise, non-leaky completions receive higher weight:

$$\begin{aligned} v = & 0.70 \\ & + 0.15 \cdot \mathbb{I}[1 \leq \text{len}(c) \leq 400] \\ & + 0.05 \cdot \mathbb{I}[\text{no stop-token leakage}] \\ & + 0.05 \cdot \mathbb{I}[\text{no role-marker leakage}] \\ & + 0.05 \cdot \mathbb{I}[\text{len}(c) > 0]. \end{aligned}$$

The value is capped at 1.0. Because all source episodes are successful, most values concentrate around 1.00 for clean short completions and 0.85 for clean long completions. Weighting imitation by a scalar per-decision value follows reward- and advantage-weighted regression (Peters and Schaal, 2007; Wang et al., 2018; Peng et al., 2019), which LLM post-training instantiates as reward-filtered or

reward-ranked fine-tuning (Gulcehre et al., 2023; Dong et al., 2023). Unlike equilibrium-derived Q-values (Xu et al., 2025) or learned advantages, our values are deliberately cheap textual proxies computed from the completion alone, prioritising clean, concise, non-leaky local decisions under a small compute budget.

We train with completion-only weighted SFT. Let ℓ_i be the mean token loss over the completion of example i , and let w_i be its value. The batch loss is:

$$L = \frac{\sum_i w_i \ell_i}{\sum_i w_i}.$$

We also train a uniform ablation where all turn-level weights are set to 1.0. This tests whether the gain comes from factorisation alone or from value weighting.

2.3 Stage 3: Teacher-guided repair as constrained teacher repair

Stage 3 keeps the Stage 2 recipe but changes the data (we add a small amount of teacher-guided material while keeping Stage 2 rows dominant). A stronger LLM (Gemma-4-31B-it, Team et al., 2026) teacher judges selected Stage 2 rows and generates invalid outputs for repair training, loosely inspired by accounts of scaffolded learning with a more capable partner (Vygotsky, 1978; Bruner, 1985). Gemma-4-31B-it is used as the teacher, with temperature 0.1 and a maximum generation length of 700 tokens. Additional teacher-generation details are provided in Appendix B.

The teacher does not invent a new gold action, the target action always remains the original gold action from a successful Playpen trace, and invalid outputs appear only on the input side of repair examples.

For judged rows, the teacher receives the dialogue/game history before the action, the original gold assistant answer, the previous Stage 2 weight, and the strict Playpen output contract. It returns JSON scores such as `format_score`, `action_score`, `task_progress_score`, `done_score`, `overall_score`, a `keep` decision, a `short reason`, and `invalid bad_outputs`. The teacher prompt explicitly states that the clean target must remain the provided gold answer.

For kept rows, the teacher’s overall score s mildly adjusts the original value v :

$$v' = \text{clip}(v(0.75 + 0.50s), 0.40, 1.20).$$

The teacher can change how much an example counts, but it never changes what the model is trained to produce. Before sampling the 30,000 original Stage 2 rows, we exclude rows that were used for teacher judging. The 1,000 teacher-judged rows are then included in reweighted form, so their original versions do not also appear among the 30,000 sampled Stage 2 rows.

For repair examples, the teacher writes realistic near-misses of a gold action: added explanation, a missing GO: prefix, wrong casing, leaked role markers, stray punctuation, several actions at once, or a premature DONE. Each repair prompt shows the game history and one such invalid answer and asks the model to rewrite it correctly; the training target is always the original gold action. Producing small corruptions of a given answer is an easy task for an instruction-tuned model, so there is little room for it to invent content. As a check, we manually inspected 20% of the generated repairs and found no confabulated text.

The final Stage 3 mixture is shown in Table 1.

Component	Count	Share
Original Stage 2 weighted-turn rows	30,000	96.5%
Teacher-judged/reweighted rows	1,000	3.2%
Teacher-generated repair rows	100	0.3%
Total	31,100	100%

Table 1: Final Stage 3 training mixture.

The teacher ratio is intentionally small. Development runs showed that heavier replay or repair can improve Playpen-specific clemscore while reducing statscore, suggesting over-specialisation.

2.4 Model adaptation, prompting, and post-processing

All stages use Qwen3.5-2B with LoRA adapter training (Hu et al., 2022) followed by merging into the base model. We use a standard configuration (rank 16, $\alpha = 32$, i.e. $\alpha = 2r$, dropout 0.05) and adapt all linear layers (target_modules="all-linear"), following evidence that adapting all layers rather than attention projections alone is important for matching full fine-tuning quality (Dettmers et al., 2023). A single adapter is created in Stage 1 and used to initialise the later weighted-SFT runs. **Qwen-GuidePlay-2B** is released as a full merged Hugging Face model, not only a LoRA adapter.

Teacher-guided augmentation uses two prompt

types (Details in the Appendix). The judging prompt asks the teacher to return JSON-only scores, a keep/drop decision, a reason, and invalid outputs, while explicitly forbidding new gold answers. The repair prompt provides the game history and an invalid answer, and asks the model to output only the corrected valid answer. The teacher (Gemma-4-31B-it) is queried at near-deterministic temperature (0.1) for judging and is drawn from a different model family than the student. The full prompt templates are included in Appendix.

For Playpen evaluation, the model is registered with the local Hugging Face backend, the premade chat template, left padding, and `eos_to_cull=<|im_end|>`. After merging, `generation_config.json` is patched to `eos_token_id=[248046, 248044]` and `pad_token_id=248044`, so the model stops at Qwen’s chat end token as well as the base EOS.

3 Experimental Setup

3.1 Data

All training supervision derives from the `colab-potsdam/playpen-data` interactions train split. Validation data is used only for public evaluation and model selection, not for constructing training examples. No private evaluation data is used for training or model selection.

Training data comes in four forms: (1) successful full trajectories for Stage 1; (2) factorised turn-level state-action rows for Stage 2; (3) teacher-judged/reweighted rows for Stage 3; and (4) teacher-generated repair rows for Stage 3.

The training split contains 20,202 successful trajectories across 16 games, from which we obtain 105,972 factorised turn-level examples for Stage 2. The final Stage 3 mixture contains 31,100 rows: 30,000 original Stage 2 rows, 1,000 teacher-judged rows, and 100 repair rows. Before drawing the 30,000 Stage 2 rows, we exclude rows that were used as inputs to the teacher judge. Teacher-judged rows are filtered using an overall-score threshold of 0.70, after which 1,000 rows are sampled from the higher-scoring candidate pool. The 100 repair rows are sampled randomly from the available repair examples. The mixture builder uses master seed 28, with derived draw seeds 29, 30, and 31 for the Stage 2, judged, and repair samples, respectively. All reported results are from single training runs, and we do not average across multiple training seeds.

The dataset-building scripts reject paths

containing validation/test/eval markers such as validation, /val, test, playpen-eval, eval_results, leaderboard, or public.

3.2 Training hyperparameters and compute

Setting	Stage 1	Stage 2	Final
Initialisation	Base + new LoRA	Stage 1 adapter	Stage 1 adapter
Data	Success dialogues	~106k turn rows	31,100-row mix
Loss tokens	All	Completion-only	Completion-only
Example weights	None	Heuristic	Heur./teacher/repair
Learning rate	2e-4	5e-5	5e-5
Epochs	1.0	0.25	0.25
Effective batch	8	16	8

Table 2: Per-stage training configuration.

Stage 1 uses a learning rate of 2e-4, consistently for LoRA fine-tuning (Dettmers et al., 2023); Stage 2 and the final run reduce this to 5e-5 to limit drift when continuing an already-trained adapter. Fractional epochs are grounded in our own ablation: 0.50 epochs of Stage 2 training did not improve over 0.25 (Table 5), so all later runs use 0.25. All stages use the PyTorch AdamW optimiser with a cosine learning-rate schedule and a warmup ratio of 0.03. Training uses bfloat16 precision without weight quantisation. Stage 1, Stage 2, and the final run comprise 1,263, 1,656, and 972 optimisation steps, respectively. Training wall-clock times were approximately 8.8 h, 5.7 h, and 4.0 h, respectively.

Training used NVIDIA A100 80GB GPUs: Stage 1 and the final run used a single process with effective batch size 8, while Stage 2 used 2-GPU DDP with effective batch size 16.

All remaining heuristic constants, the value-formula terms (§2.2), the teacher-modulation coefficients and clip range, the repair weight of 0.75, and the judge-score threshold of 0.70, were set once by inspection under a fixed compute budget and were not tuned. Our ablations therefore test each mechanism as a whole rather than its constants.

3.3 Evaluation

All models were evaluated with Playpen 3.7.0, following the LM Playschool Challenge instructions for setting up the workspace and evaluating a model. We evaluate with the public Playpen validation pipeline using:

```
playpen eval <model> --suite all
```

The evaluation includes an interactive dialogue-game suite and a static suite. We report `clemscore` for interactive performance and `statscore` as a robustness check against over-specialisation. All evaluations use Playpen/clembench default decoding.

For runs where detailed breakdowns are available, clemscore decomposes as:

$$\text{clemscore} = \frac{\text{Avg. \% Played} \times \text{Avg. Quality}}{100}.$$

This separates two failure modes. % Played measures whether the model can validly participate without aborting. Quality measures how well the model performs once it is playing.

The official Qwen3.5-2B base scores are 13.05 clemscore and 44.02 statscore. Our local base run gives the 11.91, with 28.57 average % Played and 41.70 average Quality. We use the official base for claims regarding improvement in performance and the local base for playability/quality analysis.

4 Results and Analysis

4.1 Main results

Model / Method	Avg % Played	Avg Quality	Clemscore	Statscore
Base Qwen3.5-2B official	-	-	13.05	44.02
Base Qwen3.5-2B local	28.57	41.70	11.91	-
Stage 1: Success SFT	85.60	53.46	45.76	39.74
Stage 2: Uniform weights	-	-	46.18	40.70
Stage 2: Weighted-turn	83.77	63.65	53.32	41.61
Stage 2: Weighted-turn (0.5 epoch)	82.34	64.45	53.07	40.62
Qwen-GuidePlay-2B	83.30	68.57	57.12	42.68

Table 3: Main results across training stages (*– indicates that the corresponding metric was not recorded.*)

On the public Playpen validation set used during development, the final model achieves a clemscore of 57.12 and a statscore of 42.68. This corresponds to a +44.07 clemscore improvement over the official Qwen3.5-2B base score on the same public validation setup. It also outperforms our strongest non-teacher baseline, Stage 2 Weighted-turn, by +3.80 clemscore and +1.07 statscore.

Official challenge final results. The organisers subsequently evaluated **Qwen-GuidePlay-2B** using the official LM Playschool Challenge final evaluation. Relative to the Qwen3.5-2B base model, our model obtains a +35.99 Playpen clemscore improvement, the second-highest Playpen clemscore delta among submitted systems. On the held-out challenge sets, it improves clemscore by +32.85 in-domain and +6.53 out-of-domain. Its Playpen statscore changes by -1.94 relative to the base model.

4.2 Playability and quality improve at different stages

The central pattern is simple, each stage improves a different aspect of interaction.

Comparison	Δ Avg % Played	Δ Avg Quality
Base local vs. Stage 1	+57.03	+11.76
Stage 1 vs. Stage 2 Weighted	-1.83	+10.19
Stage 2 Weighted vs. Final	-0.47	+4.92

Table 4: Playability and quality differences across training configurations.

Stage 1 mostly improves playability. The local base model plays only 28.57% of episodes, while Stage 1 plays 85.60%. This suggests that full successful trajectories teach the interactional grammar of Playpen: action format, turn structure, and stopping behaviour.

Stage 2 operates at nearly constant playability but raises average Quality from 53.46 to 63.65. This supports the hypothesis that factorising trajectories into state-action examples exposes the local decision signal more directly than full-dialogue imitation.

The final teacher-guided configuration further raises average Quality to 68.57 while maintaining similar playability. This suggests that the combined teacher-guided stage improves decision precision and strict-format behaviour rather than merely increasing the number of valid episodes.

Overall, the pipeline has a clear division of labour: complete trajectories make the model playable; value-weighted turns make it better; teacher-guided repair makes it more precise.

4.3 Ablations and negative results

We compare against four development variants: a heavier replay-and-repair mixture (RR) trained after Stage 2; RR-scaled merge, which interpolates the replay-repair LoRA delta before merging; Outcome-based weighting, which assigns heuristic weights using successful, failed, and aborted trajectories; and Hard Example Mining (HEM), which performs loss-based hard-example mining over Stage 2 rows.

To isolate the contribution of the teacher-guided data, we additionally train a matched control using the same Stage 1 initialisation and training hyperparameters, and the same 30,000 Stage 2 rows, but without the 1,000 teacher-judged and 100 repair examples. This control reaches 54.71 clemscore and 41.24 statscore, compared with 57.12 and 42.68 for the final model. Thus, adding the small teacher-guided component improves clemscore by +2.41 and statscore by +1.44 under the matched-data setting.

Method	Clemscore	Statscore	Interpretation
Stage 2 Weighted-turn	53.32	41.61	Strong non-teacher baseline
Stage 2 Weighted-turn, 0.50 epoch	53.07	40.62	More training did not help
RR	54.86	38.66	Replay/repair improved clem but hurt stat
RR-scaled merge, $\alpha = 0.75$	53.12	-	Scaling RR delta did not recover the gain
Outcome-based weighting	51.06	40.84	Outcome-based weighting was noisy
HEM	51.41	40.60	Loss-based hard mining underperformed
Matched 30k Stage 2 control	54.71	41.24	Same Stage 2 subset without teacher data
Qwen-GuidePlay-2B	57.12	42.68	Best balance

Table 5: Ablation results. Higher is better for both Clemscore and Statscore; α denotes the scaling applied to the replay–repair LoRA delta. A dash indicates that the corresponding metric was not recorded.

The ablations compare the final lightweight teacher-guided mixture against heavier alternatives: replay-and-repair training, scaled replay-repair merging, outcome-based weighting, and loss-based hard-example mining. These ablations are single development runs intended as diagnostic comparisons rather than exhaustive hyperparameter sweeps; implementation details are provided in Appendix B.2.

These results show that stronger intervention was not automatically better. RR improved clem-score over Stage 2 but reduced statscore, suggesting over-specialisation to benchmark-specific output behaviour. Outcome-based weighting introduced noisy credit assignment by weighting failed and aborted trajectories. HEM may have selected high-loss examples that were difficult because of length or context rather than because they were useful training frontiers. The final method works because it preserves the successful Stage 2 distribution and adds only a small corrective signal from the teacher.

5 Model Release and Reproducibility

Qwen-GuidePlay-2B is released as a full merged 2B Hugging Face checkpoint at [Hugging Face](#), with code available at [GitHub](#). The model is evaluated with the local Hugging Face backend. Evaluation follows the public Playpen setup, with standard local model registration, left padding, `eos_to_cull=<|im_end|>`, and the post-merge generation-configuration patch described above.

6 Limitations

The training method for Qwen-GuidePlay-2B does not explore online reinforcement learning, explicit search, or planning. The teacher does not propose new strategies; it only judges and repairs

existing successful-trace actions. The robustness of using an LLM as a teacher is not explored. Although Qwen-GuidePlay-2B improves substantially on the official held-out in-domain evaluation (+32.85 clem-score), the smaller out-of-domain improvement (+6.53) suggests that part of the gain remains domain-specific.

Acknowledgments

This work was supported in part by the Alan Turing Institute under Fundamental Research Project No. PP00029.

References

- Anne Beyer, Kranti Chalamalasetti, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2024. [clembench-2024: A challenging, dynamic, complementary, multilingual benchmark and underlying flexible framework for llms as multi-action agents](#). *Preprint*, arXiv:2405.20859.
- Jerome Bruner. 1985. [Child’s talk: Learning to use language](#). *Child Language Teaching and Therapy*, 1(1):111–114.
- Kranti Chalamalasetti, Jana Götze, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2023. [clembench: Using game play to evaluate chat-optimized language models as conversational agents](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11174–11219, Singapore. Association for Computational Linguistics.
- Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. [Qlora: efficient finetuning of quantized llms](#). In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS ’23*, Red Hook, NY, USA. Curran Associates Inc.
- Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, KaShun SHUM, and Tong Zhang. 2023. [RAFT: Reward ranked finetuning for generative foundation model alignment](#). *Transactions on Machine Learning Research*.
- Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. 2023. [Reinforced self-training \(rest\) for language modeling](#). *Preprint*, arXiv:2308.08998.
- Nicola Horst, Davide Mazzaccara, Antonia Schmidt, Michael Sullivan, Filippo Momentè, Luca Franceschetti, Philipp Sadler, Sherzod Hakimov, Alberto Testoni, Raffaella Bernardi, Raquel

- Fernández, Alexander Koller, Oliver Lemon, David Schlangen, Mario Giulianelli, and Alessandro Suglia. 2025. [Playpen: An environment for exploring learning from dialogue game feedback](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29854–29891, Suzhou, China. Association for Computational Linguistics.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. 2019. [Advantage-weighted regression: Simple and scalable off-policy reinforcement learning](#). *Preprint*, arXiv:1910.00177.
- Jan Peters and Stefan Schaal. 2007. [Reinforcement learning by reward-weighted regression for operational space control](#). In *Proceedings of the 24th International Conference on Machine Learning, ICML '07*, page 745–750, New York, NY, USA. Association for Computing Machinery.
- Qwen Team. 2026. [Qwen3.5: Towards native multi-modal agents](#).
- David Schlangen, Sherzod Hakimov, Chalamalasetti Kranti, Jonathan Jordan, and Philipp Sadler. 2025. [A third paradigm for llm evaluation: Dialogue game-based evaluation using clembench](#). *ArXiv*, abs/2507.08491.
- Gemma Team, Sherif El Abd, Vaibhav Aggarwal, Robin Algayres, Alek Andreev, Olivier Bachem, Ian Ballantyne, Cormac Brick, Victor Cărbune, Michelle Casbon, Mayank Chaturvedi, Aditya Chawla, Victor Cotruta, Alice Coucke, Phil Culliton, Robert Dadashi, Lucas Dixon, Mohamed Elhawaty, Utku Evci, and 304 others. 2026. [Gemma 4 technical report](#). *Preprint*, arXiv:2607.02770.
- L. S. Vygotsky. 1978. *Mind in Society: Development of Higher Psychological Processes*. Harvard University Press.
- Qing Wang, Jiechao Xiong, Lei Han, peng sun, Han Liu, and Tong Zhang. 2018. [Exponentially weighted imitation learning for batched historical data](#). In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc.
- Kaixuan Xu, Jiajun Chai, Sicheng Li, Yuqian Fu, Yuanheng Zhu, and Dongbin Zhao. 2025. [Dipllm: fine-tuning llm for strategic decision-making in diplomacy](#). In *Proceedings of the 42nd International Conference on Machine Learning, ICML'25*. JMLR.org.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. [Star: self-taught reasoner bootstrapping reasoning with reasoning](#). In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22*, Red Hook, NY, USA. Curran Associates Inc.

A Appendix

A.1 Judging prompt

Each selected Stage 2 row is sent to the teacher as a two-message conversation. The system message is:

You are a strict JSON-only data-quality judge and synthetic-error generator for a dialogue-game agent. Return one valid JSON object only. Do not use markdown. Do not explain outside JSON.

The user message is the following template:

You are judging one training example for a dialogue-game agent.

