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 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.

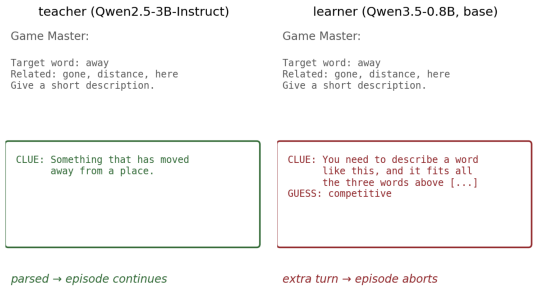

Figure 1: **Protocol competence in the Taboo Game (Horst et al., 2025b).** *Left:* the Teacher Qwen model produces a valid clue in the format expected by the game, allowing the interaction to continue and the clue to be evaluated. *Right:* the Learner Qwen SLM produces a plausible clue but then generates 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.

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

The Proceedings of Improving Language Models through Learning from Dialogue Game Interaction, pages 57–79
October 28, 2026 ©2026 Association for Computational Linguistics

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 `clembench` 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 ($\frac{\text{wins + losses}}{\text{episodes}}$), 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 $1-$ 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 $\mathcal{L}(\theta) = -\sum_{\tau \in \mathcal{S}} \sum_t w(\tau, t) \log \pi_\theta(a_t \mid s_t)$. **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 Figure 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).

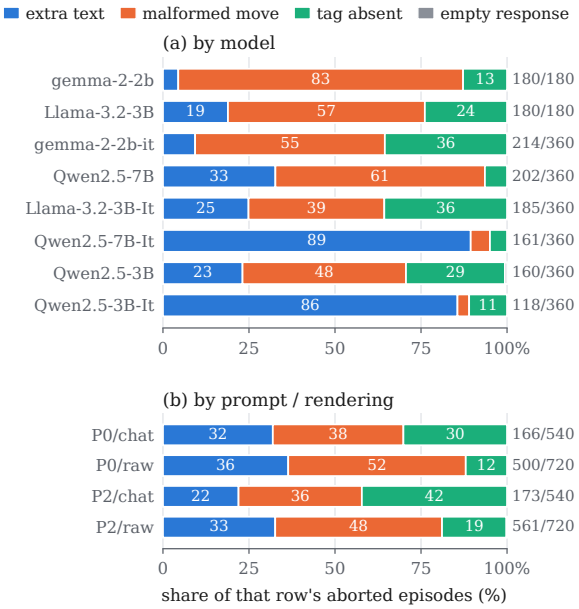

Figure 2: **Relative distribution of abort categories.** Distribution of categories across games (a) different teacher models and (b) different prompt strategies normalised to 100%

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**

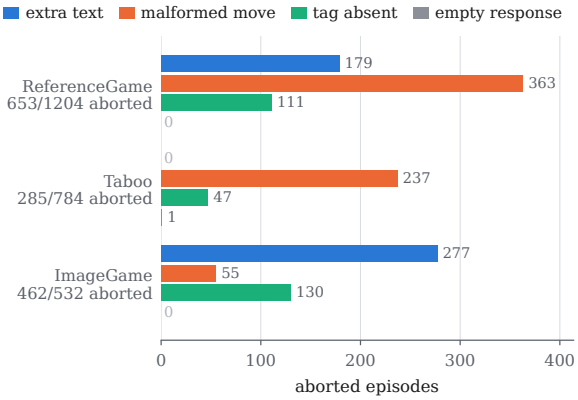

Figure 3: **Number of aborted episodes per abort category, by game.**

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.

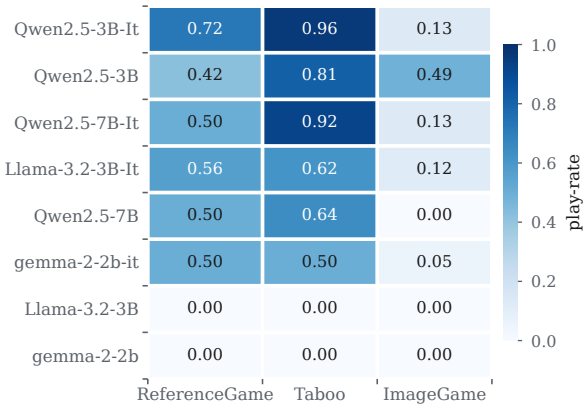

Figure 4: **Play-rate by Teacher Model and Playpen Game**

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	ΔPlay	CLEMSCORE	ΔClem
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**
Wordle-specific repair				
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.

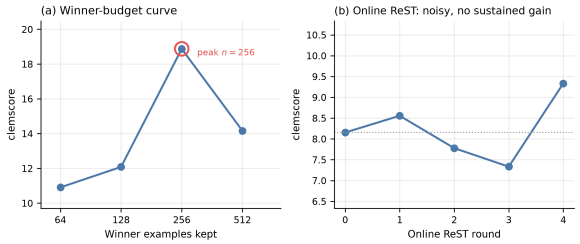

Figure 5: **Self-ReST learning curves:.** Improvement occurs rapidly once protocol competence has been acquired. The table reports the resulting CLEMSCORES after training on different numbers of winning trajectories. Each value corresponds to a single run.

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 Schlangen (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, FollowBench, 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 Almushayqih, Faisal Mirza, Nouf Alotaibi, Nora Al-Twairesh, 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. FollowBench: 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. Agentbench: 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 Procotol 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.

