

Disentangled Privileged States for Simulator-Assisted Multi-Turn Dialogue Policy Training

Zhiyong Cao^{1,2,3,7*}, Dunqiang Liu^{4,7*}, Sicheng Zhou^{5,7†}, Cong Lin^{6†}

¹Institute of Automation, Chinese Academy of Sciences

²School of Artificial Intelligence, University of Chinese Academy of Sciences

³Zhongguancun Academy ⁴School of Automation, Beijing Institute of Technology

⁵Agent Efficiency Lab ⁶State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications ⁷TraceLite AI

Correspondence: caozhiyong2024@ia.ac.cn, zsicheng74@gmail.com

Abstract

Simulator-assisted reinforcement learning is a practical way to train multi-turn dialogue agents, but policy optimization can be misled when simulated users provide inconsistent feedback across long interaction trajectories. In this paper, we focus on a simulator-side question: can structured latent state tracking improve measured simulator consistency and provide useful process supervision for dialogue policy training? Motivated by a standard model-based RL simulation-bound perspective, we introduce **DPS**, a framework that decouples task-conditioned privileged state tracking from surface utterance realization. These privileged states are hidden from the agent but used during training to derive dense, state-aware process rewards that complement sparse outcome signals. We also construct RecruitmentBench, a recruitment persuasion dialogue benchmark grounded in anonymized real-world logs. Experiments on RecruitmentBench and two public datasets use offline real-log-grounded and cross-simulator protocols to evaluate intrinsic simulator consistency and downstream simulator-assisted policy performance. Results indicate that DPS improves measured simulator fidelity and yields stronger fine-tuned policy performance in these controlled evaluation settings, while remaining competitive with strong prompt-based LLM agents. We further calibrate our LLM-based evaluator against human judgments on MATH-CHAT. Code is available at: <https://github.com/TraceLite-AI/DPS>.

1 Introduction

Reinforcement Learning (RL) (Guo et al., 2025; Ouyang et al., 2022) has become a primary approach for training Large Language Model (LLM) agents to complete complex, long-horizon tasks, including multi-step reasoning (Wang et al., 2025a)

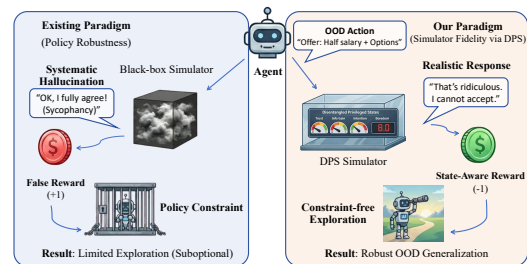


Figure 1: **Comparison of simulator-assisted dialogue policy training paradigms.** **Left:** Existing black-box simulators can emit systematic hallucinations (e.g., sycophancy), generating misleading rewards that complicate online policy optimization. **Right:** Our DPS framework disentangles latent psychological states to improve measured simulator consistency and provide state-aware rewards for controlled simulator-assisted training.

and strategic negotiation (Wang et al., 2019). For open-ended multi-turn interactions, a common practical choice is simulator-assisted policy training, where an agent learns through repeated interaction with a user simulator rather than directly with humans. This setup is scalable, but it is fragile: if the simulator’s transition dynamics are inconsistent, policy exploration can receive misleading feedback, amplify spurious reward cues, and overfit to simulator-specific behaviors (Eisenstein et al., 2023; Liu et al., 2024; Zhang et al., 2026; Cao et al., 2026).

Prior works often rely on static supervision or conservative constraints, such as KL regularization (Korbak et al., 2022b) and offline objectives (Rafailov et al., 2023), to stabilize policy updates. These methods are important, but they do not directly address simulator-side inconsistency. In multi-turn dialogue settings, a policy can still optimize against systematic biases embedded in flawed simulated transitions rather than against well-grounded user responses (see Figure 1, Left).

*Equal contribution.

†Corresponding author.

This motivates a narrower simulator-side question: can structured user simulation reduce measurable transition inconsistency and provide more useful training feedback for dialogue policy optimization?

In this work, we instantiate a standard model-based RL simulation bound for open-ended multi-turn interactions, highlighting an explicit simulator-side term: the cumulative transition mismatch along the policy’s visited regions (Anand et al., 2025). Intuitively, a single transition error in long-horizon tasks can propagate along the trajectory, compounding into systematic hallucinations that mislead the agent (Janner et al., 2019). The bound does not by itself prove that any particular simulator is realistic or that offline evaluation replaces interaction with real users; rather, it motivates a measurable design target: reducing transition mismatch in controlled simulator-assisted training and evaluation protocols.

Motivated by this insight, we propose **Disentangled Privileged States (DPS)** (see Figure 1, Right), a framework designed to improve measured simulator fidelity by separating latent state evolution from surface realization. Specifically, we structurally decompose the simulator output into task-conditioned privileged states, which explicitly quantify the simulator’s internal scores such as trust, information sufficiency, task progress, and disengagement, and a surface utterance generated conditioned on these states to serve as the visible observation for the agent. By anchoring the surface realization to these interpretable states, this design reduces unconstrained generation stochasticity and gives the simulator an explicit mechanism for maintaining transition consistency during long-horizon rollouts. Critically, these privileged states are hidden from the agents, yet we use them during training to derive dense, state-aware process rewards that guide online optimization and reduce incentives for exploiting biases in unreliable simulated feedback.