The agent must obey a strict output contract:

- Output exactly one final answer.
- For movement tasks, valid movement answers look like exactly:
 - GO: north
 - GO: south
 - GO: east
 - GO: west
- If the task is complete, the answer should be exactly:
 - DONE
- No explanation.
- No markdown.
- No role prefix.
- No punctuation around the answer.
- No stop tokens.
- No multiple actions.

You are given:

1. The dialogue state/history BEFORE the assistant answer.
2. The original gold assistant answer from a successful or high-quality training trace.
3. The old Stage 2 training weight.

You must NOT invent a new gold answer. The clean target must remain the provided gold answer.

Your tasks:

- A. Score how good this training row is for teaching strict, correct dialogue-game behaviour.
- B. Generate realistic bad assistant outputs that a smaller model might produce for this exact example.
 - These bad outputs should be close to the gold answer but invalid or less strict.

Examples of badness:

- extra explanation
- missing GO: prefix
- wrong casing
- punctuation
- markdown
- role leakage
- stop-token leakage
- multiple actions
- premature DONE
- verbose DONE

Return exactly one JSON object and nothing else.

Required JSON schema:

```
{
  "format_score": 1.0,
  "action_score": 1.0,
  "task_progress_score": 1.0,
  "done_score": 1.0,
  "overall_score": 1.0,
  "keep": true,
  "reason": "short reason",
  "bad_outputs": ["bad output 1", "bad output 2"]
}
```

Rules:

- Scores must be floats from 0.0 to 1.0.
- "overall_score" should summarize whether the row is useful for training.
- "keep" should be false only if the row looks malformed, useless, contradictory, or too ambiguous.
- "bad_outputs" must contain at most {max_bad_outputs} strings.
- "bad_outputs" must NOT include the exact clean target.
- "bad_outputs" must NOT be empty unless keep is false.

Row index:

{row_index}

Old Stage 2 weight:

{original_weight}

Dialogue state/history before answer:

{context}

Gold assistant answer:

{target}

Placeholder	Replaced with
{context}	The message history before the assistant action, serialised as JSON and truncated to its final 7,000 characters (the most recent game state matters most for the next action).
{target}	The original gold assistant answer from the Stage 2 row.
{original_weight}	The row's existing Stage 2 heuristic value.
{row_index}	The row's index in the Stage 2 file, used for resumable processing and provenance tracking.
{max_bad_outputs}	4 in all our runs.

Table 6: Template placeholders in the judging prompt.

A.2 Repair prompt

Each kept bad_output produces one repair training example. The prompt is a single user message built from the following template; the supervised target is always the original gold action, never the teacher's text:

You are playing a dialogue game. The previous assistant answer was invalid because it did not follow the exact required output format or it contained extra text.

Game state/history before the assistant answer: {context}

Invalid assistant answer: {bad}

Rewrite the assistant answer correctly. Output only one final valid answer and nothing else.

B Teacher-generation details

Teacher judgements were generated with Gemma-4-31B-it using temperature 0.1 and a maximum generation length of 700 tokens. The serialised dialogue context was limited to 7,000 characters, retaining the most recent context when necessary. The teacher could generate at most four candidate invalid outputs per judged example, and at most two repair examples per source row were retained. We used random seed 28.

B.1 Teacher-data selection

The teacher-generated judged file contained 2,943 retained Stage 2 rows, of which 2,888 had an overall score of at least 0.70. The final Stage 3 mixture samples 1,000 rows from this eligible judged pool. The repair-generation procedure produced 5,208 repair candidates derived from 2,604 Stage 2 source rows, from which 100 repair examples were sampled randomly. Before sampling the 30,000 ordinary Stage 2 rows, rows used for teacher judging were excluded from that pool.

B.2 Ablation configurations

The Stage 2 Weighted-turn 0.50-epoch variant uses the same weighted-turn training setup as Stage 2, but doubles training from 0.25 to 0.50 epoch. Outcome-based weighting replaces the Stage 2 weighting scheme with heuristic weights derived from successful and failed/aborted trajectories. HEM performs loss-based hard-example mining over Stage 2 examples. RR applies the heavier replay-and-repair intervention described in Section 4.3. RR-scaled merge does not introduce an additional training objective; it scales the replay repair LoRA delta by $\alpha = 0.75$ before merging.

Raising a Small Language Model: From Imitation to Curiosity-Driven Learning in Dialogue Games

Varad Srivastava

AI Lab, Barclays

varad.srivastava@barclays.com

Abstract

Children do not learn language from text corpora, rather they learn it in interaction by imitating competent speakers, receiving feedback, practicing, being corrected, and playing. The LM Playschool Challenge asks whether language models can profit from analogous regimes. We take the smallest recommended base model, Qwen3.5-2B (clemscore 13.63 in our evaluation environment), and traverse a sequence of five interaction-derived learning regimes that mirror a developmental progression: imitation of competent others (supervised fine-tuning on success-filtered transcripts), learning from outcome contrasts (direct preference optimisation), self-imitation (fine-tuning on the model’s own successes), corrective feedback (contrastive pairs from the model’s own failures against its own and expert successes), and intrinsically motivated trial-and-error (online GRPO with a novelty bonus). The offline regimes carry the model to 67.6 clemscore on our validation evaluation (+54.0 over the base model). On the organisers’ held-out evaluation, our two submitted checkpoints achieve the largest out-of-domain gains of any submission in the challenge (+11.9 and +10.8 clemscore, against a maximum of +6.5 for any other entry), with in-domain gains of +33.2 and +37.3 and static-benchmark performance unchanged. Online GRPO *regresses* from its initialisation (−5.0) unless augmented with a random-network-distillation novelty bonus, which fully recovers the loss (+5.0 over the plain-GRPO control), evidence that the bonus primarily counteracts the forgetting that narrow online training induces on untrained games.

1 Introduction

The majority of language models knowledge comes from learning by observation: next-token prediction over text produced by others, refined by single-turn instruction tuning. However, human language acquisition looks different. It is usage-

based and interactive (Tomasello, 2003): competence grows through joint activity with more capable partners—the zone of proximal development (Vygotsky, 1978), through feedback on one’s own productions, and through self-motivated play at the edge of one’s ability, sought out by intrinsically motivated exploration (Berlyne, 1960; Oudeyer et al., 2007). The LM Playschool Challenge operationalises this contrast: models are placed in Game-Master-mediated dialogue games (Chalamalasetti et al., 2023) inside the playpen environment (Horst et al., 2025), and submissions are judged by how much interactive competence (*clemscore*) they gain over a base model, while a static-benchmark (*statscore*) checks that general abilities survive. Appendix A describes the games and five static benchmarks.

We treat the challenge as an opportunity to sweep the space of interaction-derived developmental learning regimes in a controlled way, on a deliberately small model where every effect is visible. Qwen3.5-2B (Qwen Team, 2026) starts at clemscore 13.63: it does not so much lose games as fail to play them, aborting episodes by violating the games’ strict output formats. From this starting point we apply in sequence, five regimes chosen to mirror a developmental progression (Section 3): (R1) imitation of competent others; (R2) learning from success/failure contrasts; (R3) self-imitation of one’s own successes; (R4) corrective feedback on one’s own failures; and (R5) intrinsically motivated online play, with and without a curiosity signal.

Our contributions are: (i) a compute-light offline recipe (R1+R2, one GPU, under five training hours) that lifts a 2B model by +37.3 clemscore in-domain and +10.8 out-of-domain on the challenge’s held-out evaluation—the largest out-of-domain improvement among all submissions, at every base-model scale—while leaving statscore unchanged; (ii) a controlled demonstration that on-

line GRPO from a strong initialisation *regresses* unless equipped with an intrinsic novelty bonus, which recovers the regression exactly (R5); (iii) a systematic comparison showing that regimes using the model’s own behaviour (R3–R5) at best match the strongest offline regime and more often regress, at this scale and budget.

2 Background and Related Work

Dialogue games as curriculum. clembench (Chalamalasetti et al., 2023) investigates functional language competence through games with hard, parser-enforced move formats; playpen (Horst et al., 2025) turns these games into a learning environment and releases a corpus of gameplay transcripts. The playpen paper’s reference experiments found SFT on successful transcripts helps in-domain but erodes other skills, while online GRPO (Shao et al., 2024) gives smaller, balanced gains—a trade-off our regime sweep revisits with explicit controls.

Learning regimes and their cognitive analogues. Imitation of filtered demonstrations underlies STaR (Zelikman et al., 2022), rejection-sampling fine-tuning (Yuan et al., 2023), ReST (Gulcehre et al., 2023) and SPIN (Chen et al., 2024); it is the machine analogue of observational learning from competent others (Tomasello, 2003). Preference learning (Christiano et al., 2017; Stiennon et al., 2020; Ouyang et al., 2022; Rafailov et al., 2023; Ethayarajh et al., 2024) corresponds to learning from evaluative feedback. Intrinsic motivation—curiosity as prediction error (Pathak et al., 2017; Burda et al., 2019) or as learning progress (Oudeyer et al., 2007)—formalises self-directed play; we use random network distillation (Burda et al., 2019), to our knowledge its first application to Game-Master-mediated dialogue games.

Forgetting. Narrow fine-tuning erodes prior abilities (McCloskey and Cohen, 1989; Kirkpatrick et al., 2017; Luo et al., 2023); replaying general data mitigates it (Scialom et al., 2022). The challenge’s statscore (BBH, MMLU-Pro, IFEval, CLadder, EQ-Bench; Suzgun et al., 2023; Wang et al., 2024; Zhou et al., 2023; Jin et al., 2023; Paech, 2023) makes the erosion measurable; a 12% SmolTalk (Hugging Face TB Research, 2024) mix in R1 is our (successful) mitigation.

3 Learning Regimes

Throughout, π_θ denotes the policy, trained with LoRA adapters (Hu et al., 2022) that are merged after each regime; R3–R5 each initialise from the merged R2 model.

3.1 R1: Imitation of competent others

Rationale. The first thing a novice does in a new social practice is watch someone who can already do it. Observational learning (Bandura, 1977) and usage-based accounts of acquisition (Tomasello, 2003) agree that imitation of competent others is the entry point into a convention-governed activity, and dialogue games are nothing if not convention-governed as demonstrated here too: the base model’s 13.63 is overwhelmingly a *protocol* failure (aborted episodes), not a strategic one. Our hypothesis for R1 was therefore narrow: imitating successful play should first of all teach the move grammar, with strategy learning transferred as a by-product.

Construction. From the public playpen-data interactions corpus (34,909 per-player trajectories annotated with game, experiment, task id, role, source model and outcome) we keep only *successful* episodes, map all Game-Master and partner turns to the user role (merging consecutive same-role turns), and drop trajectories without an assistant turn ($\approx 20.2k$ trajectories) for SFT. Anticipating the forgetting cost that imitation of a narrow practice incurs (Luo et al., 2023), we mix in general instruction data (SmolTalk, smol-magpie-ultra) at $\approx 12\%$, for 22,626 conversations, and minimise cross-entropy on assistant tokens only—to ensure the model learns to *produce moves*, not to imitate the Game Master.

3.2 R2: Learning from outcome contrasts

Rationale. Imitation shows the learner only what success looks like and it carries no information about failure, and the R1 model’s residue confirmed the gap: format compliance was nearly solved while move *quality* lagged. Learners, by contrast, exploit evaluative feedback—signals that grade their productions rather than merely exemplify good ones. Explicit correction is rare and noisy in child-directed speech (Marcus, 1993); children encounter the second signal largely through communicative outcome, when an utterance fails to achieve its interactive goal and the ensuing repair sequence turns the conversation itself into evidence

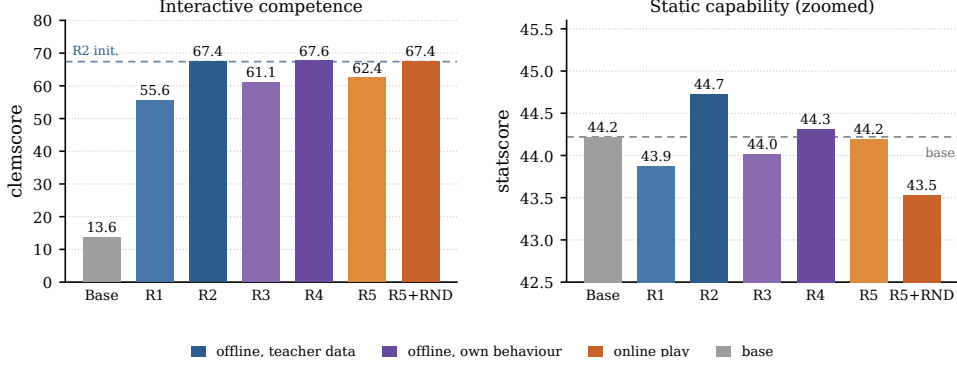


Figure 1: Learning regimes across both metrics, in a single frozen evaluation environment. Left: clemscore; the dashed line marks the R2 initialisation shared by R3–R5. Right: statscore on a zoomed axis, flat across all regimes. Each regime applies one training stage on top of the previous model: SFT (R1 - Imitation, R3 - Self-imitation), DPO (R2 - Contrast, R4 - Correction), and online GRPO without and with an intrinsic novelty bonus (R5, R5+RND). Colour encodes the source of the learning signal.

(Clark, 2020). The games make such feedback almost free: because episodes on the same instance share an identical initial state, a successful and a failed first move form a minimal pair whose only difference is quality. R2 tests whether this contrastive signal repairs exactly the residue imitation left.

Construction. For a fixed (game, experiment, task id, role) we pair the first move y^+ of a successful episode with the first move y^- of an aborted episode (falling back to lost ones), oversampling still-abort-prone games $4\times$ (3,418 pairs), and optimise the DPO objective (Rafailov et al., 2023)

$$\mathcal{L}_{\text{DPO}} = -\log \sigma \left(\beta \log \frac{\pi_{\theta}(y^+|x)}{\pi_{\text{ref}}(y^+|x)} - \beta \log \frac{\pi_{\theta}(y^-|x)}{\pi_{\text{ref}}(y^-|x)} \right),$$

with the frozen R1 model as π_{ref} : learning from contrast, anchored to what imitation established.

3.3 R3: Self-imitation

Rationale. R1 and R2 learn *from records of interaction*; the challenge’s deeper question concerns learning *in interaction*. The intuitive step across that line is practice: let the model play, keep what worked, and rehearse it—the machine analogue of a child repeating a newly mastered routine, and the pattern behind ReST-style methods (Gulcehre et al., 2023). Because the R2 model now succeeds often, its own play is a plentiful source of positive examples; R3 asks whether rehearsing them consolidates competence.

Construction. We play the R2 model through the full public instance sets of eight games (490

episodes, 63% successful; validation-overlapping instances excluded), reconstruct per-player trajectories from the interaction logs (516 examples), and run one light SFT epoch. Unlike R2, this objective has *no anchor*: whatever gradient the data carries, the policy follows.

3.4 R4: Corrective feedback

Rationale. R3 regressed (Section 5), and its failure suggested a diagnosis with a developmental intuition: rehearsing one’s own successes adds little information and when delivered without an anchor, it perturbs what already works. What children demonstrably profit from instead is correction of their *errors*—adult reformulations, or *recasts*, that follow a faulty production with a well-formed one in the same context (Chouinard and Clark, 2003). On the contrast theory of negative input, the corrective signal lies precisely in the juxtaposition of the learner’s erroneous form with a correct alternative for the same intended meaning (Saxton, 1997), and such input improves subsequent grammaticality where positive models alone do not (Saxton et al., 2005). The games again supply this almost for free: wherever our model failed an instance, the transcript corpus usually contains a stronger model succeeding on the identical instance. R4 therefore reuses the model’s own play but as *contrastive evidence*—pairing its failures against its own or a stronger model’s successes, taught through the anchored objective that R2 validated, testing at once the recast hypothesis and the anchoring hypothesis raised by R3.

Construction. From two rollout passes of the R2 model (greedy and sampled, $t=0.7$) we build 201 first-move preference pairs from two sources: (i) *on-policy* pairs (107) from instances where the two passes disagreed in outcome—chosen is the model’s own successful first move, rejected its own failed one on the identical instance; and (ii) *hybrid* pairs (94) from instances the model failed in both passes—chosen is a stronger model’s success on that instance (a recast), rejected the model’s own failure. We then run one DPO epoch from the merged R2 model. The on-policy pairs are drawn precisely from the model’s competence frontier (instances of variable outcome), linking R4 to the learning-progress principle we exploit in R5.

3.5 R5: Intrinsically motivated play

Rationale. Our last regime is: genuine online play, where the learner’s own exploration generates the curriculum. Developmental theory is specific about what makes play educational: children do not sample activities uniformly but gravitate toward tasks of intermediate difficulty—neither mastered nor impossible—driven by intrinsic signals such as novelty and learning progress (Berlyne, 1960; Kidd and Hayden, 2015; Oudeyer et al., 2007); machine counterparts formalise curiosity as prediction error (Pathak et al., 2017; Burda et al., 2019). This predicts a specific failure mode for *plain* online RL in our setting: since the R2 policy already succeeds on most training instances, many rollout groups are uniformly successful or uniformly failed, outcome-based advantages degenerate, and the surviving gradient is noise that can only damage a competent policy - in particular by drifting behaviour on games outside the online training subset. This is akin to a forgetting effect that in theory a novelty bonus should curb by preserving informative gradient. R5 tests the prediction with a controlled pair: GRPO with and without a novelty bonus, all else identical.

Construction. For each sampled instance x we roll out K episodes $\{\tau_1, \dots, \tau_K\}$ through the Game Master and compute group-relative advantages over episode returns (Shao et al., 2024):

$$A_i = \frac{r_i - \text{mean}(r_{1:K})}{\text{std}(r_{1:K}) + \varepsilon}.$$

The policy is updated on the agent’s own tokens only (environment tokens masked), with importance ratios $\rho_{i,t} = \pi_\theta(a_{i,t} \mid s_{i,t}) / \pi_{\theta_{\text{old}}}(a_{i,t} \mid s_{i,t})$

against the pre-update policy, and a per-token KL penalty to the frozen R2 reference via the low-variance k_3 estimator (Schulman, 2020), with $\Delta_{i,t} = \log \pi_{\text{ref}} - \log \pi_\theta$:

$$\mathcal{L} = -\frac{1}{|T|} \sum_{i,t} \left[\min(\rho_{i,t} A_i, \text{clip}(\rho_{i,t}, 1 \pm \epsilon) A_i) - \beta_{\text{KL}}(e^{\Delta_{i,t}} - \Delta_{i,t} - 1) \right].$$

Intrinsic novelty (RND). In the intrinsic arm, we add a bonus that is large when the agent produces behaviour unlike what it has produced before, and decays as that behaviour becomes familiar. Following random network distillation (Burda et al., 2019), familiarity is measured by how well a small trained network can predict the output of a fixed, randomly initialised one: both see the same input, the target f is frozen, and the predictor $\hat{f}$ is trained to match it on states actually visited. On behaviour seen many times the predictor has converged and the error is small; on novel behaviour it has not, and the error is large. The input is $\phi(\tau)$, the mean of a frozen random embedding of the agent’s own action tokens in episode τ ; the embedding is frozen so that the novelty measure does not shift as the policy itself is updated. With running normalisation σ_b over recent errors,

$$b(\tau) = \text{clip} \left(\frac{\|\hat{f}(\phi(\tau)) - f(\phi(\tau))\|_2^2}{\sigma_b}, 0, 5 \right),$$

$$r_i^{\text{tot}} = r_i + \lambda b(\tau_i), \quad \lambda = 0.1,$$

so each episode’s return is augmented by its novelty before advantages are computed. Because advantages are group-normalised, any bonus shared by all K rollouts of an instance cancels: only *within-group* novelty differences carry gradient, so the signal rewards the more novel of two attempts at the same game state rather than novelty in general. Crucially, this keeps gradient flowing when outcomes alone provide none—when all K rollouts succeed, or all fail, the extrinsic advantages are uniformly zero but the novelty differences are not. The bonus exists only during training; evaluation uses game outcomes exclusively.