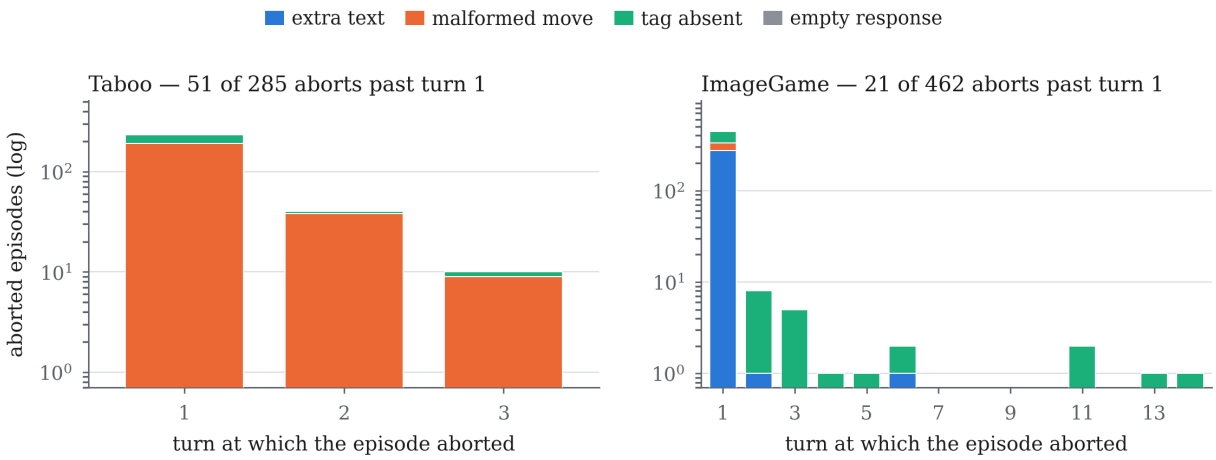

Figure 6: How far episodes progress before they fail.

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	%
Protocol failures		
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.

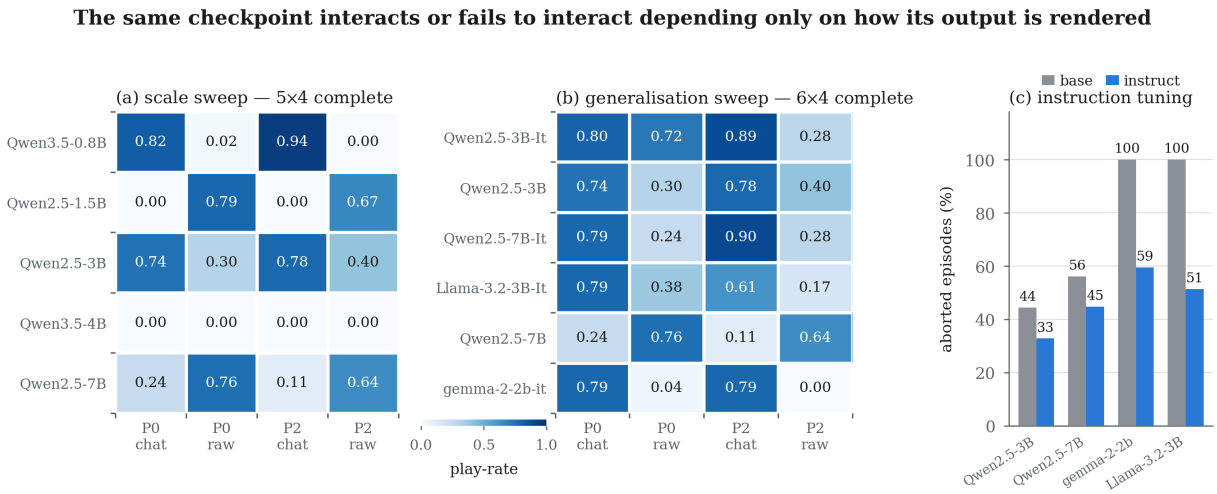

Figure 7: Play-rate changes with prompt and output rendering.

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
Base	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
Instruction-tuned	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.
by game						
ReferenceGame	1204	653	179	363	111	0
Taboo	784	285	0	237	47	1
ImageGame	532	462	277	55	130	0
by prompt / rendering						
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:**

	Play-rate		Quality	
Task	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 *(general)*	0.00	0.00	0.0	0.0
Wordle *(dedicated)*	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 *invalid format* 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_endl> 3 You are given three grids, where each of them is 5 by 5 in size. Grids have empty [. . .] 4 Answer: first<lim_endl> 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_endl> 3 You are given three grids, where each of them is 5 by 5 in size. Grids have empty [. . .] 4 Answer: first<lim_endl> 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_endl> 3 *invalid clue* 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_endl> 3 *invalid clue* 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_endl> 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_endl> 5 *invalid format* 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_endl> 3 Let us draw something together. There is an empty grid with a size 5 by 5, like [. . .] 4 □ <lim_endl> 5 *invalid format* 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 *invalid format* 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_endl> 3 *invalid format* 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_endl> 3 *invalid format* aborted
PrivateShared	1 You are an applicant in a job interview. Here is a description of your CV: WHAT: [. . .] 2 Ok. 3 *invalid format* 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 *invalid format* 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 *invalid format* 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