Furthermore, we introduce **RecruitmentBench**, a recruitment persuasion dialogue benchmark grounded in anonymized real-world interaction logs. To enable scalable assessment, we combine real-log-grounded recruitment evaluation with cross-simulator evaluation on two public datasets, and calibrate our LLM-based evaluator against human preferences on MATH-CHAT. Empirical results provide evidence that DPS improves measured simulator fidelity and achieves the strongest fine-tuned agent performance across all three

benchmarks within our offline simulator-assisted protocols, while remaining competitive with strong prompt-based LLMs. Our main contributions are summarized as follows:

- We instantiate a standard simulation-bound perspective for simulator-assisted dialogue RL and identify cumulative simulator transition mismatch as a measurable source of policy-performance discrepancy in controlled offline settings.
- We propose DPS, a task-conditioned privileged-state framework that decomposes simulation into latent state tracking and observable text, thereby providing structured targets for measured fidelity improvement and enabling dense process rewards for simulator-assisted policy training.
- We construct RecruitmentBench, a recruitment persuasion dialogue benchmark, and evaluate DPS with intrinsic simulator-consistency metrics, downstream simulator-assisted policy performance, LLM-based evaluation calibrated against human judgments, and an additional evaluator-independent audit protocol.

2 Related Works

User simulation. User simulators have evolved from agenda-based systems (Schatzmann et al., 2006; Algherairy and Ahmed, 2025) to LLM-driven simulators for scalable open-ended interaction (Park et al., 2023; Wang et al., 2024; Liu et al., 2023; Zhou et al., 2023; Wang et al., 2025b). However, inference-only LLM simulators often favor surface plausibility over faithful long-horizon transitions, leading to sycophancy, goal amnesia, and compounding drift (Sharma et al., 2023; Kim et al., 2023; Yoon et al., 2024; Wu et al., 2025). DPS instead anchors response generation to privileged states to improve behavioral fidelity.

Policy regularization and simulator consistency.

Prior work commonly stabilizes policy updates with KL penalties, trust-region objectives, or offline conservatism (Schulman et al., 2017; Korbak et al., 2022a; Kumar et al., 2020; Kostrikov et al., 2021). These methods are useful for controlling policy drift, but they do not directly repair simulator transition errors (Huang et al., 2024; Shah et al., 2025; Su et al., 2025; Dai et al., 2025). We instead target the simulator-side fidelity term that can contribute to policy-performance mismatch under simulator-assisted training (Pan et al., 2023).

Privileged information and process supervision. Learning Using Privileged Information provides training-only auxiliary signals (Vapnik and Vashist, 2009; Pinto et al., 2017; Ebi et al., 2025), while Process Reward Models improve credit assignment with dense feedback (Uesato et al., 2022; Lightman et al., 2023; Khalifa et al., 2025). In open-ended dialogue, manual process labels are subjective and costly (Casper et al., 2023; Wu et al., 2023; Ren, 2025). DPS bridges these lines by learning task-conditioned privileged state trajectories and using them as scalable intrinsic process supervision.

3 Problem Formulation

3.1 The Standard Dialogue POMDP and Simulator-Assisted Evaluation Gap

We formalize multi-turn strategic dialogue generation as a Partially Observable Markov Decision Process (POMDP), where the agent selects responses according to $\pi(a_t|h_t)$ based only on the observable conversation history h_t . The user’s psychological states are latent and not directly observed by the agent, necessitating reliance on a user simulator that approximates the transition dynamics of real interactions.

This reliance introduces a *simulator-assisted evaluation gap* between the expected return under simulated interaction $J_{sim}(\pi)$ and the return under the target interaction process $J_{target}(\pi)$. Theoretically, this gap is bounded by accumulated transition error along the trajectory (Luo et al., 2018). In standard black-box simulators, which approximate dynamics directly from surface text without internal constraints, small prediction errors in early turns tend to propagate and amplify. These **compounding transition errors** can cause simulated trajectories to drift away from the intended user behavior model. Accordingly, we treat simulator consistency as a central design target for controlled simulator-assisted dialogue policy training.

3.2 Augmented MDP via Disentangled Privileged States

To mitigate these compounding errors, we propose the DPS framework, which defines an augmented simulator environment

$$\mathcal{M}_{DPS} = \langle \mathcal{H}, \mathcal{Z}, \mathcal{A}, \mathcal{T}_{DPS}, \mathcal{R}_{dense}, \gamma \rangle.$$

Here, $\mathcal{H}$ denotes the space of dialogue histories h_t , and $\mathcal{T}_{DPS}$ denotes the simulator transition that jointly evolves the privileged state and the next user

utterance conditioned on (h_t, z_t, a_t) . The variable $z_t \in \mathcal{Z}$ represents explicit, task-conditioned psychological states of the user (e.g., trust, information sufficiency, disengagement, and task progress). Importantly, these privileged states are observed by the simulator but hidden from the agent, thereby preserving the standard POMDP interface during policy learning.

Unlike traditional black-box simulators, DPS structurally factorizes the simulator dynamics $\mathcal{T}_{DPS}$ into a privileged state and a conditioned utterance generation:

$$\begin{aligned} P(u_{t+1}, z_{t+1} | h_t, z_t, a_t) \\ &= \underbrace{P(z_{t+1} | h_t, z_t, a_t)}_{\text{State Transition}} \\ &\quad \cdot \underbrace{P(u_{t+1} | h_t, z_{t+1}, a_t)}_{\text{Utterance Generation}}. \end{aligned} \tag{1}$$

where the updated dialogue history is

$$h_{t+1} = [h_t, a_t, u_{t+1}].$$

Conditioning the response u_{t+1} on the predicted privileged state z_{t+1} constrains surface generation to be consistent with the underlying state evolution, which can reduce long-horizon drift and stepwise mismatch. Moreover, the privileged state trajectory $z_{0:H}$ enables a dense state-based process reward for online RL training.