4 Experimental Setup

Base model and adaptation. We adapt Qwen3.5-2B, more details are provided in Appendix B.

Data. All game data derives from the public playpen-data release (interactions and instances configurations); general data from SmolTalk. Sizes

Model	clem	Δ	stat
Qwen3.5-2B (base)	13.63	–	44.22
R1 imitation (SFT)	55.61	+41.98	43.87
R2 outcome contrast (DPO) [†]	67.39	+53.76	44.72
R3 self-imitation	61.06	+47.43	44.01
R4 corrective feedback	67.64	+54.01	44.31
R5 GRPO (no intrinsic)	62.43	+48.80	44.19
R5 GRPO + RND	67.44	+53.81	43.53

Table 1: Frozen-environment results (validation split). Δ = clemscore gain over the base model measured in the same environment.[†] = the model submitted to the challenge (submission predates R4/R5). For cross-environment context, the organisers’ held-out evaluation scores the base model at 13.41 in-domain and the untuned Qwen3.5-27B at 64.34 (Section 5.1)

per regime as in Section 3. No validation-split or test data is used for training; self-play instances overlapping the validation split are excluded (21/46 of 490 episodes in R3/R4 self-play data collection).

Compute. R1 $\approx$ 3.3 h and R2 $\approx$ 0.6 h on one A100 40GB; R3–R5 (generation + training) and all evaluations on one 96GB workstation GPU; total budget $\approx$ 25 GPU-hours.

Evaluation. Official pipeline (playpen eval) on the validation split, both suites, greedy decoding (pipeline default). All reported numbers come from a single frozen environment: Python 3.11 virtual environment, clemcore at the version pinned by the playpen.

5 Results

Table 1 and Figure 1 summarise all regimes.

Imitation buys protocol; contrast buys policy. R1 lifts the model +41.98, almost entirely by eliminating aborted episodes (most games reach 100% played). R2 adds another +11.78 with %-played essentially flat and quality rising—the two factors of the clemscore product respond to two different signals, cleanly separated by the two stages. Statscore never moves more than ± 0.7 from base across all seven rows: the R1 mix-in suffices, and no later regime reintroduces forgetting on the static side.

Self-imitation regresses; correction does not. R3 costs 6.3 points relative to its R2 initialisation. Practising one’s own successes, delivered as unanchored SFT, perturbs behaviour both on games absent from the self-play data and on games present in it; we attribute latter to using no general-data replay

(unlike R1) and trained on positives the policy already produced (since regression isn’t reversion toward the pre-DPO model: although R3’s score falls between R1 and R2, only 21% of its per-game deviation from R2 lies along the $R1 \rightarrow R2$ direction, remainder being orthogonal). R4, which delivers the model’s own play as contrastive pairs through the KL-anchored preference objective, does not regress and posts the best score (67.64)—though its margin over R2 (+0.25) is within the run-to-run variability, so we describe R2, R4 and R5+RND as statistically indistinguishable at the top.

Online play needs curiosity. The controlled R5 comparison is our central finding, and the per-game breakdown (Appendix: Figure 4) explains it. On the four games R5 trained on, *both* GRPO arms improved or held relative to R2 (e.g., adventuregame: 26.7 $\rightarrow$ 33.3 $\rightarrow$ 40.0; textmapworld: 65.0 $\rightarrow$ 67.0 $\rightarrow$ 68.2). Plain GRPO’s overall regression of 4.96 points came almost entirely from *untrained* two-player games (codenames: 30.8 $\rightarrow$ 15.4 $\rightarrow$ 23.1; guesswhat: 33.3 $\rightarrow$ 22.2 $\rightarrow$ 27.8; taboo: 66.7 $\rightarrow$ 58.3 $\rightarrow$ 66.7). That is, online updates on a narrow subset of games induced forgetting on the rest—the same failure mode we saw in R3. The RND bonus, identical in seed, steps, data, and hyperparameters, and differing only in λ , recovers this loss almost entirely overall (67.44, +5.01 over the control), and the recovery is concentrated in exactly those untrained games. The intrinsic signal thus acts less as an explorer than as a regulariser against distributional drift; by keeping within-group gradients informative even where extrinsic outcomes are uniform, it limits how far the policy wanders on games it is not currently practising.

5.1 Held-out evaluation

The organisers evaluated all submissions on a closed set of games not released during the challenge: 16 in-domain games (the clembench suite, with adventuregame and the static benchmarks on reduced instance sets) and 13 newly written out-of-domain games. We submitted R1 and R2; R3–R5 postdate the submission deadline and were not evaluated.

Two results stand out. First, both checkpoints improve out-of-domain, and by a wider margin than any other submission: the next-best entry gains +6.53, and *every* submission built on a larger base (4B, 9B, 27B) *declines* out-of-domain, by up to

	ID Δ	OOD Δ	stat Δ
R1 imitation	+33.16	+ 11.90	+0.11
R2 contrast	+ 37.34	+10.84	-0.71
best other entry	+32.85	+6.53	—

Table 2: Held-out evaluation (organisers’ run; deltas over the Qwen3.5-2B base, which scores 13.41 in-domain, 3.72 out-of-domain and 44.24 on statscore). “Best other entry” is the strongest competing submission on each metric across all base-model scales.

−13.49. What the offline recipe teaches transfers to games it has never seen, which is not true of the field’s other approaches.

Second, the two regimes trade off in the direction our validation analysis predicts. R2 is clearly stronger in-domain (+37.34 vs. +33.16) but slightly weaker out-of-domain (+10.84 vs. +11.90), with a wider generalisation gap (−36.19 vs. −30.95). Imitation of expert transcripts installs competence that transfers; contrastive refinement against outcomes buys additional in-domain quality that is more specific to the games it was derived from. In absolute terms our best submission reaches 50.75 in-domain, short of the untuned 27B baseline (64.34): the advantage over that baseline that we observed on the validation split does not survive the held-out evaluation.

6 Discussion

Two factors, two currencies. Imitation moved %-played and barely touched quality; contrastive feedback moved quality and barely touched %-played. Demonstrations specify the *support* of good behaviour (what a legal move looks like), while contrasts specify its *gradient* (which legal move is better)—and dialogue games, defined by conventions plus quality distinctions, reward exactly this two-course curriculum, in the order children meet it: immersion first, feedback on one’s own attempts second.

Own behaviour is a treacherous teacher. Every self-generated-data regime underperformed the best offline one, but how they failed is the lesson. Unanchored self-imitation (R3) hurt; the same self-play episodes, used as contrastive evidence inside an anchored objective (R4) matched the best model; online updates from own play hurt without curiosity and were rescued by it (R5). Own-behaviour data is least informative exactly where the policy is already competent—successes teach little, out-

come advantages vanish, so the residual gradient is dominated by noise, which unanchored objectives turn into drift. What worked either re-anchored the policy (KL to a trusted reference) or re-injected information where outcomes carried none (novelty at the competence boundary). Curiosity here acted as a stabiliser rather than the usual explorer-under-sparse-reward—consistent with learning-progress accounts (Oudeyer et al., 2007), and predicting that the RND advantage should shrink with larger groups and grow as the policy nears ceiling. Two-player games with a learning partner, longer horizons, and learning-progress-based instance selection are our natural next steps.

What transferred. The held-out evaluation supplies the test our validation split could not. Both offline regimes generalise to unseen games, and imitation generalises slightly better than contrast—consistent with the division of labour above, in which demonstrations specify what a legal, sensible move looks like while contrasts refine which of them is better in a particular game. That every larger-model submission lost ground out-of-domain suggests the effect turns on what the training signal encodes rather than on scale.

Limitations

All results are for one base model at one scale; our regime sweep is evaluated on a small validation split, and only R1 and R2 were run on the held-out set. R1, R2 and R4 depend on transcripts from stronger models, so the recipe does not demonstrate teacher-free learning; R5 covers only single-player games, as the training interface lacks a partner mechanism. The R5 comparison uses one seed per arm. The mechanism behind R3’s regression remains unidentified; we rule out format mismatch and reversion toward the pre-DPO policy (Section 5). Belief-intensive games (privateshared, codenames clue-giving) remain the weakest and untargeted by our methods. Training curves (Appendix: Figure 3) are noisy over 60 steps and do not separate the arms; the RND effect appears in final evaluation.

References

- Albert Bandura. 1977. *Social Learning Theory*. Prentice Hall.
- Daniel E. Berlyne. 1960. *Conflict, Arousal, and Curiosity*. McGraw-Hill, New York.

- Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. 2019. Exploration by random network distillation. In *International Conference on Learning Representations*.
- Kranti Chalamalasetti, Jana Götze, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2023. clembench: Using game play to evaluate chat-optimized language models as conversational agents. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.
- Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. 2024. Self-play fine-tuning converts weak language models to strong language models. In *Proceedings of the 41st International Conference on Machine Learning*.
- Michelle M. Chouinard and Eve V. Clark. 2003. [Adult reformulations of child errors as negative evidence](#). *Journal of Child Language*, 30(3):637–669.
- Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*.
- Eve V. Clark. 2020. [Conversational repair and the acquisition of language](#). *Discourse Processes*, 57(5-6):441–459.
- Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. KTO: Model alignment as prospect theoretic optimization. In *Proceedings of the 41st International Conference on Machine Learning*.
- Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, Ksenia Konyushkova, Lotte Weerts, Abhishek Sharma, Aditya Siddhant, Alex Ahern, Miaosen Wang, Chenjie Gu, Wolfgang Macherey, Arnaud Doucet, Orhan Firat, and Nando de Freitas. 2023. Reinforced self-training (ReST) for language modeling. *arXiv preprint arXiv:2308.08998*.
- Nicola Horst, Davide Mazzaccara, Antonia Schmidt, Michael Sullivan, Filippo Momentè, Luca Franceschetti, Philipp Sadler, Sherzod Hakimov, Alberto Testoni, Raffaella Bernardi, Raquel Fernández, Alexander Koller, Oliver Lemon, David Schlangen, Mario Giulianelli, and Alessandro Suglia. 2025. Playpen: An environment for exploring learning from dialogue game feedback. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.
- Hugging Face TB Research. 2024. Smoltalk. <https://huggingface.co/datasets/HuggingFaceTB/smoltalk>.
- Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng Lyu, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, and Bernhard Schölkopf. 2023. CLadder: Assessing causal reasoning in language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 31038–31065.
- Celeste Kidd and Benjamin Y. Hayden. 2015. [The psychology and neuroscience of curiosity](#). *Neuron*, 88(3):449–460.
- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. [Overcoming catastrophic forgetting in neural networks](#). *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.
- Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*.
- Gary F. Marcus. 1993. [Negative evidence in language acquisition](#). *Cognition*, 46(1):53–85.
- Michael McCloskey and Neal J. Cohen. 1989. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation*, 24:109–165.
- Pierre-Yves Oudeyer, Frédéric Kaplan, and Verena V. Hafner. 2007. Intrinsic motivation systems for autonomous mental development. *IEEE Transactions on Evolutionary Computation*, 11(2):265–286.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744.
- Samuel J. Paech. 2023. EQ-Bench: An emotional intelligence benchmark for large language models. *arXiv preprint arXiv:2312.06281*.
- Deepak Pathak, Pulkrit Agrawal, Alexei A. Efros, and Trevor Darrell. 2017. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the 34th International Conference on Machine Learning*.
- Qwen Team. 2026. [Qwen3.5: Towards native multimodal agents](#).
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language

- model is secretly a reward model. In *Advances in Neural Information Processing Systems*.
- Matthew Saxton. 1997. [The contrast theory of negative input](#). *Journal of Child Language*, 24(1):139–161.
- Matthew Saxton, Phillip Backley, and Clare Gallaway. 2005. Negative input for grammatical errors: Effects after a lag of 12 weeks. *Journal of Child Language*, 32(3):643–672.
- John Schulman. 2020. Approximating KL divergence. [://joschu.net/blog/kl-approx.html](https://joschu.net/blog/kl-approx.html).
- Thomas Scialom, Tuhin Chakrabarty, and Smaranda Muresan. 2022. Fine-tuned language models are continual learners. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F. Christiano. 2020. Learning to summarize with human feedback. In *Advances in Neural Information Processing Systems*.
- Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc V. Le, Ed H. Chi, Denny Zhou, and Jason Wei. 2023. Challenging BIG-bench tasks and whether chain-of-thought can solve them. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051.
- Michael Tomasello. 2003. *Constructing a Language: A Usage-Based Theory of Language Acquisition*. Harvard University Press.
- Lev S. Vygotsky. 1978. *Mind in Society: The Development of Higher Psychological Processes*. Harvard University Press.
- Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhui Chen. 2024. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems*.
- Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. Scaling relationship on learning mathematical reasoning with large language models. *arXiv preprint arXiv:2308.01825*.
- Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. 2022. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems*.
- Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Sid-dhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

A The Playpen: Games and Benchmarks

Dialogue games (clemscore). The evaluation suite comprises fourteen games, each mediated by a programmatic Game Master that delivers instructions, validates every move against a strict format (violations abort the episode), and scores completed episodes for success and efficiency. The games cluster into four families that stress different competences. *Word games*: taboo (describe a word without using it or listed relatives, for a guessing partner), wordle and its variants wordle_withclue and wordle_withcritic (letter-feedback word guessing, optionally with a clue giver or a critic to negotiate with), and codenames (one-word clues that connect several target words while avoiding others). *Reference and drawing games*: referencegame (describe which of three grids is the target), imagegame (instruct a partner to reproduce an ASCII drawing), matchit_ascii (two players must decide whether they see the same image). *Information-tracking games*: guesswhat (twenty-questions-style search over a candidate list), privateshared (answer probes about which information has already been shared with the interlocutor—explicit book-keeping of common ground). *Exploration games*: adventuregame (text-adventure task completion) and the textmapworld family (navigate a graph-structured world; variants add graph reasoning and goal rooms). Success requires simultaneously respecting the move grammar (*protocol*) and playing well (*policy*); clemscore is the product of macro-averaged %-played (non-aborted episodes) and macro-averaged quality on played episodes, so the two factors remain separately visible.

Static benchmarks (statscore). The second metric averages five standard single-turn benchmarks: BBH (Suzgun et al., 2023) (multi-step reasoning), MMLU-Pro (Wang et al., 2024) (broad knowledge, hard multiple choice), IFEval (Zhou et al., 2023) (verifiable instruction following), CLadder (Jin et al., 2023) (causal reasoning) and EQ-Bench (Paeck, 2023) (emotional understanding). Its role is

	Objective	LoRA	Optimisation
R1	SFT	$r=32, \alpha=64$	2 ep, lr 10^{-4}
R2	DPO	$r=16, \alpha=32$	1 ep, lr 5×10^{-6}
R3	SFT	$r=16, \alpha=32$	1 ep, lr 2×10^{-5}
R4	DPO	$r=16, \alpha=32$	1 ep, lr 5×10^{-6}
R5	GRPO	$r=16, \alpha=32$	60 steps, lr 5×10^{-6}

Table 3: Adaptation settings per regime. All regimes use LoRA on every attention and MLP projection with dropout 0.05, bf16, effective batch 16, and maximum length 4096; SFT computes loss on assistant tokens only. Cosine schedules throughout (warmup 0.03). DPO uses $\beta=0.1$; GRPO uses AdamW with gradient clipping 1.0, $\epsilon=0.2$, $\beta_{KL}=0.04$, $K=6$ rollouts over 4 instances per step. Seeds: 42 (data, training), 0 (RND networks).

to detect the well-documented cost of narrow fine-tuning (McCloskey and Cohen, 1989; Kirkpatrick et al., 2017; Luo et al., 2023): a submission that turns the model into a pure game-player should pay here.

B Adaptation Settings

We adapt Qwen3.5-2B with LoRA; Table 3 lists the settings per regime. Three details are regime-specific. R2’s preference training converged to 0.88 accuracy with a reward margin of 1.90, the movement lying almost entirely on the rejected side (chosen -0.18 , rejected -2.07). R4 uses 201 preference pairs (107 on-policy, 94 hybrid) built from a greedy and a sampled ($t=0.7$) rollout pass. R5 samples at temperature 0.8 with top- p 0.95, 128 new tokens per turn and a 4096-token trajectory cap enforced during rollout, on the four single-player games (wordle, textmapworld, textmapworld_specificroom, adventuregame); the training interface provides no partner for two-player games. The RND networks are MLPs $256 \rightarrow 512 \rightarrow 128$ with predictor lr 10^{-4} and $\lambda=0.1$.

C Qualitative Examples

Qualitative example is shown in figures below, comparing the Base model (Figure 5) with the R2 model (Figure 6), on episode 2 of the game wordle.

D Reproducibility

1. Released checkpoints. All checkpoints are merged (no adapters) and public on the Hugging Face Hub under varadsrivastava/; each carries a model card describing its regime and results. Repository names retain

the internal identifiers used during development.

Regime	Repository suffix
R1 imitation	-sft
R2 outcome contrast [†]	-sft-dpo
R3 self-imitation	-iter3
R4 corrective feedback	-iter4
R5 GRPO (control)	-grpo-base-s42
R5 GRPO + RND	-grpo-rnd-s42

Table 4: Checkpoint mapping. All repositories share the prefix varadsrivastava/lm-playschool-qwen3.5-2b.

[†] = the model submitted to the challenge; the s42 suffix records the training seed. A further repository, -sft-dpo-vllm, republishes the R2 weights with a composite (vision-language) config.json for vLLM compatibility and is not a separate model.

2. Evaluation environment: Python 3.11 venv; clemcore pinned via the playpen package; clembench pinned requirements.
3. Registry entry: huggingface_local backend, premade chat template, end-of-sequence culling on the im_end token, left padding.
4. Data: colab-potsdam/playpen-data, HuggingFaceTB/smoltalk.
5. Frameworks: Transformers, PEFT, TRL (R1–R4); R5 uses our from-scratch GRPO implementation.
6. Official results: our submissions appear as DAIR__qwen3.5-2b__sft-v1 and DAIR__qwen3.5-2b__sft-dpo-v2 in the challenge results release, together with full transcripts of the held-out runs: <https://github.com/lm-playpen/lm-playschool-2026-final-results>

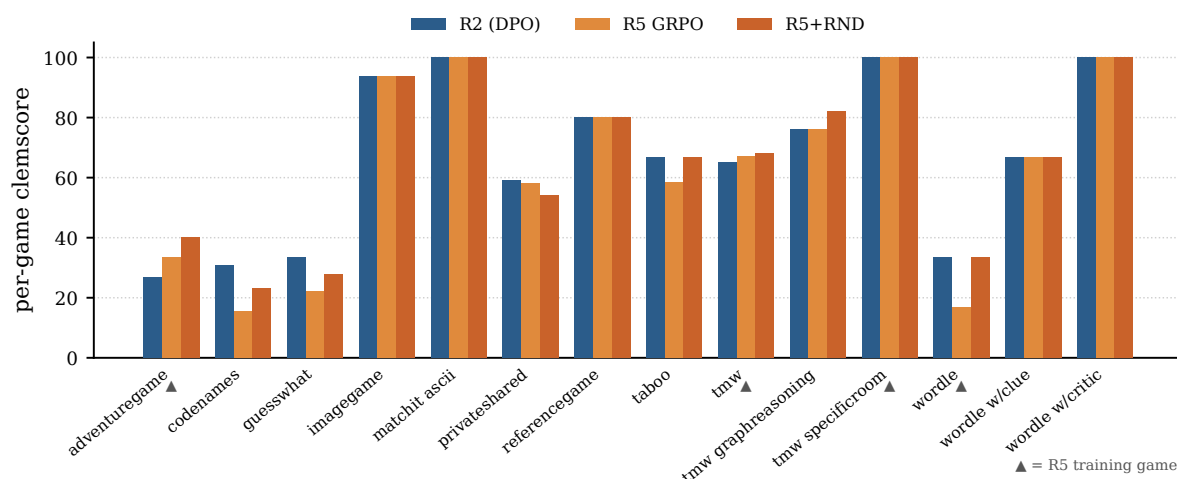


Figure 4: Per-game performance of R2, R5, and R5 + RND. ▲ are training-game markers referring to games R5 was trained on.