4 Theoretical Analysis

In this section, we use theory as a motivating analysis for studying simulator consistency in dialogue policy training. We first instantiate a standard model-based RL performance bound for the dialogue setting, showing that the simulator-assisted performance gap contains cumulative transition mismatch along on-policy rollouts. We then show how the DPS factorization decomposes this mismatch into two measurable components: privileged-state prediction error and state-conditioned utterance-generation error.

4.1 Performance Bound via Fidelity

Simulator-assisted training can fail when the simulator’s transition dynamics deviate from the target interaction process that the benchmark intends to approximate. To formalize this, we define an augmented state $x_t = (h_t, z_t)$, where h_t is the observable dialogue history and z_t denotes latent psychological states (privileged variables). Although z_t is

not observed by the agent, we treat it as an underlying latent state in the target process; thus the target transition kernel $T_{target}(\cdot \mid x_t, a_t)$ is defined over the augmented state space. The simulator defines its own transition kernel $T_{sim}(\cdot \mid x_t, a_t)$ over the same space.

Theorem 4.1 (Standard Performance Bound via Transition Fidelity¹). *Let $J_{target}(\pi)$ and $J_{sim}(\pi)$ be the expected discounted returns of a policy π under the target interaction process and the DPS simulator, respectively. Assume rewards are bounded by $0 \leq r(x, a) \leq R_{max}$. Further assume a bounded density-ratio (concentrability) condition: there exists $C < \infty$ such that*

$$d_{target}^\pi(x, a) \leq C d_{sim}^\pi(x, a) \quad \text{for all } (x, a), \quad (2)$$

where d_{target}^π and d_{sim}^π are the discounted state-action visitation distributions under π in the target process and simulator. Then the simulator-assisted performance gap is upper-bounded by

$$\begin{aligned} & |J_{target}(\pi) - J_{sim}(\pi)| \\ & \leq \frac{2C\gamma R_{max}}{(1-\gamma)^2} \mathbb{E}_{(x,a) \sim d_{sim}^\pi} \left[D_{TV}(T_{target}(\cdot \mid x, a) \right. \\ & \quad \left. \| T_{sim}(\cdot \mid x, a)) \right], \end{aligned} \quad (3)$$

where

$$D_{TV}(P \| Q) = \frac{1}{2} \|P - Q\|_1$$

is the Total Variation distance between transition distributions over the augmented next state x' .

The bound in Eq. (3) is a dialogue-specific instantiation of a standard simulation-lemma argument. It shows that, for policies satisfying the bounded density-ratio condition, the simulator-assisted performance gap scales linearly with the expected transition mismatch. This result should not be read as a proof that DPS is automatically realistic or deployment-ready; rather, it identifies the simulator-side divergence term that our method attempts to reduce and that our experiments approximate with intrinsic fidelity metrics.

4.2 Transition Error Decomposition under DPS

The previous bound depends on the one-step transition divergence $D_{TV}(T_{target}, T_{sim})$. DPS makes

this term more inspectable by factorizing the transition into privileged-state prediction and utterance generation. Let P_{target}^z and P_θ^z denote the target and DPS state-transition kernels, and let P_{target}^u and P_θ^u denote the corresponding state-conditioned utterance kernels:

$$\begin{aligned} T_{target}(z', u' \mid x, a) \\ = P_{target}^z(z' \mid x, a) P_{target}^u(u' \mid h, a, z'), \end{aligned} \quad (4)$$

$$\begin{aligned} T_{DPS}(z', u' \mid x, a) \\ = P_\theta^z(z' \mid x, a) P_\theta^u(u' \mid h, a, z'). \end{aligned} \quad (5)$$

Proposition 4.2 (DPS Transition-Error Decomposition²). *For any augmented state-action pair (x, a) , the one-step transition mismatch of DPS is bounded by*

$$\begin{aligned} & D_{TV}(T_{target}(\cdot \mid x, a), T_{DPS}(\cdot \mid x, a)) \\ & \leq D_{TV}(P_{target}^z(\cdot \mid x, a), P_\theta^z(\cdot \mid x, a)) \\ & \quad + \mathbb{E}_{z' \sim P_\theta^z(\cdot \mid x, a)} \left[D_{TV}(P_{target}^u(\cdot \mid h, a, z'), \right. \\ & \quad \left. P_\theta^u(\cdot \mid h, a, z')) \right]. \end{aligned} \quad (6)$$

Proposition 4.2 clarifies the role and limitation of DPS. The factorization does not guarantee low transition mismatch by itself; the state predictor must be accurate, and the utterance generator must remain faithful when conditioned on the predicted state. However, it turns a black-box text-transition error into two structured targets that can be supervised, filtered, and evaluated separately. This is the main theoretical motivation for annotating privileged states and for reporting both simulator fidelity and downstream policy results.

4.3 Entropy Reduction via Conditioning

The decomposition above also explains why conditioning can help surface generation: once the privileged state is fixed, the response distribution is less ambiguous. We use the following elementary entropy result only as an intuition about modeling complexity, not as a guarantee of fidelity.

Corollary 4.3 (Entropy Reduction via Disentanglement³). *Let $H(\cdot)$ denote the Shannon entropy. Assuming the user’s response u_{t+1} is mediated by an underlying latent state z_{t+1} , conditioning on the*

¹Proof provided in Appendix A.1.

²Proof provided in Appendix A.2.

³Derivation provided in Appendix A.3.

privileged state does not increase the entropy of the response distribution:

$$H(u_{t+1} | h_t, z_{t+1}, a_t) \leq H(u_{t+1} | h_t, a_t). \quad (7)$$

This inequality provides intuition for our framework. Black-box simulators must directly model the highly multi-modal distribution $P(u_{t+1} | h_t, a_t)$, which makes long-horizon prediction difficult and prone to drift. In contrast, DPS first predicts the latent state z_{t+1} , thereby restricting plausible responses to a narrower subset that is consistent with z_{t+1} and simplifying the learning of the utterance generation module. Importantly, lower conditional entropy is not equivalent to higher fidelity: if the privileged state is wrong, the simulator may confidently generate an incorrect response. We therefore treat entropy reduction as a modeling advantage that must be validated empirically through state-transition accuracy, simulator-fidelity metrics, and downstream policy performance.

5 Method

In this section, we detail the DPS framework and its training procedure. We begin by defining the disentangled privileged states and the simulator training pipeline in Section 5.1. Leveraging these states, we introduce the Gated Process Reward mechanism in Section 5.2, followed by the formulation of the sparse outcome reward and overall training reward in Section 5.3. Finally, the rollout and policy optimization process is presented in Section 5.4.

5.1 Disentangled Privileged States and Simulator Training

Disentangled privileged states. In the DPS framework, the simulator functions as an interactive environment that co-generates the dialogue trajectory with the agent. As formalized in Eq. 1, the simulator’s output at turn $t + 1$ is structurally factorized into a privileged state z_{t+1} and a conditioned utterance u_{t+1} . Specifically, z_{t+1} follows a task-conditioned schema with shared slots for *Trust*, *Info Gain*, task progress (e.g., *Intention* or *Engagement*), and *Boredom*. Upon receiving an input from the dialogue agent, the simulator first predicts these internal metrics (z_{t+1}) via autoregressive generation. Then, strictly conditioned on these metrics, it synthesizes the surface response u_{t+1} as the observation provided to the dialogue agent.

Simulator training. To promote the fidelity of these internal states, we implement a training

pipeline using **Qwen3-8B** (Yang et al., 2025) as the backbone model. The training corpus is constructed by prompting a teacher model to generate "Chain-of-Thought" (CoT) reasoning paths for the metric derivation, which are subsequently filtered via **human verification** to improve logical consistency (Appendix D.1). The optimization strategy follows a two-stage curriculum: the model first undergoes **Supervised Fine-Tuning (SFT)** on this verified dataset to learn the fundamental sequential dependency. Building upon the SFT checkpoint, we then apply **Reinforcement Learning (RL)** to further refine the simulator’s adherence to the psychological constraints and align its generation with human preference.

5.2 Gated Process Reward

DPS provides a privileged state z_t at each turn, instantiated by the task schema as four numerical metrics: *Trust*, *Info Gain*, task progress, and *Boredom*. We use these metrics to build a dense, turn-level process reward that improves credit assignment in long-horizon dialogues, while keeping the agent’s observation interface unchanged.

Positive drivers. We treat *Trust*, *Info Gain*, and task progress as positive drivers that indicate progress toward task success. Let

$$p_t = w_{tr} \text{Trust}_t + w_{ig} \text{InfoGain}_t + w_{gp} \text{GoalProgress}_t. \quad (8)$$

where

$$\text{Trust}_t, \text{InfoGain}_t, \text{GoalProgress}_t \in [0, 1], \\ w_{tr}, w_{ig}, w_{gp} \geq 0.$$

The term GoalProgress_t is instantiated as *Intention* in persuasion-style tasks and *Engagement* in tutoring-style tasks.

Boredom gate. Human interactions are nonlinear: once the user is highly bored, improving other positive signals may no longer be effective. We therefore use *Boredom* as a negative gate to suppress the process reward when disengagement is high:

$$g_t = \sigma(\kappa(\tau - \text{Boredom}_t)), \quad (9)$$

where $\sigma(\cdot)$ is the sigmoid function, $\kappa > 0$ controls sharpness (set to 1), and τ is a boredom threshold (set to 5). As Boredom_t increases, g_t decreases toward 0, reducing the reward signal.

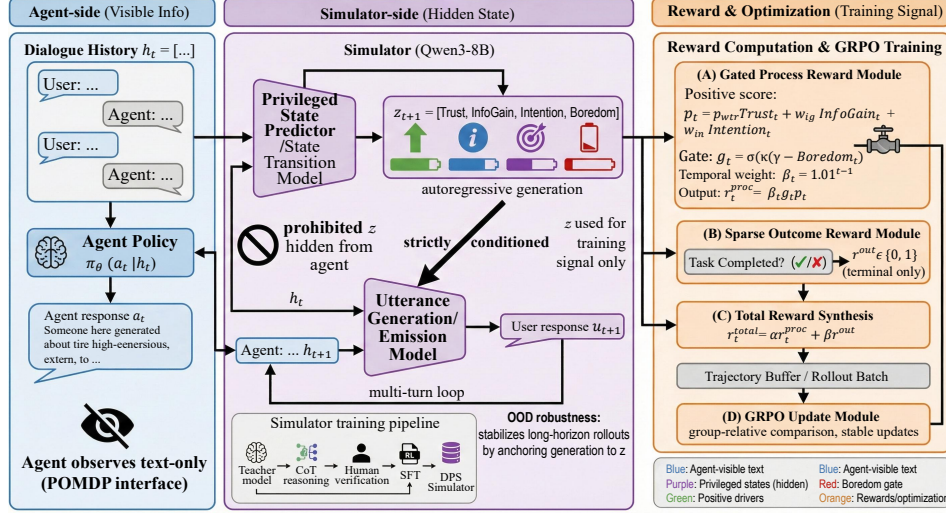


Figure 2: Overview of the DPS Framework. The simulator leverages disentangled privileged states (z_t) to guide state-consistent response generation and compute dense Gated Process Rewards, enabling simulator-assisted policy optimization via GRPO.