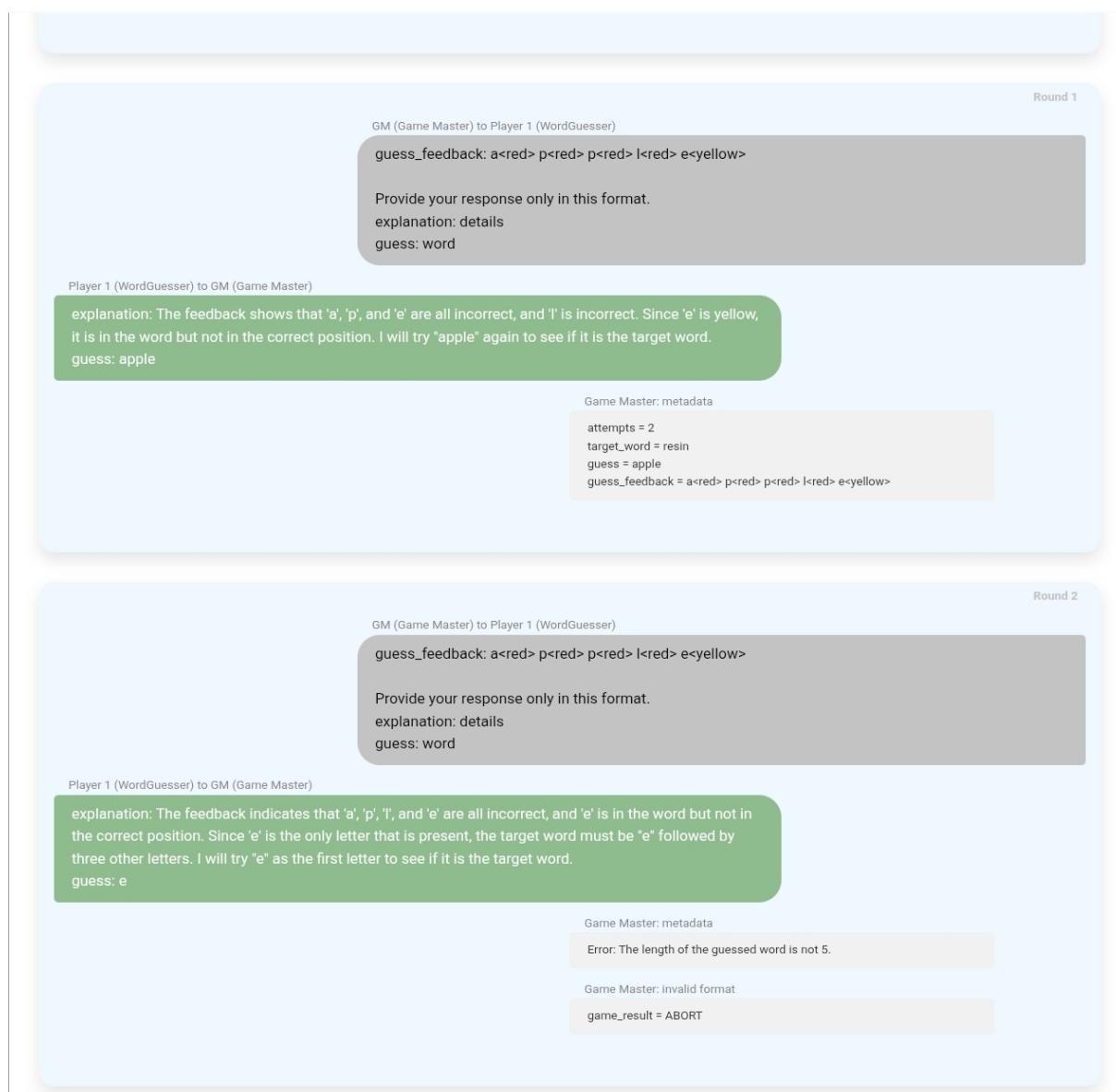


Figure 5: Example episode transcript before training.

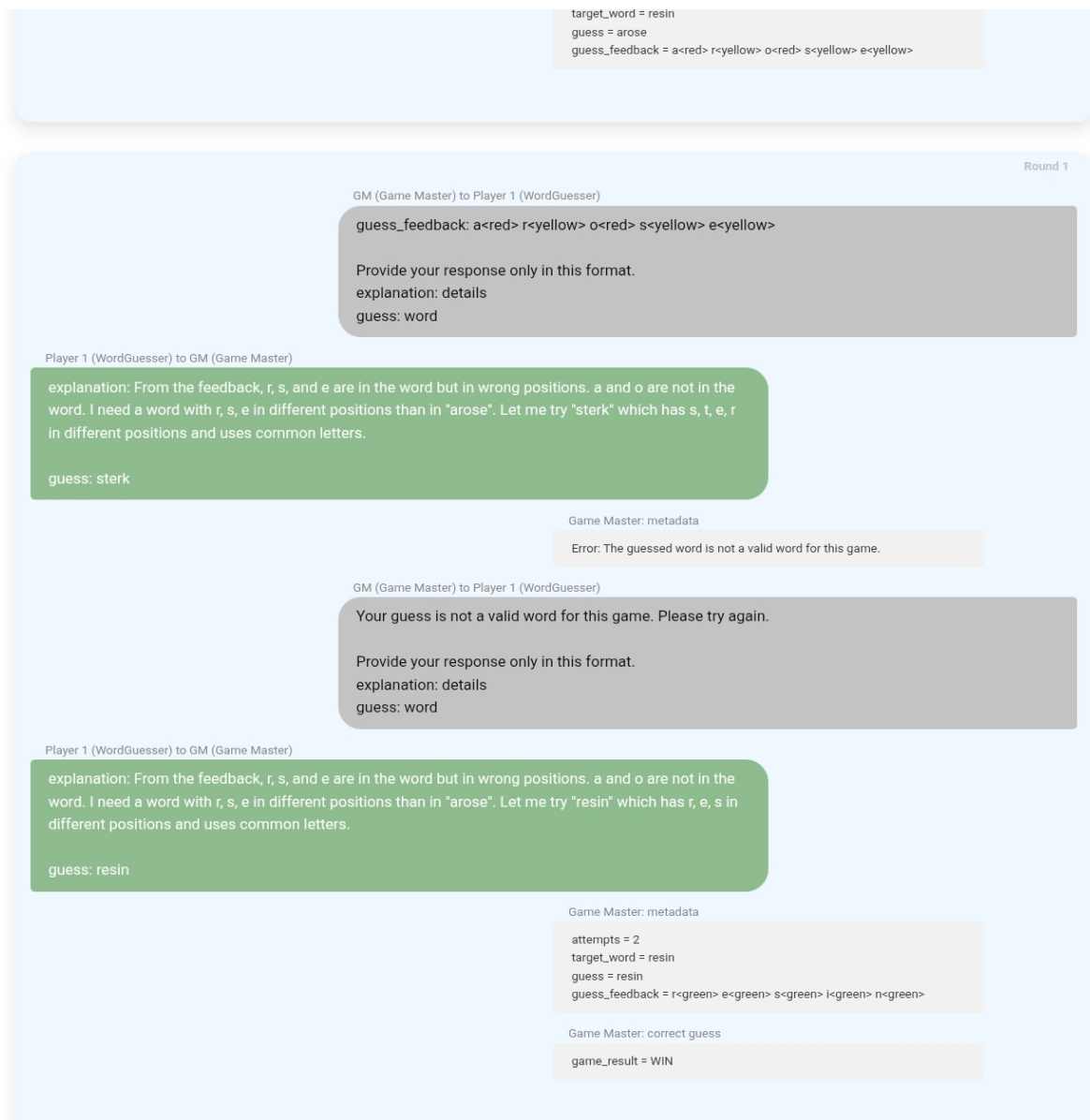


Figure 6: Example episode transcript after training (R2).

Teaching or Sharpening? An Exploration of the Potential of DPO Post-Training for Dialogue Games

Marika Sarzotti

University of Potsdam

marika.sarzotti@uni-potsdam.de

Simone Baratella

Independent Researcher

simone@baratella.biz

Abstract

Direct Preference Optimization (DPO) has emerged as a lightweight alternative to RLHF for post-training LLMs, yet it remains unclear whether it can teach novel skills or only sharpen existing ones. We investigate this question in the context of dialogue games, constructing multiple sets of on-policy preference pairs from the rollouts of supervised-fine-tuned Qwen3.5 models (2B and 9B) on the clembench benchmark, targeting rule adherence, strategic behaviour and excessive verbosity. Across all ablations, performance tracks a single variable, namely the distance of the chosen response from the policy’s own distribution. Pairs built from the model’s own winning moves raise the clemscore from 58.82 to 70.21 while pairs built around GPT-corrected moves land below the baseline at both scales. Transcript analysis points to a structural reason: bare preference pairs reduce conditional rules to unconditional preferences, so that DPO proves able to elicit successful behaviour already latent in the policy, but not to instill behaviour it does not yet exhibit.

1 Introduction

Dialogue Games, defined as "constructed activities driven by language use" (Schlangen, 2023), have been gathering increasing interest among the Natural Language Generation and Understanding research communities, as they can be used as gyms to train and test LLM-based agents’ interactive reasoning capabilities (Chalamalasetti et al., 2023; Horst et al., 2025).

Experimental findings suggest that passive exposure to text via self-supervised next-token-prediction over static corpora might not be a powerful enough learning signal to train models towards handling language in dialogue (Horst et al., 2025; Bisk et al., 2020; Laban et al., 2026; Sarzotti et al., 2025), and therefore various attempts have been carried out to explore the potential of

interaction-based (post-)training. Reinforcement-based approaches in particular have been tested towards this objective, as their "trial-feedback-error" pipelines appear to model the dynamicity of learning in interaction better than more static training regimes such as Supervised Fine-Tuning (Horst et al., 2025; Carta et al., 2023; Cheng et al., 2024). However, a commonly understood limitation of on-line reinforcement learning techniques is their cost in terms of computation and resources. Direct Preference Optimization (Rafailov et al., 2023) offers a valid alternative specifically to preference-based RL pipelines such as RLHF by approximating preference optimization to a binary contrastive classification problem.

In this landscape, with the present work we aim to test the potential of DPO as a post-training approach in the specific context of dialogue games, in particular asking if a debate originally developed around reinforcement learning — namely, whether such methods are able to teach novel skills to the model (Chu et al., 2025; Liu et al., 2025) or whether they are better suited for sharpening competences already acquired through previous stages (Huang et al., 2025; Gandhi et al., 2025; Yue et al., 2025; Zhao et al., 2025) — extends to DPO as well.

To this end, multiple sets of on-policy preference pairs for training were constructed starting from the supervised-fine-tuned models’ own rollouts on the full clembench benchmark (Chalamalasetti et al., 2023), incrementally tackling combinations of three behaviours that can significantly impact performance in dialogue games, namely the more fine-grained skills of rule-following and strategic thinking, along with general verbosity-driven rambling and looping.

Ablation studies revealed that the best performing model was the one solely optimized against excessive verbosity, which increased the benchmark-specific clemscore from 58.82 to 70.21 w.r.t. the supervised-fine-tuned baseline, nudging our find-

ings towards the claim that preference-based post-training approaches are the most helpful when it comes to sharpening already acquired tendencies. We thus present a technical breakdown of our best model — submitted to the LM Playschool Challenge — alongside an evaluation of how the different training ablations impacted the behaviour and performance of models of two varying sizes, precisely Qwen3.5-2B and Qwen3.5-9B (Qwen Team, 2026).

2 Methods

2.1 Base Model and Adaptation

As starting points for the present study we used a Qwen3.5-9B model and Qwen3.5-2B model. Both are instruction-tuned reasoning models, which we loaded in non-thinking mode to avoid an explosion in compute and wall-clock time at inference. We chose to use models of the Qwen family because they rank among the top open-weight models on the clembench benchmark¹ (Chalamalasetti et al., 2023), and specifically selected the two above-mentioned models for ablation comparison, to analyze how the impact of the proposed methodology scales with number of parameters.

The models were trained following a two stage pipeline, which involved a first phase of supervised fine-tuning training and a subsequent DPO one.

2.2 SFT training phase

SFT was conducted by training LoRA adapters (Hu et al., 2022) on the successful game rounds in the playpen dataset² (Horst et al., 2025) via assistant-only language-modeling loss. The learnt adapters were applied to all linear layers of the base models and merged with them to obtain ready-made models to use in the following stage.

2.3 DPO training phase

DPO fine-tuning made use of on-policy preference pairs constructed starting from the supervised-fine-tuned models’ own rollouts on the full clembench benchmark. Apart from data curation, all the other training details remain constant across conditions. During the DPO stage, LoRA adapters were trained

with the standard sigmoid loss (Rafailov et al., 2023). All training hyperparameters are reported in Table 1. Full replication instructions for the best model are provided in our [GitHub repository](#).

The submitted model was trained on a single NVIDIA H100 (94 GB), with SFT taking 3 h 50 m of wall-clock time and DPO a further 25 m. For further details on the model, please consult the dedicated [model card](#) on Hugging Face. The full ablation study — 2 SFT runs, 2 rollout harvests, 14 DPO runs, and 18 evaluated checkpoints — ran as single-GPU jobs on NVIDIA A100 80 GB cards and one H100 94GB, for an estimated total of 25–35 GPU-hours.

Hyperparameter	SFT	DPO
Optimizer	AdamW	AdamW
Epochs	1	1
Learning rate	2e-4	5e-5
Effective batch size	16	8
LoRA rank / α	16 / 32	16 / 32
β	0.3	
Max sequence length	1024	2048

Table 1: Training hyperparameters for the SFT and DPO stages. All runs used seed 42.

2.4 Dataset

The SFT training phase was conducted using the successful game rounds in the interactions subset of the playpen dataset’s train split. Only the game rounds that were carried out successfully — i.e. resulting in the player model winning — were retained for training, for a total of 20,202 game rounds.

For the DPO stage, on the other hand, we started by collecting on-policy rollouts of the supervised-fine-tuned models on the whole clembench v2.0 benchmark. Rounds that are part of the validation subset of the playpen dataset — and that will therefore be used later to evaluate the models’ performances — were excluded. Such data was then curated in specific ways to construct preference pairs for three main training ablations, each one tackling a different skill or behaviour that can easily impact game-playing performance: adherence to formatting rules, strategic behaviour and excessive verbosity. The complete dataset of preference pairs is available at the following [Hugging Face page](#).

¹Consult the clembench leaderboard and the full list of games in the benchmark [on the dedicated website](#).

²Dataset available [on HuggingFace](#)

2.4.1 Rule adherence

In order to construct the dataset of preference pairs targeting rule adherence, the rollouts which ended in the game rounds being aborted were extracted. In the context of clembench, a round is interrupted and marked as aborted when the player model fails, in its output answer, to respect game-specific formatting rules.

For each on-policy rollout which encountered this outcome, a GPT-5.2 model was prompted to generate a better, rule following answer in place of the faulty one provided by the SFT baseline model.

The dataset of preference pairs was then constructed by using the whole game rounds' conversation history up until the rule breaking answer as prompt, the GPT-corrected move as "chosen" (positive) continuation and the one originally proposed by the player model as "rejected" (negative) continuation. We thus obtained a dataset comprising 42 pairs for the Qwen3.5-9B SFT model and 79 for the Qwen3.5-2B SFT one.

2.4.2 Strategic Behaviour

Preference pairs tackling strategic game-playing were constructed in a very similar way to those described above for rule adherence. This time, however, the on-policy rollouts used were those where game rounds were failed (completed without rule violations, but not won).

The same GPT-5.2 model was prompted to identify, among the conversation history of each failed round, the first instance where the SFT model made a strategically poor move, and suggest a better one. Navigation-based partially observable games — namely games in the textmapworld family and adventuregame — were passed to GPT separately from all the others and under a different prompt. Given that the GPT model has access to the whole conversation history, it was necessary to clearly state in its prompt that any better move suggested had to be based solely on the information that could have been available to the player at that specific time in the game. We obtained 286 pairs for the 9B model and 325 for the 2B one.

2.4.3 Anti-verbosity

Finally, we constructed a set of preference pairs to tackle the generic tendency of LLMs to produce overly-verbose text and ramble in their outputs.

After extracting the successful rollouts of the SFT model, we sampled up to two assistant messages each and constructed the preference pairs with

Condition	9B		2B	
	clem	stat	clem	stat
Qwen3.5 (no fine-tuning)	41.92	54.16	13.05	44.02
SFT baseline	58.82	61.09	38.05	42.88
Anti-verbosity (AV)	70.21	58.73	44.69	37.38
Chosen-only (ctrl.)	65.36	59.35	46.41	44.82
Rule adher. (AB)	60.96	56.82	54.28	42.11
Strategy (F)	44.43	61.55	25.26	42.59
AV+AB	63.69	60.28	42.50	40.79
AB+F	43.45	61.58	28.81	39.81
AV+F	42.71	58.54	24.94	39.57
All three	26.92	43.55	29.45	33.02

Table 2: clemscore and statscore on the playpen validation split for the 9B and 2B models. The first two rows are reference points, namely the untuned Qwen3.5 models and the SFT baselines from which all conditions start. Following rows are ordered by the distance of the chosen response from the policy's own distribution: own winning move (AV, chosen-only), minimally edited own move (AB), GPT-written move (F).

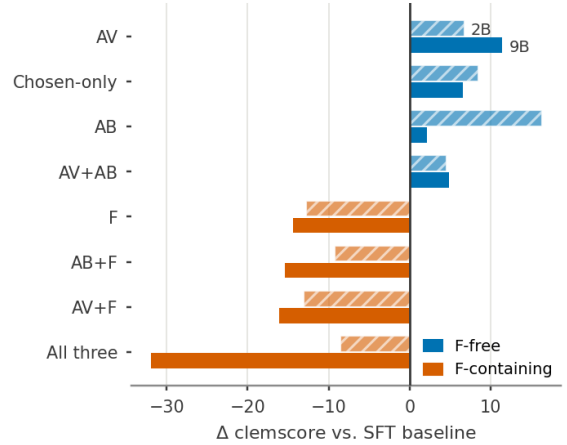


Figure 1: Clemscore difference vs. each scale's SFT baseline for all conditions, ordered by the distance of the chosen response from the policy (hatched bars = 2B).

the player model's own move as "chosen" and, as "rejected", the same move with one of five hand-written verbose continuations of empty prose appended to it. We thus obtained 995 pairs for the 9B model and 770 pairs for the 2B one.

Dataset imbalances are due to the availability of suitable rollouts being strictly dependent on how many rounds played by the two models fell under each of the three outcomes.