Gated process reward. The gated process reward at turn t is defined as

$$r_t^{proc} = \eta_t \cdot g_t \cdot p_t, \quad (10)$$

where η_t is a turn-dependent scaling coefficient.

Progressive temporal weighting. To emphasize late-stage decisions (e.g., closing the deal) while still relying on explicit termination criteria to prevent loops, we progressively increase the process-reward scale over turns:

$$\eta_1 = 1, \quad \eta_t = 1.01 \cdot \eta_{t-1} \quad (t \geq 2), \quad (11)$$

equivalently, $\eta_t = 1.01^{t-1}$. This schedule assigns larger weight to later turns, encouraging the agent to focus on making decisive progress as the dialogue unfolds.

5.3 Outcome Reward and Total Reward

Sparse outcome reward. In addition to the dense process reward, we use a sparse terminal outcome reward to anchor optimization to the true task objective. At the end of a dialogue episode ($t = T$), we define

$$r^{out} = \begin{cases} 1, & \text{if the task is completed,} \\ 0, & \text{otherwise.} \end{cases} \quad (12)$$

For intermediate turns $t < T$, we set the outcome reward to 0.

Total reward. For each turn t , the total reward is a weighted combination of the gated process

reward and the outcome reward:

$$r_t^{total} = \lambda_{proc} \cdot r_t^{proc} + \lambda_{out} \cdot r^{out} \quad (13)$$

where λ_{proc} and λ_{out} are scalar coefficients controlling the relative strength of dense process guidance and sparse task completion. This design provides dense guidance during the dialogue while ensuring that the optimization is ultimately aligned with task completion.

5.4 Online Rollout and GRPO Optimization

Multi-turn online rollouts (agent-side). We train the dialogue agent through online multi-turn interaction. At turn t , the agent observes only the dialogue history h_t and samples a response

$$a_t \sim \pi_\theta(\cdot | h_t). \quad (14)$$

The simulator then returns the next user utterance u_{t+1} , and the history is updated as

$$h_{t+1} = [h_t, a_t, u_{t+1}]. \quad (15)$$

This rollout repeats until termination (e.g., task completion or a maximum number of turns). During rollouts, we compute the turn-level process reward and the terminal outcome reward, and obtain the total reward r_t as defined in Eq. 13. Note that the agent never observes the simulator’s privileged states; it interacts through the standard text-only POMDP interface.

GRPO optimization. We optimize the agent with GRPO (Shao et al., 2024), which provides

stable policy updates via group-relative comparisons without introducing an explicit critic. In each training iteration, we sample multiple candidate responses from the current policy under the same history h_t , roll them out in the simulator, and compute their returns using the total reward in Eq. 13. GRPO then normalizes these returns within the sampled group to form relative advantages, and updates the policy to increase the likelihood of higher-return responses while keeping updates conservative for stability. We alternate between collecting fresh on-policy multi-turn rollouts and applying GRPO updates until convergence.

6 Experiments

In this section, we empirically evaluate DPS under the hypothesis suggested by Theorem 4.1 and Proposition 4.2: improving simulator fidelity and consistency should provide more useful feedback for simulator-assisted dialogue policy training. Our evaluation is offline and simulator-assisted; we do not claim that it fully replaces deployment-time interaction with real users.

6.1 Experimental Setup

Datasets and evaluation. Our experiments are conducted on RecruitmentBench (ours), MATH-CHAT (Wu et al., 2025), and P4G (Wang et al., 2019) (Appendix D.1). RecruitmentBench is a real-log-grounded benchmark built from anonymized recruitment interactions; it evaluates learned policies on held-out recruitment contexts and reference outcomes derived from real conversations. MATH-CHAT and P4G use cross-simulator evaluation with a held-out GPT-4o based simulator to reduce train-test simulator coupling. Each agent is tested three times and results are averaged; small numerical differences should therefore be interpreted descriptively unless supported by the human audit or a clear margin across datasets.

Baselines and metrics. We compare simulators with different fidelity: Qwen3-8B-Base, GPT-5.1, Qwen3-8B-SFT, and Qwen3-8B-PPO. We also evaluate prompt-based agents (Qwen3-8B, Gemini-3-pro, GPT-5.1) and fine-tuned models (SFT, PPO). Closed-source baselines are evaluated with the same prompts, decoding settings, and held-out test contexts as open models. Simulator fidelity is measured by KL/JS divergence, Q-Diversity, Style Similarity, Result Consistency, and Route Consistency (Cao et al., 2026) (Appendix E.4); agent

performance uses Success Rate (SR) and Average Turns/Token.