3 Results

All conditions are evaluated on the same 68 clembench instances of the playpen validation split, decoded at temperature zero, so all comparisons in this section are paired per instance. We report clem-

score, the harmonized clembench metric obtained as the product of %Played, which measures the share of episodes completed without formatting violations, and Quality, the average game-specific score on completed episodes. Alongside it we report statscore, the analogous score on playpen’s static benchmark suite, as a check on general capabilities³. Throughout this section we abbreviate the three conditions of Section 2.4 as AB (rule adherence, built from aborted rounds), F (strategic behaviour, built from failed rounds) and AV (anti-verbosity).

As a first reference point, Table 2 reports the results obtained by the untuned 2B and 9B models. The SFT training stage alone provides the largest single improvement observed in this study — +16.9 clemscore at 9B and +25.0 at 2B over the untuned models — and the resulting supervised-fine-tuned models serve as the baseline against which all DPO conditions are compared throughout this section.

3.1 Distance from model’s policy affects performance

As Table 2 shows, performance decreases monotonically with the distance of the chosen (positive) response in the preference pairs from the policy’s own distribution (Figure 1). Considering the 9B models’ clemscore differences over the SFT baseline, the model’s own winning moves (AV) yield +11.4, minimally edited own moves (AB) +2.1, GPT-written moves (F) −14.4, and every condition containing the F pairs lands below the baseline at both scales.

Statscores, on the other hand, were largely preserved across conditions and sizes, with only the three-way mixture showing substantial degradation, suggesting that game-targeted training left general capabilities mostly untouched.

3.2 Interferences from mixtures of conditions

Mixtures of conditions consistently underperform the sum of their parts, as additivity would predict +13.5 for AV+AB where +4.9 is observed, and −0.9 for the three-way mixture where −31.9 is observed — a gap of 8.7 and 31.0 points respectively. This interference pattern generalizes across scales, as at 2B combining the two individually best conditions (AB +16.2, AV +6.6) yields +4.5 against

an additive prediction of +22.9, leaving AV+AB below both of its ingredients. Full table and visualization in Appendix A. Finally, while the best performing model flips with scale, with AV first at 9B and AB first at 2B, the structure of the grid is preserved exactly, as every F-free condition improves on the baseline, every F-containing one lands below it, and the chosen-only control ranks second at both scales.

3.3 Effects of anti-verbosity training

Against the baseline, AV at 9B gained 17 wins and lost 2 (net +15, exact McNemar $p = 0.0007$), which makes it the only condition with a statistically significant gain. 15 of the 17 gained wins are conversions of failed episodes into successes, the number of aborted episodes is unchanged, and the entire lift accordingly sits in Quality (+12.0) rather than %Played (+1.0).

Notably, the verbosity that the AV pairs were designed to suppress was not reduced, as the model produces 14.7 words per move against the baseline’s 12.5, with a median paired delta of exactly zero, and its preambles and format-violation rates are identical to the baseline, while Quality on played episodes rose by 15 points (Wilcoxon $p = 0.0014$). This dissociation between unchanged length and improved play replicates at 2B (Quality +15.5, $p = 0.005$).

The chosen-only control recovers most of the gain at 9B (65.36, net +8 wins) and all of it at 2B, where it surpasses AV itself (46.41 vs. 44.69), though not the scale’s best condition, namely AB.

3.4 Game-specific effects of training

Gains and damage are localized by game structure. Word games with strict answer formats — namely the wordle family and taboo — respond best to AB at both scales, reference games — namely, referencegame, imagegame and matchit — are largely insensitive to every condition, and dialogue-heavier games — guesswhat, codenames and privateshared — being the hardest group at both scales, are reliably improved by no manipulation.

Exploration games, namely the navigation-based partially observable games listed in Section 2.4, carry most of the variance in the grid, as they gain +28 to +35 under every F-free condition and drop to nearly zero under every F-containing one, even though most of the strategy pairs were harvested from these very games.

The AV gain in this group is clearly behavioural and

³Playpen’s static benchmark suite contains 430 instances sampled across an array of five "static" — i.e. not involving dialogue and interaction — conventional NLP benchmarks, namely BBH, CLadder, EQ-Bench, IFEval and MMLU-Pro.

driven by persistence, since at 9B the SFT model quits almost immediately (a mean of 2.0 moves per episode in textmapworld, typically a single move followed by DONE), whereas the AV model plays an average of 8.8 moves per episode and converts all five textmapworld instances into wins.

The failure of the F pairs with respect to this kind of game has a legible mechanism that drags performance in the opposite direction to that of the AV pairs.

The most frequent correction in the F condition rejects a premature DONE, but since the pairs carry only bare moves (Section 2.4), no rationale is included and DONE never appears as chosen, so that the pairs uniformly penalize stopping — not only when triggered too soon in the round — and never reward it.

In the textmapworld family, the F model produces DONE once in the whole set (0.5% of moves). It explores 19 moves per episode and aborts 10 out of 11 episodes at the turn limit, even though its moves remain perfectly well-formed. At 2B the effect is total, with zero DONE across 220 moves and 11 out of 11 episodes aborted. By contrast, the AV model, trained on pairs containing chosen moves drawn from complete winning episodes — and therefore including well-timed final DONEs — uses the command in 13.9% of its moves and solves all 11 episodes. Per-game breakdowns are reported in Appendix B.

4 Discussion

Our work can be positioned within the debate on the potential of reinforcement- and preference-based post-training approaches, by asking whether they are more suited to sharpen capabilities already encoded in the model’s own distribution (Huang et al., 2025; Gandhi et al., 2025; Yue et al., 2025; Zhao et al., 2025) or to teach new ones (Chu et al., 2025; Liu et al., 2025). We call *elicitation* the raising of the probability of behaviour the model already produces, and *instillation* the attempt to inject behaviour it does not.

4.1 Elicitation

Our results (Table 2) are explained by a single variable, namely how far the chosen response sits from the policy’s own distribution. This aligns with the difficulties reported by Horst et al. (2025), whose DPO conditions — built from off-policy pairs drawn from a static corpus of other models’

interactions — failed to improve over their SFT baseline, and extends the known advantage of on-policy preference data (Tajwar et al., 2024) to a graded phenomenon, as in our grid performance degrades with every step the chosen response takes away from the policy’s own outputs.

The AV pairs represent pure elicitation, as both sides of each pair share almost identical content. Moreover, the appended padding is text the model essentially never generates, and the 9B model is not, on its own, verbose enough for it to affect game-playing. The extra text in the rejected moves thus does not act as a suppressing signal so much as a signal to produce more of the model’s own successful behaviour.

The gain is quality-driven and length-neutral, it largely survives without the preference contrast — as the chosen-only control shows — and it amplifies whatever behaviour each game’s winning rounds happen to instantiate, such as persistence in exploration or safe guessing in codenames. Where the SFT model never won in the harvested rollouts, as in plain wordle, no pairs exist to display proficient behaviour, and so nothing can be elicited and nothing moves performance-wise.

4.2 Instillation

Instillation, on the other hand, did not merely fail, as every F-containing condition landed below the SFT baseline at both scales, and this outcome has a structural reason. Such motivation cannot be attributed to corrections’ quality, as GPT’s suggested moves verify as overwhelmingly legal and feedback-consistent against game ground truth (Appendix C).

Rather, a correction such as “stopping is wrong while unvisited exits remain” expresses a conditional rule, but preference pairs composed of bare moves have no channel for such rationale — as they can directly express *what* was preferred but not *why*. While they can provide indirect information through the round’s chat history, which encodes a signal about *when* a specific choice should be preferred, that signal was not strong enough as to carry information regarding the underlying reason. The model thus learnt an unconditional rather than context-dependent preference in navigation-based games at both scales, namely that “stopping is bad”. The resulting damage is thus a corrupted decision rather than a malformed output.

Consistent with this, F’s statscore is preserved, indicating a bad game-specific strategy rather than

degraded general capabilities. This instillation-oriented regime is also the one in which DPO’s gradients act at their most destructive, since they simultaneously suppress a high-probability behaviour of the model and inflate text it cannot yet produce, the setting where likelihood displacement is known to arise (Razin et al., 2025).

4.3 Model size effects

The fact that the best-performing condition changes with model size follows from the same account, since the type of pairs that can increase scores the most depends on where the model’s headroom lies. The 9B rarely breaks formatting rules, so its headroom is in the quality of play and self-imitation wins, while the 2B aborts frequently, so minimally edited corrections win — although even there most of the gained wins are converted losses rather than rescued aborts. Both scales therefore improve in the same way, namely by playing better on the episodes they already play to completion.

5 Conclusion

Returning to the question in our title, our findings show that DPO was not able to teach any of the three skills it was aimed at, corroborating the idea that such method is not the best suited for instilling novel capabilities into a model’s distribution.

On the other hand, the preference-based training was able to elicit, wholesale, the successful behaviour latent in the policy, sharpening the models’ already acquired skills in handling different game types, and bringing substantial gains in the final scores.

The best model in this study thus came from letting the model be its own teacher, with DPO functioning as an amplifier on the policy’s own successes, and as a liability the moment it was pointed at anything else.

Limitations

Evaluation uses 68 validation instances at temperature zero. Paired tests are exact, but per-game cells hold 3 to 13 episodes, and powering per-game claims would require running the models on the whole clembench benchmark. For this work specifically, our compute budget prioritized the breadth of the ablations grid over replication depth. Therefore, each condition is a single training run, and multi-seed replication of the ablations would be needed to gain statistical soundness. Temperature

zero decoding leaves no within-condition variance. For future work, sampled evaluation would be necessary to conduct pass@k-style tests of whether the amplified wins already exist in the SFT policy’s samples. Significance is reported uncorrected. The AV result survives Bonferroni correction, the secondary ones do not. Results cover one model family, and the chosen-only control exists only for the AV condition, leaving the dispensability of the preference contrast untested for the other sets of pairs. Corrections were produced by a single judge model, and our validation certifies their legality and feedback-consistency, but not their strategic optimality.

References

- Yonatan Bisk, Ari Holtzman, Jesse Thomason, Jacob Andreas, Yoshua Bengio, Joyce Chai, Mirella Lapata, Angeliki Lazaridou, Jonathan May, Aleksandr Nisnevich, Nicolas Pinto, and Joseph Turian. 2020. [Experience Grounds Language](#). In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8718–8735. Online. Association for Computational Linguistics.
- Thomas Carta, Clément Romic, Thomas Wolf, Sylvain Lamprier, Olivier Sigaud, and Pierre-Yves Oudeyer. 2023. [Grounding Large Language Models in Interactive Environments with Online Reinforcement Learning](#). In *Proceedings of the 40th International Conference on Machine Learning*, pages 3676–3713. PMLR.
- Kranti Chalamalasetti, Jana Götze, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2023. [clembench: Using Game Play to Evaluate Chat-Optimized Language Models as Conversational Agents](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11174–11219, Singapore. Association for Computational Linguistics.
- Pengyu Cheng, Yong Dai, Tianhao Hu, Han Xu, Zhisong Zhang, Lei Han, Nan Du, and Xiaolong Li. 2024. [Self-playing adversarial language game enhances LLM reasoning](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 126515–126543. Curran Associates, Inc.
- Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. 2025. [SFT memorizes, RL generalizes: A comparative study of foundation model post-training](#). In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 10818–10838. PMLR.
- Kanishk Gandhi, Ayush K Chakravarthy, Anikait Singh, Nathan Lile, and Noah Goodman. 2025. [Cognitive](#)

- behaviors that enable self-improving reasoners, or, four habits of highly effective STars. In *Second Conference on Language Modeling*.
- Nicola Horst, Davide Mazzaccara, Antonia Schmidt, Michael Sullivan, Filippo Momentè, Luca Franceschetti, Philipp Sadler, Sherzod Hakimov, Alberto Testoni, Raffaella Bernardi, Raquel Fernández, Alexander Koller, Oliver Lemon, David Schlangen, Mario Giulianelli, and Alessandro Suglia. 2025. [Playpen: An Environment for Exploring Learning From Dialogue Game Feedback](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29854–29891, Suzhou, China. Association for Computational Linguistics.
- Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Audrey Huang, Adam Block, Dylan J Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jordan T. Ash, and Akshay Krishnamurthy. 2025. [Self-improvement in language models: The sharpening mechanism](#). In *The Thirteenth International Conference on Learning Representations*.
- Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2026. [LLMs get lost in multi-turn conversation](#). In *The Fourteenth International Conference on Learning Representations*.
- Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. 2025. [ProRL: Prolonged reinforcement learning expands reasoning boundaries in large language models](#). In *Advances in Neural Information Processing Systems*, volume 38, pages 17998–18031. Curran Associates, Inc.
- Qwen Team. 2026. [Qwen3.5: Towards native multi-modal agents](#).
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.
- Noam Razin, Sadhika Malladi, Adithya Bhaskar, Danqi Chen, Sanjeev Arora, and Boris Hanin. 2025. [Unintentional unalignment: Likelihood displacement in direct preference optimization](#). In *The Thirteenth International Conference on Learning Representations*.
- Marika Sarzotti, Giovanni Duca, Chris Madge, Raffaella Bernardi, and Massimo Poesio. 2025. [MLLMs Construction Company: Investigating Multimodal LLMs’ Communicative Skills in a Collaborative Building Task](#). In *Proceedings of the Eleventh Italian Conference on Computational Linguistics (CLiC-it 2025)*, pages 1037–1047, Cagliari, Italy. CEUR Workshop Proceedings.
- David Schlangen. 2023. [Dialogue Games for Benchmarking Language Understanding: Motivation, Taxonomy, Strategy](#). *arXiv preprint*. ArXiv:2304.07007 [cs.CL].
- Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. 2024. [Preference fine-tuning of LLMs should leverage suboptimal, on-policy data](#). In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 47441–47474. PMLR.
- Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025. [Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model?](#) In *Advances in Neural Information Processing Systems*, volume 38, pages 57654–57689. Curran Associates, Inc.
- Rosie Zhao, Alexandru Meterez, Sham M. Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach. 2025. [Echo chamber: RL post-training amplifies behaviors learned in pretraining](#). In *Second Conference on Language Modeling*.

Appendix

In the following, we report additional details regarding the present work. Specifically, Section A shows the full table with clemscore Δ s as predicted by additivity and the ones we actually obtained, along with a visualization, Section B presents a per-game breakdown of our results and Section C outlines further information on how the GPT corrections were validated.

A Additivity Table

Table 3 reports, for every pair-type mixture at both scales, the observed clemscore Δ s against the additive prediction obtained by summing Δ s of the single condition constituents, while Figure 2 visualizes the same relationship. The gap is negative in all eight cases, and at 2B two mixtures with positive additive predictions (AB+F and All three) still land below the baseline, showing that interference can reverse the sign of predicted benefits rather than merely shrink them.

Scale	Mixture	Predicted	Observed	Gap
9B	AV+AB	+13.5	+4.9	-8.7
9B	AB+F	-12.3	-15.4	-3.1
9B	AV+F	-3.0	-16.1	-13.1
9B	All three	-0.9	-31.9	-31.0
2B	AV+AB	+22.9	+4.5	-18.4
2B	AB+F	+3.4	-9.2	-12.7
2B	AV+F	-6.2	-13.1	-7.0
2B	All three	+10.1	-8.6	-18.7

Table 3: Observed clemscore Δ s vs. the SFT baseline for every pair-type mixture, against the additive prediction (sum of the constituent single conditions’ Δ s). The gap (observed – predicted) is negative in all eight cases. At 2B, two mixtures (AB+F, all three) have positive additive predictions yet land below the baseline.

B Per-Game Breakdown of Results

Figures 3 and 4 show heatmaps of the distributions of clemscore Δ s with respect to the SFT baselines for the 2B and 9B models respectively. It is possible to appreciate how, across sizes, overall performance of the post-trained models decreases the moment F-pairs enter the training data. As stated in Section 3, for the 2B model most of the gains were observed in the AB condition, while the training that the 9B one benefitted from the most was AV. Moreover, in the case of the 9B model, the heatmap shows how the games that post-training (except for when F-pairs were introduced) helped the most were the navigation-based ones, then, less

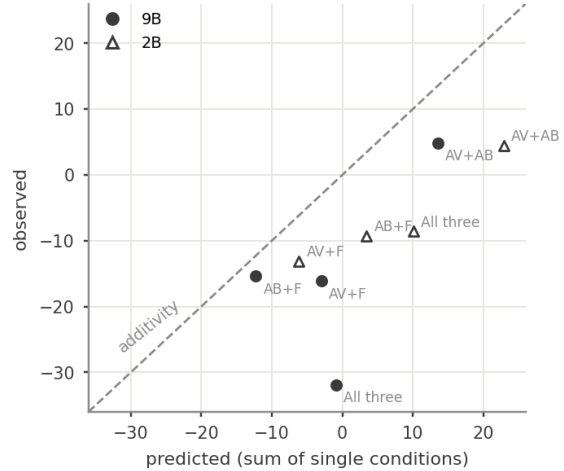


Figure 2: Observed clemscore Δ s (vs. the SFT baseline) against the additive prediction for every pair-type mixture (filled circles: 9B; open triangles: 2B). Points on the dashed identity line would indicate that pair types combine additively. All eight mixtures fall below it.

consistently, the text games with strict formatting rules, then reference games — where hardly any gain is observed, potentially because performance was already very high, even in the untuned model ⁴ — and finally dialogue-heavier games, arguably the hardest family since performance was steadily below that of the other kinds of games, even under the most beneficial ablations.

C Validation of GPT-Generated Corrections

As outlined in Section 2.4, the chosen (positive) responses in AB and F preference pairs for DPO training consist of bare corrected moves generated by a GPT-5.2 model.

In order to test the potential presence of confounds driven by poor correction quality — especially as a possible cause for the F conditions’ damage to task performance — we validated every correction in both pair sets and at both scales against game ground truth, namely the objective state recoverable from the pair’s own prompt — i.e. the dialogue history up to the correction point — such as the letter feedback revealed so far in wordle or the exits listed for the current room in textmapworld. Each correction receives a verdict as either valid, invalid or unverifiable, where the latter marks cases

⁴Given %Played-Quality Score pairs, for the games in the reference games group — namely imagegame, matchit and referencegame — the untuned model achieved 75.0-88.33, 100-100, 100-80 respectively, while the SFT one scored 100-88.5, 100-100, 80-100.

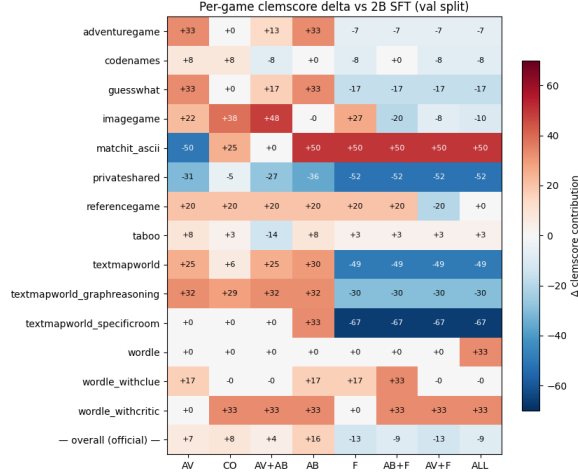


Figure 3: Per-game clemscore Δ s w.r.t. the 2B SFT baseline on the playpen validation split. Rows represent games, while columns are training conditions (CO = chosen-only control), and the bottom row reports the official overall clemscore Δ s. Redder cells indicate gains over the baseline, bluer ones losses.

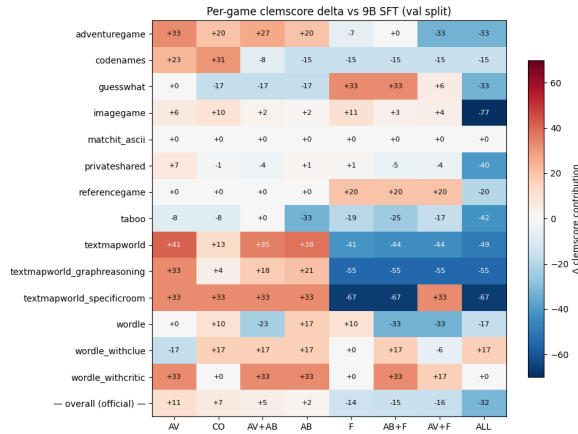


Figure 4: Per-game clemscore Δ s w.r.t. the 9B SFT baseline on the playpen validation split. Rows represent games, while columns are training conditions (CO = chosen-only control), and the bottom row reports the official overall clemscore Δ s. Redder cells indicate gains over the baseline, bluer ones losses.

in which no move was proposed at all, hence there is no material to score.

Note that this evaluation is based on objective, machine-readable ground truth, so it allows to test legality, rule compliance, feedback consistency and overall soundness of moves, but not strategic optimality. For this same reason, the depth of each check is dictated by what the game makes mechanically decidable. Specifically, the wordle variants, codenames and textmapworld allowed for a more precise evaluation, since they expose formal ground truth through the feedback players receive from the programmatic game master at each turn, while for all the other game families corrections evaluation could rely solely on adherence to the required answer format. The checks conducted by the per-game validator thus were:

- **Feedback consistency (wordle family):** the suggested guess must be a well-formed five-letter word that keeps all letters confirmed green, avoids letters known absent, and does not replay a yellow letter in the same position, given the feedback revealed before the correction point.
- **Board rules (codenames):** for clue-giver corrections, the suggested clue must be a single word absent from the board and the stated targets must belong to the team’s words, while for guesser corrections, each suggested word must appear in the candidate list, respect the stated guess limit, and differ from the clue word.
- **Move legality (textmapworld family):** the suggested direction must be among the exits listed for the current room.
- **Format conformance (remaining games):** the correction must match the game’s required answer format, specified in its rules.

In addition to our strictly programmatic validation, we note that GPT-5.2 achieved a clemscore of 84.19 at high effort, indicating how the model is an overall capable player beyond the dimensions on which its corrections were evaluated⁵.

The results of our validation, outlined in Table 4, qualify the corrections as reliable, as both kinds of pairs receive near-ceiling scores across sizes, with the lowest percentage of valid corrections —

⁵Refer to [clembench’s official website](#) for the model’s full results on the benchmark

Pair set	n	valid	invalid	unverif.
9B failed	286	99.0%	0.7%	0.3%
9B aborted	42	85.7%	14.3%	0.0%
2B failed	325	98.8%	1.2%	0.0%
2B aborted	79	92.4%	7.6%	0.0%

Table 4: Validation scores percentages for the GPT-generated corrections, by pair set and scale. A single correction — a wordle one containing no suggested move — could not be decided against ground truth and was thus scored as unverifiable.

namely, 85.7% — belonging to the AB pairs for the 9B model.

We signal how these scores argue against correction quality as the source of the F sets’ damage, as the corrections for failed rounds achieved the highest percentages of positive scores for both the 9B and 2B training sets (99.0% and 98.8% of valid corrections, respectively). In the textmapworld family, the largest source of per-game losses in the F pairs, every 9B suggestion is legal (86 of 86), and 85 out of 86 are valid at 2B. Moreover, validity and damage to performance dissociate across pair sets, as the aborted corrections are markedly less valid (85.7% at 9B, 92.4% at 2B), yet the aborted sets did not produce the regressions that the near-perfect failed sets did.

The same dissociation holds within the failed sets across games, as per-game correction validity does not correlate with per-game score change (Spearman $\rho = -0.24$, $p = 0.43$, $n = 13$).

Acquire, Repair, Preserve: A Diagnosis-Guided Post-Training Recipe for Small-Model Dialogue Game Agents

Nan Li

Department of Information and Computing Sciences,
Utrecht University, Utrecht, The Netherlands
n.li@uu.nl

Abstract

Interactive dialogue games test a capability that static benchmarks largely leave implicit: a model must carry state across turns, interpret feedback, and choose valid actions under changing constraints. We study this setting in the LM Playschool Challenge with a 2B open-weight model, and find that many failures are not only broad knowledge failures but also local decision failures: repeated guesses, malformed actions, and violations of feedback that the model has just seen. These diagnostics motivate a training recipe organized around three steps: acquire broad game participation through supervised fine-tuning, repair mechanically verifiable failures within one targeted dialogue-game family using turn-local preference pairs, and preserve general capabilities beyond these dialogue games. In the official final evaluation, our submission improves public clemscore from 10.67 to 38.92 and closed in-domain score from 13.41 to 41.17, while approximately preserving aggregate static performance (44.14 vs. 44.24 for the baseline). Out-of-domain clemscore remains low at 7.88, with the largest gains concentrated in unseen variants of the targeted family. Our results suggest that broad SFT brings most of the model’s capability improvement; turn-local supervision can be effective when failure detection is precise, with observed transfer concentrated primarily within-family.

1 Introduction

Dialogue games evaluate language models as agents whose next utterance changes the state they must later reason over. A word guess rules out letters; a clue constrains a partner’s action; a navigation command changes the room from which the next command is interpreted. The clembench benchmark and the Playpen learning environment (Chalamalasetti et al., 2023; Horst et al., 2025) make these dependencies explicit: success requires not only producing plausible language, but also

Dialogue state	Model action	Local failure
After crane: red c/r/n; guess: crane yellow a; green e		repeats guess
(same)	guess: abid	wrong length
(same)	guess: cable	uses red c

Table 1: Simplified Wordle failure states. The model output can be syntactically close to valid while violating a constraint entailed by the immediately preceding feedback. A real transcript exhibiting this pattern is reproduced in Table 7.

maintaining state, interpreting feedback, and choosing a valid action under accumulated constraints.

Playpen frames this setting as learning from Dialogue Game Feedback (Horst et al., 2025): programmatic, goal-directed feedback over trajectories, without relying on a learned judge over isolated responses. The feedback does not itself locate where an interaction stops making progress; interactive failure diagnostics are needed to construct turn-local preference pairs for reinforcement learning. The question for a dialogue game agent is therefore not only whether dialogue game data helps, but which part of the trajectory should receive supervision.

Small models are a natural target for this setting. They are cheaper to serve, better suited to low-latency interaction, and more plausible for local assistants or assistants with limited resources, but their interactive behavior is brittle. In the LM Playschool Challenge, the official Qwen3.5-2B baseline (Qwen Team, 2026) scores 10.67 clemscore in the final evaluation, far below the 60.30 score of its 27B sibling.¹ The challenge also reports a static benchmark score, so a useful small model recipe must improve interaction without sacrificing general capability.

Inspecting the 2B model’s transcripts suggested

¹Official final evaluation results: <https://github.com/lm-playpen/lm-playschool-2026-final-results>.

that the gap was not mainly a lack of game knowledge. In Wordle, the model often preserves the expected answer format while failing at the next local decision (Table 1). These errors are frequent, mechanically checkable, and localized to a specific decision made after receiving feedback. They point to a different training question: not how to teach the model the whole dialogue again, but where to place supervision so that it reaches the particular decision that fails.

This paper describes the playornotplay team submission built around that question. We first use broad imitation to make the model participate in games, then diagnose the remaining failures, repair those of one weak and mechanically diagnosable game family (Wordle) with turn-local preference pairs, and finally scale the learned update to preserve general capabilities, as measured by aggregate static performance. In the official final evaluation, the submitted model reaches 38.92 public clemscore and 41.17 closed in-domain score, above the official 4B and 9B baselines, while approximately preserving aggregate static performance over the 2B baseline (44.14 vs. 44.24).

The main contribution is an empirical account of supervision placement for small interactive agents. A small number of preference pairs anchored to real turns after feedback produces the clearest positive preference-tuning signal in our pipeline, whereas whole-dialogue preference tuning destroyed protocol compliance in our experiments. Follow-up experiments further show that the usefulness of an update depends on the stage and parent to which it is applied. Together, these results locate turn-local preference learning within a broader progression from acquisition to repair and model selection.

2 Task and Evaluation Setting

The LM Playschool Challenge evaluates whether learning in conversational interaction improves agentic capability. Playpen is the learning environment for Dialogue Game Feedback (Hurst et al., 2025): it organizes rollouts, training data, and on-line sampling over dialogue game trajectories. The dialogue games are implemented in clembench (Chalamalasetti et al., 2023), where programmatic game masters mediate turns, track state, and enforce output protocols, while game-specific scorers compute Quality.

In the public validation setting used here (from

the playpen-data dataset²), the interactive suite contains 14 clembench games and 67 episodes in our pinned development run. We pin the public-development data to revision 557d8caf and the clembench game tree to commit ed39486. The official final evaluation comprises an expanded public suite (502 episodes), a closed in-domain suite (1,272 episodes), and a closed out-of-domain suite covering unseen game families (360 episodes)³. We use the public-development set for iteration and reserve the closed suites for final evaluation.

The LM Playschool Challenge uses Clemscore and Statscore as metrics. **Clemscore** combines two quantities for each game g : **Played**, the percentage of episodes not marked aborted by the evaluator, and **Quality**, the task score on played episodes. With $G_Q = \{g \in G : \text{Played}_g > 0\}$, we report it as

$$C = \left(\frac{1}{|G|} \sum_{g \in G} \frac{\text{Played}_g}{100} \right) \times \left(\frac{1}{|G_Q|} \sum_{g \in G_Q} \text{Quality}_g \right). \quad (1)$$

Quality is undefined for a game with no played episode, so the Quality macro-average runs over G_Q only. Thus a model can lose score either by failing to play or by playing poorly after it stays in the game. The playpen-data validation split contains 430 static-benchmark instances drawn from BBH (Suzgun et al., 2023), CLadder (Jin et al., 2023), EQ-Bench (Paech, 2023), IFEval (Zhou et al., 2023), and MMLU-Pro (Wang et al., 2024); applying the same aggregation yields **Statscore**.

The pinned public split contributes one or two episodes for each represented game–experiment combination, and this small sample is the main source of run-to-run variance: differences of a few points should be treated cautiously. We therefore use paired same-host comparisons for development decisions and report the official final evaluation as the primary result.

3 Our Method

Our recipe is a narrow instantiation of the training regimes enabled by Playpen: successful transcripts support SFT, paired alternatives support DPO,

²<https://huggingface.co/datasets/colab-potsdam/playpen-data>.

³<https://github.com/lm-playpen/lm-playschool-2026-closed-set>

Phase	Objective	LR / batch / len	Clem
—	Qwen3.5-2B	—	13.05
A. Acquire	broad SFT	2e-4 / 32 / 2048	43.85
B. Repair	turn-DPO: rpt + len	5e-6 / 16 / 1024	46.83
	turn-DPO: inv + dup	5e-6 / 16 / 1024	45.52
C. Refine	branch-DPO	5e-6 / 16 / 1024	48.52
D. Preserve	delta scaling, $s=0.85$	no gradient	50.43 [†]
	shipped fp32 merge	—	48.70

Table 2: Pipeline trajectory on the pinned public-development set (dataset 557d8caf, clembench ed39486). Batch denotes effective batch size; all Clem values are the full 14-game suite. Phases B and C train on Wordle data only. Both Phase B passes use $\beta=0.1$; Phase C uses $\beta=0.2$. All trainable phases use LoRA on Qwen3.5-2B, AdamW, bf16, and one epoch; Phase C branches the model (factor 2, rounds 1–3, $T=0.7$) against a frozen reference. rpt/len = repeated guesses and bad length; inv/dup = invalid words and duplicated answer fields. [†]Mean of two runs (50.82 and 50.03). Per-game and per-phase Wordle detail: Tables 13 and 14.

and fresh rollouts support online reward learning. Playpen itself includes turn-level DPO (Horst et al., 2025); our specific choice is to use game-specific checks to construct preference pairs at mechanically verifiable failures after feedback. We use broad SFT to make the model participate, then target those diagnosed decisions.

All training uses LoRA (Hu et al., 2022) ($r=16$, $\alpha=32$, all-linear, adapter dropout 0.05) on Qwen3.5-2B.⁴ Source trajectories and instances come from the train split of playpen-data; Phase B negatives are synthetic corruptions using the clembench word list; Phase C data is model-generated. Table 2 gives the trajectory on the pinned public-development set. Following the official clemscore definition in Eq. 1, broad imitation primarily improves Played; our later phases target Quality on states the model already plays; final delta scaling targets statscore rather than game behavior directly.⁵

⁴Qwen3.5-2B (Qwen Team, 2026) is a vision-language model; we fine-tune and serve it in text-only mode.

⁵The manuscript submitted for review described the training process for the model, which is also used in the official final evaluation, as six stages. Subsequent verification showed that two nominal stages did not update the model, so we omit them here. Appendix A.1 identifies the evaluated model revision; Appendix A.4 explains the two removed stages.

3.1 Diagnosing Small-Model Game Failures

Phase A performs SFT on success-only interactions from the training split of playpen-data, balanced to at most 700 examples per game (9,522 rows total at the reconstructed training-time snapshot; per-game counts in Table 6). This lifts the model from 13.05 to 43.85 on the pinned public-development set. The model has learned the surface form of game play, yet raw Wordle quality remains at zero: it still repeats rejected guesses and violates immediate feedback constraints. Wordle was not the only remaining weakness, but its failures were frequent, local to a single decision, and mechanically verifiable, so we selected it as the target of all subsequent optimization. Phases B and C use only Wordle data; every reported clemscore remains the full 14-game suite.

3.2 Turn-Local Repair

Given a diagnosed failure mode, we construct preference pairs that share the same dialogue state. For a history h , the rejected completion y^- exhibits one mechanical error and the preferred completion y^+ is a valid alternative from a clean successful episode. We then apply Direct Preference Optimization (DPO; Rafailov et al., 2023):

$$\mathcal{L}_{\text{DPO}} = -\mathbb{E} \log \sigma \left(\beta [r_{\theta}(h, y^+) - r_{\theta}(h, y^-)] \right), \quad (2)$$

with policy-reference log-ratio $r_{\theta}(h, y) = \log \frac{\pi_{\theta}(y|h)}{\pi_{\text{ref}}(y|h)}$. Holding h fixed is the point: the contrast lands on the few decision tokens that determine whether the model repeats a guess, uses the wrong length, produces an invalid word, or duplicates the answer field. This construction does not identify a single failure; it enumerates all eligible post-feedback turns in clean successful episodes and creates a synthetic negative at each one. Appendix A.3.3 provides the source-episode filters, turn-eligibility criteria, and the four corruption rules.

Phase B applies this in two passes: the first addresses repeated guesses and bad length (1,905 pairs, $\beta=0.1$); the second addresses invalid words and duplicated answer fields (1,906 pairs, $\beta=0.1$). After the first pass, the old repeat loops no longer appear in the raw Wordle probe, but invalid words and duplicated answer fields emerge instead. The second pass corrects these; its aggregate suite effect is sign-inconsistent across the two evaluation rails (Table 8).

After addressing fixed failure classes with synthetic pairs, we turn to errors arising in the current model’s own trajectories. Phase C plays training-split Wordle with the current checkpoint, branches at early decision points (factor 2, rounds 1–3, $T=0.7$), and pairs the immediate diverging responses from the highest- and lowest-scoring siblings. DPO against a frozen reference ($\beta=0.20$) targets the residual distribution gap. Branch-DPO follows a 3.0-point increase in the submitted-model trajectory, whereas the controlled single-host comparison shows a 2.07-point clemscore decrease and a 1.23-point statscore increase (Tables 2 and 8). We therefore report its contribution as evaluation-rail-dependent while retaining it as part of the submitted model’s training sequence. Appendix A.3.4 details instance selection, branch grouping, downstream scoring, and pair extraction.

3.3 Preserving General Capability

By this point, every strong interactive candidate pays a specialization tax: statscore drops several points from the starting checkpoint on our pinned development set. Rather than add more data or replay, we treat preservation as a post-training weight-space problem. We scale the learned delta, $W(s) = W_{\text{base}} + s \Delta_{\text{LoRA}}$, and select the scalar $s \in [0, 1]$ by constrained model selection,

$$\begin{aligned} s^* &= \arg \max_s S(W(s)) \\ \text{s.t. } C(W(s)) &\geq C_{\min}, \end{aligned} \quad (3)$$

where $C_{\min}$ is a paired interactive gate on the pinned public-development set. At $s^*=0.85$, the mean of two clemscore evaluations is 50.43 (Table 2). The operation is training-free and composes with any plain LoRA adapter, but the scalar must be selected on both axes: too low a value fails the interactive gate. In the official final evaluation, the submitted fp32 merge scores 38.92 public clemscore with statscore 44.14 against the 2B baseline’s 44.24.

4 Results

Table 3 reports the official final evaluation. Our submitted artifact⁶ improves public clemscore from 10.67 to 38.92 (+28.25) and closed in-domain

⁶The submitted checkpoint is the fp32-merged model `chnln/Qwen3.5-2B-playpen-playornotplay` at Hugging Face revision 828e356. Table 5 gives the complete training configuration; Appendix A.2 describes the evaluation protocols.

score from 13.41 to 41.17 (+27.76), both above the official 4B and 9B baselines. For comparison, scaling the official baseline from 2B to 9B raises public clemscore from 10.67 to 31.91 (+21.24) with no training at all. Aggregate static performance is approximately preserved: 44.14 against the baseline’s 44.24.

Table 4 decomposes the official endpoint into Played and Quality. The in-domain gain is driven by both components: Played nearly doubles while Quality also increases. Out-of-domain Played, however, falls below the untrained base (33.08 vs. 35.19), so the OOD clemscore gain from 3.72 to 7.88 reflects only conditional quality on fewer played episodes.

Of the total positive out-of-domain movement (84.86 effective score), three held-out Wordle-crazy configurations contribute 70.02, or 82.5%; the remainder comes from TA Frozen Lake (+9.17), TA Mastermind (+5.00), and Cryptoelect (+0.67). The largest hidden in-domain gains are in games we never targeted: TMW GraphReasoning (+71), ImageGame (+60), and AdventureGame (+58), while the targeted Wordle family remains near zero on hidden in-domain episodes (3.3, 10.0, and 3.3). Tables 11 and 12 give the full per-game decompositions.

On the pinned public-development set, the four-phase trajectory progresses from 13.05 through 43.85 (Phase A), 46.83 and 45.52 (Phase B passes), and 48.52 (Phase C) to a mean 50.43 after delta scaling (Table 2). Most adjacent-phase differences are small relative to the overall gain, and per-game changes outside Wordle are mixed (Tables 13 and 14), so we report these as a trajectory rather than a decomposition into causal effects.

5 Discussion

The official evaluation combines a large in-domain gain with weak out-of-domain transfer. The training trajectory and its follow-up experiments point to a common pattern: supervision is most useful when matched to what the current policy still lacks. Broad imitation creates participation, local preferences repair mechanically identifiable decisions, and weight-space selection chooses an operating point that retains the resulting gains.