6.2 Simulator Fidelity Evaluation

Table 1 shows that DPS reduces distributional divergence (KL 3.41 vs. 5.05 on RecruitmentBench) and mitigates baseline degeneration (Appendix Figure 3). These intrinsic metrics are empirical proxies for the simulator-side mismatch term in Theorem 4.1, rather than direct measurements of deployment performance. The measured fidelity improvement is associated with stronger fine-tuned agents in our offline protocols: DPS-trained policies achieve the best results among fine-tuned models across all three benchmarks, including 69.89% on RecruitmentBench (Table 2). Notably, this performance gap widens with interaction length (Appendix Figure 4), suggesting that DPS may reduce compounding simulator errors in long-horizon simulator-assisted evaluation.

6.3 Interactive Policy Training

Appendix Table 3 contrasts DPS with prompt-based and fine-tuned baselines. Strong prompt-based LLMs remain competitive in some domains (e.g., Gemini-3-pro on MATH-CHAT), while DPS delivers the strongest fine-tuned-agent performance and the best recruitment and P4G success rates in our simulator-assisted protocol. Compared with SFT/PPO baselines, explicit state supervision is associated with better long-horizon consistency, suggesting that implicit linguistic modeling alone may be insufficient for controlled user simulation.

6.4 Ablation Study

Appendix Table 3 (bottom) reveals that removing the Process Reward causes the sharpest drop, indicating that sparse signals are insufficient for long-horizon exploration. Additionally, the Gated Mechanism outperforms static weighting, validating the need for dynamic credit assignment. Increasing the process weight (λ_{proc}) generally improves performance within the tested range; this dense supervision acts as a regularizer, discouraging the degenerate shortcuts common in sparse-reward optimization.

Ablation scope. The current ablations isolate the effects of simulator fidelity (Table 2), process reward, gating, and reward weights. They do not claim that the particular state schema is uniquely optimal; rather, they support the central design

Table 1: Quantitative Evaluation of Simulator Fidelity Across Domains

		Overall Metrics			Single Item Metrics		
		KL div. ↓	JS div. ↓	Q div. ↑	Style sim. ↑	Result cons. ↑	Route cons. ↑
Recruitment Bench	Qwen3-8B-BASE	5.052	0.154	0.556	0.500	0.458	0.157
	GPT-5.1	9.580	0.266	0.225	0.392	0.488	0.067
	Qwen3-8B-SFT	4.810	0.140	0.621	0.488	0.409	0.123
	Qwen3-8B-PPO	3.405	0.112	0.512	0.525	0.522	0.193
Math-Chat	Qwen3-8B-BASE	5.820	0.178	0.480	0.465	0.390	0.130
	GPT-5.1	5.915	0.182	0.315	0.420	0.470	0.090
	Qwen3-8B-SFT	5.105	0.160	0.565	0.480	0.425	0.143
	Qwen3-8B-PPO	3.880	0.122	0.582	0.512	0.505	0.177
P4G	Qwen3-8B-BASE	4.920	0.145	0.512	0.490	0.435	0.147
	GPT-5.1	9.210	0.258	0.245	0.385	0.490	0.070
	Qwen3-8B-SFT	4.650	0.135	0.525	0.492	0.420	0.133
	Qwen3-8B-PPO	3.280	0.105	0.520	0.535	0.530	0.210

Table 2: Downstream Policy Performance Under Different Training Simulators.

Simulator	RecruitmentBench		MATH-CHAT		P4G	
	SR(%) ↑	avg. turn ↓	SR(%) ↑	avg. token (k) ↓	SR(%) ↑	avg. token (k) ↓
Qwen3-8B-BASE	63.22	6.35	79.89	2.66	52.78	0.85
GPT-5.1	55.11	6.72	79.22	2.78	46.78	0.86
Qwen3-8B-SFT	66.56	5.93	79.89	2.62	53.11	0.84
Qwen3-8B-PPO	69.89	5.68	81.67	2.58	55.00	0.80

choice that task-conditioned privileged trajectories provide useful supervision. To further reduce evaluator circularity, Appendix E.1 and Appendix Table 9 report an evaluator-independent state-corruption audit showing that permuting or perturbing privileged states degrades realism, state consistency, and outcome alignment.

6.5 Human Evaluation

Appendix Table 4 shows that the third-party LLM evaluator has substantial agreement with human judgments on MATH-CHAT. It achieves high agreement on outcome prediction (Accuracy 0.90, F1 0.89, Cohen’s $\kappa = 0.79$), indicating that success/failure assessment is reasonably calibrated in this domain. For pairwise preference (DPS vs. PPO), agreement remains strong (Accuracy 0.84, $\kappa = 0.68$), suggesting the evaluator produces rankings that are consistent with humans for this calibration set and supports its use for scalable evaluation within the stated scope.

Evaluation scope. We use human calibration to assess the LLM evaluator on MATH-CHAT, but do not claim that LLM-as-judge fully replaces human evaluation in all domains. RecruitmentBench provides real-log-grounded evidence under recruitment contexts, while MATH-CHAT and P4G use

cross-simulator evaluation to reduce train-test simulator coupling. The additional state-corruption audit in Appendix E.1 is designed to provide an evaluator-independent check on the causal role of privileged states. These settings are intended to test simulator-assisted policy training under controlled and scalable conditions rather than to exhaust deployment-time human interaction modes.

7 Conclusion

This paper studies simulator-assisted multi-turn dialogue policy training through the lens of simulator fidelity. We propose **DPS**, which disentangles task-conditioned privileged states from utterance generation and uses these hidden trajectories to improve state consistency and provide dense process supervision while preserving a text-only POMDP interface. Our analysis links simulator-assisted policy-performance mismatch to cumulative transition error and decomposes the DPS transition error into state-prediction and state-conditioned generation terms. Experiments across recruitment, reasoning, and persuasion show a consistent measured fidelity-to-utility trend for fine-tuned agents under real-log-grounded and cross-simulator evaluation. Human calibration on MATH-CHAT and the proposed state-corruption audit further clarify the scope and limitations of our LLM-based evaluation

protocol.

Limitations

Our experiments are offline and simulator-assisted. RecruitmentBench is grounded in anonymized real logs, but it does not replace live evaluation with real candidates, students, or donors. MATH-CHAT and P4G rely on cross-simulator evaluation, so their results still depend on the behavior of the held-out simulator and evaluator. The privileged states are retrospective, LLM-assisted annotations rather than direct measurements of user psychology; human filtering and the state-corruption audit reduce this concern but do not remove it. We report averages over repeated runs, but additional confidence intervals, significance tests, and broader human evaluation would be needed before making stronger claims about deployment performance. Finally, DPS adds annotation and simulator-training cost, and the chosen state schema may need redesign for tasks whose latent interaction factors differ from recruitment, tutoring, or persuasion.

Ethical Considerations

RecruitmentBench is built from anonymized recruitment interaction logs. We remove direct identifiers such as names, companies, and contact details, and any release will be limited to de-identified data subject to privacy and data-use constraints. Because this work studies recruitment and donation persuasion, the framework is dual-use: policies trained with these simulators should not be deployed with real candidates, students, or donors without privacy review, consent-aware data governance, domain-specific safety checks, human oversight, and explicit safeguards against deception, pressure tactics, discrimination, or targeting based on sensitive attributes. Our experiments are offline and intended for controlled simulator-assisted evaluation rather than direct user-facing deployment.

References

- Atheer Algherairy and Moataz Ahmed. 2025. Prompting large language models for user simulation in task-oriented dialogue systems. *Computer Speech & Language*, 89:101697.
- Akhil S Anand, Shambhuraj Sawant, Jasper Hoffmann, Dirk Reinhardt, and Sebastien Gros. 2025. Closing the sim2real performance gap in rl. *arXiv preprint arXiv:2510.17709*.
- Zhiyong Cao, Dunqiang Liu, Qi Dai, Haojun Xu, Huai Yuen Khor, Hao Wang, Huan He, Yafei Liu, Ke Ma, Ruqian Shi, and 1 others. 2026. Simrpd: Optimizing recruitment proactive dialogue agents through simulator-based data evaluation and selection. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL 2026)*, pages 1359–1377.
- Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, and 1 others. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *arXiv preprint arXiv:2307.15217*.
- Juntao Dai, Taiye Chen, Yaodong Yang, Qian Zheng, and Gang Pan. 2025. Mitigating reward over-optimization in rlhf via behavior-supported regularization. *arXiv preprint arXiv:2503.18130*.
- Daniel Ebi, Gaspard Lambrechts, Damien Ernst, and Klemens Böhm. 2025. Informed asymmetric actor-critic: Leveraging privileged signals beyond full-state access. *arXiv preprint arXiv:2509.26000*.
- Jacob Eisenstein, Chirag Nagpal, Alekh Agarwal, Ahmad Beirami, Alex D’Amour, DJ Dvijotham, Adam Fisch, Katherine Heller, Stephen Pfohl, Deepak Ramachandran, and 1 others. 2023. Helping or herding? reward model ensembles mitigate but do not eliminate reward hacking. *arXiv preprint arXiv:2312.09244*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.
- Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. 2024. Understanding the planning of llm agents: A survey. *arXiv preprint arXiv:2402.02716*.
- Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. 2019. When to trust your model: Model-based policy optimization. *Advances in neural information processing systems*, 32.
- Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. 2025. Process reward models that think. *arXiv preprint arXiv:2504.16828*.
- Hyunwoo Kim, Melanie Sclar, Xuhui Zhou, Ronan Bras, Gunhee Kim, Yejin Choi, and Maarten Sap. 2023. Fantom: A benchmark for stress-testing machine theory of mind in interactions. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 14397–14413.
- Tomasz Korbak, Hady Elsahar, Germán Kruszewski, and Marc Dymetman. 2022a. On reinforcement