System	Base	Public	Stat	Closed ID	Closed OOD	Δ ID	Δ OOD
Baseline	Qwen3.5-2B	10.67	44.24	13.41	3.72	—	—
	Qwen3.5-4B	26.66	51.33	34.02	17.99	—	—
	Llama-3.1-8B	19.53	45.59	31.24	22.62	—	—
	Qwen3.5-9B	31.91	53.90	41.12	24.91	—	—
	Qwen3.5-27B	60.30	65.05	64.34	43.51	—	—
LLP llp-final	Qwen3.5-9B	53.39	57.80	59.23	22.09	+18.11	−2.82
DAIR sft-dpo-v2	Qwen3.5-2B	49.60	43.53	50.75	14.56	+37.34	+10.84
SLED-BSU	Qwen3.5-27B	46.90	63.80	56.36	30.02	−7.98	−13.49
CityUoL GuidePlay	Qwen3.5-2B	46.66	42.30	46.26	10.25	+32.85	+6.53
DAIR sft-v1	Qwen3.5-2B	46.01	44.35	46.57	15.62	+33.16	+11.90
playornotplay (ours)	Qwen3.5-2B	38.92	44.14	41.17	7.88	+27.76	+4.16
BSU-SLIM	Qwen3.5-9B	36.61	49.92	39.64	15.18	−1.48	−9.73
Dialogue Architects	Qwen3.5-9B	34.39	53.26	43.12	23.37	+2.00	−1.54
Bentel iter_3	Qwen3.5-4B	33.35	54.27	37.76	16.60	+3.74	−1.39
Bentel iter	Qwen3.5-4B	31.58	54.27	37.41	16.60	+3.39	−1.39
Bentel iter_2	Qwen3.5-4B	24.27	47.37	31.97	10.19	−2.05	−7.80

Table 3: Official final evaluation results of the LM Playschool Challenge (<https://github.com/lm-playpen/lm-playschool-2026-final-results>, snapshot 2dd5a533). Public is the public interactive clem score; Stat is the static benchmark score; Closed ID and OOD are the organizer’s hidden in-domain and out-of-domain clem scores. Δ ID and Δ OOD are differences against each system’s own matching base. Submissions are sorted by public clem score.

	Played (base $\rightarrow$ ours)	Quality (base $\rightarrow$ ours)	Clem (base $\rightarrow$ ours)
Public	27.22 $\rightarrow$ 78.63	39.20 $\rightarrow$ 49.50	10.67 $\rightarrow$ 38.92
Closed ID	47.32 $\rightarrow$ 82.62	28.33 $\rightarrow$ 49.83	13.41 $\rightarrow$ 41.17
Closed OOD	35.19 $\rightarrow$ 33.08	10.56 $\rightarrow$ 23.81	3.72 $\rightarrow$ 7.88

Table 4: Played and Quality decomposition of the official final evaluation. OOD Played falls below the untrained base, indicating that the OOD clem score gain is driven entirely by higher conditional Quality on the episodes that remain in play.

5.1 Acquisition Creates the Conditions for Local Repair

Phase-level evidence comes from the pinned public-development trajectory and the controlled single-host campaign (Appendix A.5). Phase A is the largest single step in that trajectory (Table 2). Played rises sharply while conditional Quality also improves (Appendix B), and the final model’s official decomposition is consistent with this breadth: its largest hidden in-domain gains appear in games no later phase targeted (TMW GraphReasoning, ImageGame, AdventureGame; Table 11). The breadth of these final-model gains is consistent with the role of the only all-game training stage.

The later phases operate in a narrower regime. Only turn-DPO pass 1 has a positive clem marginal on both the submitted-model trajectory and the controlled campaign (+2.98 and +1.49); pass 2 and branch-DPO are sign-inconsistent across the two rails (Tables 2 and 8). The clearest repair signal

remains inside the Wordle family: three held-out Wordle-crazy configurations account for 82.5% of total positive out-of-domain movement, while the targeted Wordle family remains near zero on hidden in-domain episodes (Tables 12 and 11). The observed transfer is therefore concentrated in related Wordle variants.

Locality appears to be the hinge. In an earlier experiment, the preference covered the entire remaining dialogue rather than the next post-feedback action. Unfiltered whole-dialogue DPO scored 12.35 and produced natural-language preambles that violated game protocols across Wordle, Codenames, and other games; filtering the chosen side to stronger source models partially recovered the score to 25.76 but left the principal abort patterns intact (Appendix A.6). Taken together, the positive turn-local result and the whole-dialogue failures suggest that the clearest preference signal arises when the contrast is attached to a mechanically verifiable post-feedback decision. Whole-dialogue DPO, which paired entire remaining interactions rather than single responses, did not preserve protocol behavior even after source filtering.

5.2 Preservation as Weight-Space Model Selection

Delta scaling addresses a different problem from acquisition and repair. It adds no supervision and takes no gradient step; instead, it selects a point along the learned LoRA update. In the of-

ficial evaluation, aggregate static performance is approximately preserved (44.14 against the baseline’s 44.24), while the interactive gains remain substantial. The aggregate nevertheless hides redistribution: BBH Quality rises from 0.00 to 30.47, whereas IFEval falls from 68.52 to 51.85 and EQ-Bench Quality from 64.78 to 56.43.

Preservation is therefore a model-selection problem along the learned update, rather than another supervision problem. Delta scaling selects an operating point between interactive gain and aggregate static performance, with component-level redistribution.

5.3 Training Effects Depend on Stage and Parent

To test whether correction-based preference training remained useful after model selection, we continued DPO from the selected $s=0.85$ parent using solver corrections, success-versus-failure pairs, or teacher corrections. None of the five arms exceeded the 50.43 development endpoint; their scores ranged from 44.15 to 48.92 (Appendix A.6). At this point in the pipeline, additional correction-based training did not produce a stronger model.

The absence of further gains does not mean that the 2B base had exhausted its headroom. DAIR’s SFT-only system uses the same Qwen3.5-2B base and reaches 46.01 public, 46.57 hidden in-domain, and 15.62 hidden out-of-domain clemscore. This same-base comparison points back to acquisition: stronger broad supervision can create a better starting policy before local repair begins.

The effect of an acquisition update also depends on its parent. In a matched 4B probe, low-dose SFT on 700 off-policy successful episodes reduced clemscore from 40.75 to 24.36 on the same host, dataset revision, and evaluation rail. A separate 4B probe, trained on 303 self-generated successful dialogues, reached 35.06 (Appendix A.6). Together with the late-continuation results, these probes suggest that training effects are path-dependent: broad imitation helps when participation is missing, local preference learning helps once failures become mechanically identifiable, and further updates after the policy or parent has changed can instead degrade behavior the model already possesses.

5.4 Model Selection Shapes What Generalizes

Across the eleven submissions, public clemscore is strongly associated with closed in-domain score ($r=0.93$) but only weakly with closed out-of-

domain score ($r=0.30$). Every decision in our pipeline was made on the public-development set, and the final result mirrors that selection signal: a large public and hidden in-domain gain, but an out-of-domain clemscore of 7.88 and out-of-domain Played below the untrained base. The public set was informative about hidden in-domain progress but offered little guidance for broad transfer.

This pattern exposes a practical limit of the recipe-development process: the endpoint combines a large improvement over a weak starting point with limited absolute out-of-domain capability. Broader transfer would require either diagnostics or a selection signal that reaches beyond the Wordle family and the public development distribution.

6 Conclusion

Interactive dialogue-game failures can reflect both broad capability deficits and local decision errors. In our pipeline, broad SFT produces the largest improvement, while turn-local preference pairs provide the clearest positive repair signal among the preference-tuning variants we tested. Combining broad SFT, turn-local repair, and delta scaling raises official public clemscore from 10.67 to 38.92 and closed in-domain score from 13.41 to 41.17, above the official 4B and 9B baselines, while approximately preserving aggregate static performance. Out-of-domain transfer remains limited: clemscore is 7.88 and Played falls below the untrained base. The results support a scoped recipe for acquiring in-domain game participation and repairing precise within-family failures, with transfer concentrated in related Wordle variants.

Limitations

Scope of mechanically verifiable repair. Turn-local repair relies on game-specific diagnostics that verify whether a post-feedback action violates an explicit state constraint. That precision restricts demonstrated coverage to the Wordle family, the only game for which Phases B and C construct preference data; the method has not been tested on failures requiring semantic or strategic judgment. The demonstrated out-of-domain coverage is related-variant transfer: three held-out Wordle-crazy configurations account for 70.02 of 84.86 total positive movement (82.5%). The submitted artifact uses a single Qwen3.5-2B base, so scale applicability remains untested.

Untested dependence on the acquisition-stage parent. Broad SFT accounts for most of the measured improvement before repair is applied. The repair phases are evaluated after one specific, capped broad-SFT recipe; the study does not vary the strength or coverage of the acquisition-stage parent while holding the repair corpus and training configuration fixed. The experiments therefore cannot establish whether turn-local repair complements a stronger imitation stage, compensates for weaknesses specific to the submitted parent, or becomes less useful after stronger acquisition. Local repair is informative in the regime this parent creates; its contribution under a differently trained parent remains unknown.

Limits of phase attribution and model selection.

The official evaluation validates the submitted endpoint, but the sequential trajectory does not identify the causal contribution of each phase. Phase and scaling choices reuse a 67-episode public-development set, and later-phase marginals are not consistently signed across the two evaluation rails. Two later phases are sign-inconsistent across rails: turn-DPO pass 2 is -1.31 on the development trajectory and $+1.34$ on the controlled campaign; branch-DPO is $+3.00$ and -2.07 . Only pass 1 is positive on both ($+2.98$ and $+1.49$). Adjacent checkpoint differences combine training, sequential dependence, repeated selection, and evaluation variance, so they describe a trajectory rather than additive causal effects. The endpoint improves substantially over the base; the magnitude and necessity of individual later phases remain uncertain.

Aggregate rather than component-wise preservation. Delta scaling approximately preserves the official aggregate statscore (44.14 vs. 44.24) while retaining interactive gains. The preservation objective is itself aggregate: Eq. 3 maximizes aggregate static score subject to an aggregate interactive gate, so gains in one component can offset losses in another by design. BBH Quality rises from 0.00 to 30.47 while IFEval drops from 68.52 to 51.85 and EQ-Bench Quality from 64.78 to 56.43. The defensible conclusion is aggregate-level trade-off control, not preservation of every general capability. A stronger claim would require component-wise floors or a multi-objective selection criterion.

Acknowledgments

We appreciate the helpful comments and suggestions from the anonymous reviewers. This work is funded by the Dutch Research Council (NWO) through the AiNed Fellowship Grant NGF.1607.22.002, *Dealing with Meaning Variation in NLP*. This work used the Dutch national e-infrastructure with the support of the SURF Cooperative using grant no. EINF-18781/L1.

References

- Kranti Chalamalasetti, Jana Götze, Sherzod Hakimov, Brielen Madureira, Philipp Sadler, and David Schlangen. 2023. [clembench: Using game play to evaluate chat-optimized language models as conversational agents](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 11174–11219, Singapore. Association for Computational Linguistics.
- Nicola Horst, Davide Mazzaccara, Antonia Schmidt, Michael Sullivan, Filippo Momentè, Luca Franceschetti, Philipp Sadler, Sherzod Hakimov, Alberto Testoni, Raffaella Bernardi, Raquel Fernández, Alexander Koller, Oliver Lemon, David Schlangen, Mario Giulianelli, and Alessandro Suglia. 2025. [Playpen: An environment for exploring learning from dialogue game feedback](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 29854–29891, Suzhou, China. Association for Computational Linguistics.
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [LoRA: Low-rank adaptation of large language models](#). In *International Conference on Learning Representations*.
- Zhijing Jin, Yuen Chen, Felix Leeb, Luigi Gresele, Ojasv Kamal, Zhiheng LYU, Kevin Blin, Fernando Gonzalez Adauto, Max Kleiman-Weiner, Mrinmaya Sachan, and Bernhard Schölkopf. 2023. [CLadder: Assessing causal reasoning in language models](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 31038–31065. Curran Associates, Inc.
- Samuel J. Paech. 2023. [EQ-Bench: An emotional intelligence benchmark for large language models](#). *Preprint*, arXiv:2312.06281.
- Qwen Team. 2026. [Qwen3.5: Towards native multi-modal agents](#).
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. [Direct preference optimization: Your language model is secretly a reward model](#). In *Advances in Neural Information Processing Systems*, volume 36, pages 53728–53741. Curran Associates, Inc.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. [DeepSeekMath: Pushing the limits of mathematical reasoning in open language models](#). *Preprint*, arXiv:2402.03300.

Mirac Suzgun, Nathan Scales, Nathanael Schärli, Sebastian Gehrmann, Yi Tay, Hyung Won Chung, Aakanksha Chowdhery, Quoc Le, Ed Chi, Denny Zhou, and Jason Wei. 2023. [Challenging BIG-bench tasks and whether chain-of-thought can solve them](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13003–13051, Toronto, Canada. Association for Computational Linguistics.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhuranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhui Chen. 2024. [MMLU-Pro: A more robust and challenging multi-task language understanding benchmark](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 95266–95290. Curran Associates, Inc.

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. [Instruction-following evaluation for large language models](#). *Preprint*, arXiv:2311.07911.

A Training and Evaluation Details

A.1 Submitted Model

The submitted artifact is the checkpoint at Hugging Face revision 828e356⁷. This is the revision used for the official evaluation.

The submitted model is an fp32 merge of a LoRA adapter into the Qwen3.5-2B base. Its training costs approximately 5.3 hours on a single A100-80GB GPU.

A.2 Evaluation Pins

The paper reports scores on two dev/test sets, which are not directly comparable.

Official final evaluation. The organizer’s evaluation at snapshot 2dd5a533 uses clemcore version 3.7.2. The official metadata does not record the exact clembench game-tree revision. Public, closed in-domain, and closed out-of-domain scores constitute the primary result. The public suite contains 502 episodes (72 interactive, 430 static); the closed suites contain 1,272 in-domain and 360 out-of-domain episodes. All runs use temperature 0, a 5,000-token generation limit, and thinking mode disabled.

Pinned public-development evaluation. Our development evaluations pin dataset revision 557d8caf and clembench tree commit ed39486, yielding 67 interactive episodes across 14 games. This set produces the per-phase trajectory in Table 2 and the matched per-game comparison in Table 13.

A.3 Training Details

A.3.1 Toolkits

The training pipeline uses Playpen trainers built on Hugging Face transformers, peft, and trl. Phase A uses TRL’s SFTTrainer; Phases B and C use DPOTrainer.

A.3.2 Broad-SFT Corpus

Phase A trains on success-only episodes from playpen-data, capped at 700 per game, with seed-42 shuffling and a 95/5 train-eval split. Table 6 lists per-game counts at the training-time snapshot.

A.3.3 Turn-Local Preference Construction

Phase B constructs preference pairs from clean successful Wordle episodes. Eligibility is decided in three steps:

⁷<https://huggingface.co/chnln/Qwen3.5-2B-playpen-playornotplay>

Ph.	Data	Objective	LR / sched. / warmup	Batch / length / epochs	Other
A	9,522 rows	SFT, AdamW	2e-4 / cosine / 0.03	micro 8 × accum 4 = 32; 2,048; 1	LoRA $r16/\alpha32$ /dropout 0.05/all-linear; bf16; grad. ckpt. on; seed 42
B1	1,905 pairs	DPO, $\beta=0.1$	5e-6 / cosine / 0.03	micro 1 × accum 16 = 16; 1,024; 1	bf16; grad. ckpt. off; separate frozen reference
B2	1,906 pairs	DPO, $\beta=0.1$	5e-6 / cosine / 0.03	micro 1 × accum 16 = 16; 1,024; 1	bf16; grad. ckpt. off; separate frozen reference
C	77 pairs	DPO, $\beta=0.2$	5e-6 / cosine / 0.03	micro 1 × accum 16 = 16; 1,024; 1	bf16; grad. ckpt. on; separate frozen reference; factor 2, rounds 1–3, $T=0.7$, 256 new tokens
D	—	delta scaling $s=0.85$	no optimizer	—	multiply each lora_B by 0.85, then merge in fp32

Table 5: Complete training configuration for the four effective phases. All trainable phases continue the same LoRA adapter on Qwen3.5-2B in bf16.

Game	Success	Capped	Train
AdventureGame	429	429	403
Codenames	2,402	700	669
DoND	1,256	700	663
GuessWhat	2,170	700	662
HotAirBalloon	1,102	700	658
ImageGame	1,018	700	662
MatchIt ASCII	1,654	700	668
PrivateShared	228	228	221
ReferenceGame	3,872	700	658
Taboo	2,522	700	678
TextMapWorld	742	700	666
TMW GraphReasoning	356	356	338
TMW SpecificRoom	717	700	660
Wordle	371	371	351
Wordle w/ Clue	438	438	419
Wordle w/ Critic	740	700	669
Total	20,017	9,522	9,045

Table 6: Broad-SFT per-game counts at the reconstructed training-time snapshot (cf23af92). Success counts successful episodes; Capped applies the 700-per-game limit; Train is the count after the 95/5 split.

1. The source row has game wordle and outcome success. The optional source-model filter was empty, so all successful episodes contribute.
2. The whole episode is rejected if any user message contains “not a valid word” or if any parsed assistant guess repeats within the episode.
3. The selected assistant reply contains a parseable lowercase guess: field, is assistant turn 2 or later, and immediately follows a user message containing guess_feedback:.

The prompt is the exact dialogue history before that reply. The preferred completion y^+ is

the unmodified response from the clean successful episode; it is not a teacher-generated correction. The rejected completion y^- applies one of four deterministic corruptions to that same response:

- **repeat_last**: replace the current guess with the previous guess; skip if already equal.
- **bad_length**: replace a guess of at least five characters with its first four.
- **invalid_word**: try fixed single-letter substitutions and then fixed candidates, taking the first five-letter alphabetic token absent from the official Wordle lexicon, with zzzzz as fallback.
- **duplicate_guess_keyword**: append a second guess: line to the otherwise unchanged response.

Pass 1 uses repeat_last and bad_length; pass 2 uses invalid_word and duplicate_guess_keyword. One-negative sampling was not enabled, so every configured corruption that applies produces a pair. Table 7 shows a real failure of the kind the corruptions imitate.

A.3.4 Branch-DPO Construction

Phase C loads the train-split instance configuration, filters to Wordle, shuffles with seed 42, and selects 27 instances. The current checkpoint plays each instance with branching factor 2 at rounds 1–3, temperature 0.7, and at most 256 generated tokens per turn. Sibling branches are grouped by shared game-snapshot origin, and a group needs at least two branches. Within a group, siblings

Role	Content
Model	guess: slate
GM	guess_feedback: s<red> l<red> a<red> t<red> e<red>
Model	guess: crane
GM	guess_feedback: c<red> r<green> a<red> n<green> e<red>
Model	The feedback shows that 'r' and 'n' are correct in their positions, while 'c', 'a', and 'e' are incorrect... guess: ringer
GM	Error: The length of the guessed word is not 5. game_result = ABORT

Table 7: A real Wordle failure from the broad-SFT checkpoint on the pinned public-development set (medium-frequency instance 14, target word *grind*). The model reasons correctly about letter positions and then produces a six-letter guess. This is a diagnostic example, not a Phase B training row: Phase B pairs are built from clean successful episodes.

are ranked by the score of the complete downstream episode, which combines terminal outcome, speed, closeness, strategy, and penalties for repeated, invalid, bad-length, duplicated, missing, or feedback-inconsistent guesses; it is not a binary win/loss label. The chosen and rejected items are only the immediate diverging assistant responses of the highest- and lowest-scoring siblings, with the shared history as prompt. A group is dropped when the two scores are equal.

The run produced 90 trajectories, 295 branch points, and 77 DPO pairs (73 train, 4 eval).

A.4 Two Nominal Stages Removed

The training process for the submitted model was originally described as six stages. Nominal Stage 2 was a continued SFT pass over the weakest games; nominal Stage 3 was a GRPO (Shao et al., 2024) probe on Wordle with a dense reward that parses feedback and penalizes six mechanical violations. Neither changed a single weight, for two unrelated reasons.

Nominal Stage 2. The run loaded the broad-SFT adapter through the inference loader, which sets `is_trainable=False`, and the warm-start path passed `peft_config=None`, so no new trainable adapter was created. TRL’s `enable_input_require_grads()` kept the forward and backward passes from raising, so the run completed and wrote checkpoints. The output adapter is byte-identical to the broad-SFT checkpoint (372 of 372 tensors equal) and its optimizer state contains zero parameters.

Nominal Stage 3. With $\beta \neq 0$, GRPOTrainer in TRL 0.29.1 builds the frozen reference by calling `add_adapter` with the same configuration object, which still carries `inference_mode=True`. Adapter injection therefore silently re-froze every adapter, the policy included, and the optimizer was constructed afterwards on an empty parameter set. The log reports 16.8M trainable parameters and 54 non-zero gradient norms, but every gradient was computed and discarded; the KL term is identically zero across all 54 steps.

Behavioral corroboration. Re-evaluating the three nominal checkpoints at temperature 0 on the pinned split, 13 of 14 games produce byte-identical transcripts. The one divergence is AdventureGame, traced to Python set iteration order under PYTHONHASHSEED. The three scores 43.85, 44.09, and 43.35 are thus three evaluations of one set of weights, and their 0.74-point spread is a direct measurement of evaluation noise on this suite.

Corrected re-runs. Both stages were re-run with the bugs fixed on a single host, using the same seeds, learning rates, batch geometry, and reward shaping as the originals. Because the corrected runs used the paper’s pinned dataset revision rather than the original snapshot, the weak-game SFT corpus changed from 2,119 to 2,224 rows and the GRPO instance set shifted by two task IDs. The corrected runs therefore reproduce the recipe family with slightly different corpora. The corrected weak-game SFT scored 42.07 (−1.78 from its 43.85 parent) and the corrected GRPO scored 43.55 (+1.48 from that parent), a net −0.30 over both. Training dynamics confirm genuine optimization: non-zero gradient norms where the originals showed zeros, and a non-zero KL term where the original was identically zero.

A.5 Controlled Single-Host Ablation

Table 8 reports a controlled campaign run on one host, approximately one month before the final Hub commit. All rows share the same host, harness, and session. The campaign predates the submission and was available at ship time.

Of the three later phases, only turn-DPO pass 1 has a positive clem marginal on both the submitted-model trajectory and this campaign (+2.98 and +1.49); pass 2 (−1.31/+1.34) and branch-DPO (+3.00/−2.07) are sign-inconsistent across the two rails. A consistent trade appears across phases: the

Row	Configuration	Clem	Stat
A	broad SFT only	44.09	39.88
B	A + turn-DPO pass 1	45.58	39.62
C	A + turn-DPO pass 1 (repeat)	45.94	40.96
D	C + turn-DPO pass 2	47.28	40.31
E	D + branch-DPO	45.21	41.54

Table 8: Controlled single-host ablation. Rows B and C independently apply the same turn-DPO pass-1 configuration to the same effective parent; their difference (+0.36 clem, +1.34 stat) estimates repeat-run variation. Adding branch-DPO to Row D reduces clem by 2.07 and raises stat by 1.23 (Row E). Row A’s adapter is byte-identical to the broad-SFT adapter that scored 43.85 on the pinned rail; the 0.24-point difference is evaluation variation measured in §A.4. This controlled campaign and Table 2 use separate evaluation rails.

Arm	Pairs	Clem	vs. 50.43
Wordle-only solver	66	48.39	−2.0
All failures	675	44.15	−6.3
Wordle high-dose	66	44.99	−5.4
Conservative SVF	500	48.92	−1.5
SVF, then $s=0.95$	—	48.08	−2.4
SVF, then $s=0.90$	—	46.68	−3.8
SVF from Phase A parent	500	49.76	−0.7
SVF from unscaled Phase C	500	48.74	−1.7
GPT-5.5 teacher, shipped parent	411	46.35	−4.1
GPT-5.5 teacher, Phase A parent	411	27.20	−23.2

Table 9: Ten evaluated late-continuation endpoints. Eight are DPO runs ($\beta=0.20$, max length 1,024, micro-batch 1, separate frozen reference); two are training-free rescalings of a preceding DPO arm (marked —). SVF denotes success-versus-failure pairs. Of the eight DPO runs, five start from the selected $s=0.85$ adapter and three start from other parents (Phase A or unscaled Phase C). The 50.43 comparator is the mean of two evaluations of the selected adapter and serves as the contextual reference for this campaign.

phases that raise clem cost a small amount of stat, and the one that costs clem gains stat.

A.6 Abandoned Directions

Late continuation. Table 9 reports ten evaluated endpoints from the late-continuation campaign: eight DPO runs and two post-DPO training-free rescalings.

Solver pairs identify a bad Wordle turn and pair the actual response against a rule-based correction. Success-versus-failure pairs select the first error-followed assistant turn, or otherwise the last, and match a successful same-game response by prompt similarity. The teacher corpus is 345 GPT-5.5 corrections plus 66 solver corrections. Continuation of an existing adapter passes no new PEFT configu-

ration, so the effective adapter architecture remains the parent’s, despite what some launcher variables advertise.

Both cross-parent comparisons in the table jointly vary parent, learning rate, and epoch count (arms 4 vs. 7 for the 500-pair SVF corpus; arms 9 vs. 10 for the 411-pair teacher corpus).

Whole-trajectory preference tuning. Two earlier DPO arms trained on full-dialogue preference pairs from the broad-SFT checkpoint, continuing at LR $5e-6$, $\beta=0.1$, micro-batch $1 \times$ accumulation 8, max length 2,048, one epoch. The prompt was the initial user message and both completions were entire remaining dialogues, pairing one success against one random failure or abort from the same game, experiment, task, role, and opening message.

- **Unfiltered whole-dialogue DPO:** the chosen side is any successful transcript; 3,973 pairs. Clemscore 12.35. The model began emitting natural-language preambles before game actions, violating strict request format across Wordle, Codenames, and other games.
- **Source-filtered whole-dialogue DPO:** the chosen side is filtered to transcripts from six stronger source models; 1,337 pairs. Clemscore 25.76. Protocol compliance partially recovered, but Wordle and Codenames still aborted; all six GuessWhat episodes were played, but none was won.

No live teacher was queried in either arm: the source-filtered variant uses transcripts that already exist in playpen-data, filtering them by source-model name. Unfiltered whole-dialogue preference learning did not merely fail to help; it destroyed protocol compliance across multiple games. The contrast with Phase B motivates held-fixed histories and token-level contrasts.

4B supervised probes. We tested two SFT-only updates on Qwen3.5-4B. In the matched off-policy probe, a fresh LoRA adapter trained for one epoch on 700 successful episodes reduced clemscore from 40.75 to 24.36 on the same host, dataset revision, clembench revision, and evaluation rail. A separate probe let the vanilla 4B model generate training trajectories, retained its 303 successful full dialogues across 14 games, and applied success-only SFT; the resulting model scored 35.06. Both probes used LR $2e-5$, effective batch size 16, and maximum length 2,048.

Benchmark	Baseline		Ours	
	Played	Quality	Played	Quality
BBH	100.00	0.00	100.00	30.47
CLadder	100.00	48.51	100.00	46.53
EQ-Bench	97.06	64.78	94.12	56.43
IFEval	100.00	68.52	100.00	51.85
MMLU-Pro	100.00	40.71	100.00	38.05

Table 10: Public static components of the official evaluation. Applying Eq. 1 gives statscore 44.24 for the baseline and 44.14 for our submission. The aggregate is approximately preserved, but the components are redistributed: BBH Quality rises by 30.47 while IFEval falls by 16.67 and EQ-Bench Quality by 8.35.

A.7 Official Per-Game Decompositions

Tables 11 and 12 decompose the official final evaluation per game for the 2B baseline and our submission. Table 10 gives the public static components behind the reported statscores.

B Per-Game Matched Comparison

Table 13 decomposes the pipeline per game at four checkpoints: the vanilla base, the broad-SFT checkpoint (Phase A), the last Wordle-trained checkpoint (Phase C, before delta scaling), and the shipped fp32 merge. To make the per-game deltas internally comparable, all four checkpoints were re-evaluated on a single host with the same evaluation harness, dataset revision 557d8caf, and clembench commit ed39486; the matched base score coincides with the organizer-reported baseline (13.05). We use these matched runs for the per-game analysis and the official evaluations for the headline results in the body.

At suite level, macro-averaged Played rises from 30.95 to 84.19 between the base and Phase A, while macro-averaged Quality over played games rises from 42.15 to 52.09.

The three delta columns separate what changed during broad SFT (Base→SFT), during the Wordle-only phases (SFT→Branch), and under gradient-free delta scaling plus the fp32 merge (Branch→Final). Changes outside Wordle during the Wordle-only phases are mixed in sign and rest on 3–13 episodes per game; their interpretation is descriptive.

C Plain-Wordle Behavior by Phase

Table 14 tracks the three public plain-Wordle episodes across the pipeline, using process-level metrics rather than the game score alone. Among

checkpoints with played plain-Wordle episodes, mean repetitions fall from 3.0 after broad SFT to 0 after the second Phase B pass; repetition is unobservable after the first Phase B pass and after scaling because no episode is played. Phase C shows the most stable protocol behavior among the evaluated checkpoints (100% played, no violated requests, 100% request success), yet conditional Quality is 0 at every checkpoint with played episodes: the Wordle-focused phases repaired protocol-level failures on this public probe but did not solve the underlying guessing task.

Across repeated same-harness evaluations, the unscaled Phase C checkpoint plays all three public episodes, while the $s=0.85$ -scaled checkpoint and all downstream artifacts abort all three, reintroducing a bad-length guess of the kind targeted in the first Phase B pass. Repeated evaluations establish this pattern on the fixed three-episode probe.

Game	Qwen3.5-2B baseline			playornotplay (ours)			Δ Eff.
	Played	Quality	Eff.	Played	Quality	Eff.	
TMW GraphReasoning	0.00	0.00	0.00	100.00	70.99	70.99	+70.99
ImageGame	80.00	12.03	9.62	97.50	71.46	69.67	+60.05
AdventureGame	0.00	0.00	0.00	100.00	58.33	58.33	+58.33
TextMapWorld	0.00	0.00	0.00	100.00	57.91	57.91	+57.91
TMW SpecificRoom	0.00	0.00	0.00	100.00	53.33	53.33	+53.33
DoND	30.00	0.00	0.00	100.00	39.75	39.75	+39.75
Taboo	93.33	8.63	8.05	100.00	47.78	47.78	+39.73
BBH	100.00	0.00	0.00	100.00	39.42	39.42	+39.42
PrivateShared	0.00	0.00	0.00	56.00	38.69	21.67	+21.67
GuessWhat	0.00	0.00	0.00	100.00	18.33	18.33	+18.33
Codenames	0.00	0.00	0.00	46.92	29.51	13.85	+13.85
MatchIt ASCII	100.00	75.00	75.00	100.00	85.00	85.00	+10.00
Wordle w/ Clue	6.67	50.00	3.33	10.00	100.00	10.00	+6.67
MMLU-Pro	100.00	23.00	23.00	100.00	28.00	28.00	+5.00
Wordle	16.67	0.00	0.00	43.33	7.69	3.33	+3.33
Wordle w/ Critic	3.33	0.00	0.00	3.33	100.00	3.33	+3.33
Clean Up	66.67	3.17	2.11	88.89	4.85	4.31	+2.20
CLadder	100.00	52.00	52.00	100.00	50.00	50.00	-2.00
ReferenceGame	100.00	45.56	45.56	100.00	43.33	43.33	-2.23
IFEval	100.00	64.00	64.00	100.00	50.00	50.00	-14.00
EQ-Bench	97.00	63.20	61.30	89.00	52.09	46.36	-14.94

Table 11: Official hidden in-domain per-game decomposition, 21 scored configurations, sorted by change in effective score. Eff. is the displayed diagnostic $\text{Played} \times \text{Quality}/100$; these values are not averaged to reconstruct the suite clemscore (Eq. 1).

Game	Qwen3.5-2B baseline			playornotplay (ours)			Δ Eff.
	Played	Quality	Eff.	Played	Quality	Eff.	
Wordle-crazy w/ Critic	6.67	0.00	0.00	36.67	90.91	33.34	+33.34
Wordle-crazy w/ Clue	6.67	50.00	3.33	36.67	100.00	36.67	+33.34
TA Frozen Lake	50.00	26.11	13.05	40.00	55.56	22.22	+9.17
TA Mastermind	0.00	0.00	0.00	15.00	33.33	5.00	+5.00
Wordle-crazy	3.33	0.00	0.00	80.00	4.17	3.34	+3.34
Cryptoelect	56.67	0.00	0.00	33.33	2.00	0.67	+0.67
Clockwork Courier	0.00	0.00	0.00	56.67	0.00	0.00	+0.00
Get to the Point	40.00	0.00	0.00	35.00	0.00	0.00	+0.00
TA Sokoban	10.00	0.00	0.00	10.00	0.00	0.00	+0.00
Tower of Hanoi	20.83	0.00	0.00	25.00	0.00	0.00	+0.00
Chronicle	100.00	5.02	5.02	3.33	0.00	0.00	-5.02
ST Clean Up	83.33	11.65	9.71	33.33	4.85	1.62	-8.09
TA Blackjack	80.00	23.33	18.66	25.00	18.67	4.67	-14.00

Table 12: Official hidden out-of-domain per-game decomposition, 13 scored configurations from eight game families, sorted by change in effective score. The three Wordle-crazy configurations contribute 82.5% of total positive movement. Columns are as in Table 11.

Game	n	Base	Broad SFT	Branch-DPO	Final	Δ Base→SFT	Δ SFT→Branch	Δ Branch→Final
AdventureGame	5	0.0	33.3	40.0	33.3	+33.3	+6.7	−6.7
Codenames	13	0.0	0.0	7.7	7.7	0.0	+7.7	0.0
GuessWhat	6	0.0	50.0	33.3	50.0	+50.0	−16.7	+16.7
ImageGame	4	35.8	81.2	89.8	85.2	+45.4	+8.6	−4.6
MatchIt ASCII	4	75.0	50.0	75.0	75.0	−25.0	+25.0	0.0
PrivateShared	4	0.0	35.1	21.3	18.3	+35.1	−13.8	−3.0
ReferenceGame	5	100.0	80.0	60.0	80.0	−20.0	−20.0	+20.0
Taboo	6	0.0	41.7	41.7	25.0	+41.7	0.0	−16.7
TextMapWorld	5	0.0	55.6	65.2	65.7	+55.6	+9.6	+0.5
TMW GraphReasoning	3	0.0	58.9	58.9	58.9	+58.9	0.0	0.0
TMW SpecificRoom	3	0.0	66.7	66.7	66.7	+66.7	0.0	0.0
Wordle	3	0.0	0.0	0.0	0.0	0.0	0.0	0.0
Wordle w/ Clue	3	0.0	33.3	50.0	50.0	+33.3	+16.7	0.0
Wordle w/ Critic	3	0.0	66.7	66.7	66.7	+66.7	0.0	0.0
Overall	67	13.05	43.85	48.52	48.70	+30.80	+4.67	+0.18

Table 13: Matched per-game comparison of four checkpoints on the pinned public-development set (single host, single harness, dataset 557d8caf, clembench ed39486). The per-game score is the derived $P_g Q_g / 100$; it is displayed as 0 when Played = 0, although Quality itself is undefined there (Eq. 1). The overall clemscore is the product of macro-averaged Played and macro-averaged Quality over played games, not the mean of the per-game column, so per-game deltas do not sum to the overall delta. The Broad SFT column is the Phase A checkpoint and the Branch-DPO column is the Phase C checkpoint (Table 2). Δ Branch→Final includes gradient-free delta scaling and the fp32 merge, not additional training. Each game has 3–13 episodes; the changes are descriptive.

Checkpoint	Training scope	Suite Clem	W Played	W Quality	Rep./played ep.	Viol./ep.	Req. success
Broad SFT weights*	all games	43.85	66.7	0	3.00	0.33	88.9
Phase B pass 1	Wordle	46.83	0	—	—	1.00	63.9
Phase B pass 2	Wordle	45.52	66.7	0	0.00	0.33	91.7
Phase C	Wordle	48.52	100	0	0.00	0.00	100.0
+ scaling $s=.85$	none (no gradient)	50.82 / 50.03	0	—	—	1.00	69.4
Shipped fp32 merge	none (merge only)	48.70 / 49.10	0	—	—	1.00	69.4

Table 14: Plain-Wordle behavior after each phase on the three public validation episodes (development evaluations; pinned dataset 557d8caf). *Three greedy evaluations of byte-identical weights span 43.35–44.09; the difference is evaluation noise (§A.4). Suite Clem is the full 14-game clemscore; for the scaled and shipped artifacts we list the two runs of each. W Quality is undefined (—) when no episode is played. Rep./played ep. is the mean clembench *Guess Repetitions* over played episodes; Viol./ep. and Req. success are the mean *Violated Request Count* and *Request Success Ratio* (%) over all three episodes.

Author Index

- Baratella, Simone, 160
Bernardi, Raffaella, 110
Boukourt, Sami, 1
Buttery, Paula, 57
- Fernández, Raquel, 110
Foroutan, Neda, 132
Fujita, Haruto, 49
- Ghosh, Shreya, 127
Giulianelli, Mario, 110
Gu, Tianle, 19
- Hakimov, Sherzod, 110
Harshavaishnav, Velineni, 127
Hasegawa, Ryo, 97
Hayashi, Katsuhiko, 49
Huq, Syed Mahbubul, 140
- Kamigaito, Hidetaka, 49
Kanayama, Ryoki, 49
Karasawa, Karina, 49
Keeling, Geoff, 1
Koki, Raika, 49
Koller, Alexander, 110
- Le, Dieu-Thu, 110
Lemon, Oliver, 110
Li, Nan, 170
Liu, Guanming, 30
- Madhyastha, Pranava, 140
Manina, Maria, 110
Matsuno, Shogo, 49
Mazzaccara, Davide, 110
McCallum, Sabrina, 110
- Meshram, Valhari, 132
Momentè, Filippo, 110
Monz, Christof, 41
- Naszadi, Kata, 41
- Obara, Ryoma, 49
Osswald, Karl, 110
- Rajput, Abhinav, 80
Ristic, Dimitrije, 110
Rocca, Roberta, 1
- Sadler, Philipp, 110
Sakai, Yusuke, 49
Salhan, Suchir, 57
Sarzotti, Marika, 160
Schlangen, David, 110
Sonoda, Tetsuya, 97
Srivastava, Varad, 148
Street, Winnie, 1
Suglia, Alessandro, 110
Suzumura, Yuki, 49
- Tej, Vasamsetti Nihal, 127
Teng, Yan, 19
Tsuchida, Rikuto, 97
- Utsuro, Takehito, 97
- Vogelstein, Acey, 80
- Yang, Xinqi, 19
Yang, Yiwen, 30