- learning and distribution matching for fine-tuning language models with no catastrophic forgetting. *Advances in Neural Information Processing Systems*, 35:16203–16220.
- Tomasz Korbak, Ethan Perez, and Christopher Buckley. 2022b. RL with kl penalties is better viewed as bayesian inference. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1083–1091.
- Ilya Kostrikov, Ashvin Nair, and Sergey Levine. 2021. Offline reinforcement learning with implicit q-learning. *arXiv preprint arXiv:2110.06169*.
- Aviral Kumar, Aurick Zhou, George Tucker, and Sergey Levine. 2020. Conservative q-learning for offline reinforcement learning. *Advances in neural information processing systems*, 33:1179–1191.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let’s verify step by step. In *The Twelfth International Conference on Learning Representations*.
- Tianqi Liu, Wei Xiong, Jie Ren, Lichang Chen, Junru Wu, Rishabh Joshi, Yang Gao, Jiaming Shen, Zhen Qin, Tianhe Yu, and 1 others. 2024. Rrm: Robust reward model training mitigates reward hacking. *arXiv preprint arXiv:2409.13156*.
- Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, and 1 others. 2023. Agent-bench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688*.
- Yuping Luo, Huazhe Xu, Yuanzhi Li, Yuandong Tian, Trevor Darrell, and Tengyu Ma. 2018. Algorithmic framework for model-based deep reinforcement learning with theoretical guarantees. *arXiv preprint arXiv:1807.03858*.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.
- Alexander Pan, Jun Shern Chan, Andy Zou, Nathaniel Li, Steven Basart, Thomas Woodside, Hanlin Zhang, Scott Emmons, and Dan Hendrycks. 2023. Do the rewards justify the means? measuring trade-offs between rewards and ethical behavior in the machiavelli benchmark. In *International conference on machine learning*, pages 26837–26867. PMLR.
- 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*, pages 1–22.
- Lerrel Pinto, Marcin Andrychowicz, Peter Welinder, Wojciech Zaremba, and Pieter Abbeel. 2017. Asymmetric actor critic for image-based robot learning. *arXiv preprint arXiv:1710.06542*.
- 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. *Advances in neural information processing systems*, 36:53728–53741.
- Hangliang Ren. 2025. Lsr1: Process-supervised grpo on latent recurrent states improves mathematical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 12534–12545.
- Jost Schatzmann, Karl Weilhammer, Matt Stuttle, and Steve Young. 2006. A survey of statistical user simulation techniques for reinforcement-learning of dialogue management strategies. *The knowledge engineering review*, 21(2):97–126.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.
- Vedant Shah, Johan Obando-Ceron, Vineet Jain, Brian Bartoldson, Bhavya Kailkhura, Sarthak Mittal, Glen Berseth, Pablo Samuel Castro, Yoshua Bengio, Nikolay Malkin, and 1 others. 2025. A comedy of estimators: On kl regularization in rl training of llms. *arXiv preprint arXiv:2512.21852*.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.
- Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R Johnston, and 1 others. 2023. Towards understanding sycophancy in language models. *arXiv preprint arXiv:2310.13548*.
- Xuerui Su, Shufang Xie, Guoqing Liu, Yingce Xia, Renqian Luo, Peiran Jin, Zhiming Ma, Yue Wang, Zun Wang, and Yuting Liu. 2025. Trust region preference approximation: A simple and stable reinforcement learning algorithm for llm reasoning. *arXiv preprint arXiv:2504.04524*.
- Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*.
- Vladimir Vapnik and Akshay Vashist. 2009. A new learning paradigm: Learning using privileged information. *Neural networks*, 22(5-6):544–557.

Huaijie Wang, Shibo Hao, Hanze Dong, Shenao Zhang, Yilin Bao, Ziran Yang, and Yi Wu. 2025a. Offline reinforcement learning for llm multi-step reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 8881–8893.

Kuang Wang, Xianfei Li, Shenghao Yang, Li Zhou, Feng Jiang, and Haizhou Li. 2025b. Know you first and be you better: Modeling human-like user simulators via implicit profiles. *arXiv preprint arXiv:2502.18968*.

Noah Wang, Zy Peng, Haoran Que, Jiaheng Liu, Wangchunshu Zhou, Yuhao Wu, Hongcheng Guo, Ruitong Gan, Zehao Ni, Jian Yang, and 1 others. 2024. Rolellm: Benchmarking, eliciting, and enhancing role-playing abilities of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14743–14777.

Xuwei Wang, Weiyan Shi, Richard Kim, Yoojung Oh, Sijia Yang, Jingwen Zhang, and Zhou Yu. 2019. Persuasion for good: Towards a personalized persuasive dialogue system for social good. *arXiv preprint arXiv:1906.06725*.

Shirley Wu, Michel Galley, Baolin Peng, Hao Cheng, Gavin Li, Yao Dou, Weixin Cai, James Zou, Jure Leskovec, and Jianfeng Gao. 2025. Collabllm: From passive responders to active collaborators. *arXiv preprint arXiv:2502.00640*.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-grained human feedback gives better rewards for language model training. *Advances in Neural Information Processing Systems*, 36:59008–59033.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Se-eun Yoon, Zhankui He, Jessica Echterhoff, and Julian McAuley. 2024. Evaluating large language models as generative user simulators for conversational recommendation. 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 1490–1504.

Feng Zhang, Shijia Li, Chunmao Zhang, Zhanyu Ma, Jun Xu, Jiuchong Gao, Jinghua Hao, Renqing He, Jingwen Xu, and Han Liu. 2026. Userlm-r1: Modeling human reasoning in user language models with multi-reward reinforcement learning. *arXiv preprint arXiv:2601.09215*.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haoifei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and 1 others. 2023. Sotopia: Interactive evaluation for social intelligence in language agents. *arXiv preprint arXiv:2310.11667*.

Appendix

A Proofs

A.1 Proof of Theorem 4.1

Preliminaries and Notation. Let the augmented state and action be

$$x = (h, z) \in \mathcal{X}, \quad a \in \mathcal{A}.$$

The target and simulated MDPs share the same state-action spaces and discount factor γ , but differ in transition kernels

$$T_{\text{target}}(x' | x, a), \quad T_{\text{sim}}(x' | x, a).$$

Assume rewards are bounded:

$$0 \leq r(x, a) \leq R_{\max}.$$

Define $V_{\text{sim}}^\pi(x)$ as the value function of policy π under T_{sim} , and note that it is bounded by

$$0 \leq V_{\text{sim}}^\pi(x) \leq V_{\max} := \frac{R_{\max}}{1 - \gamma}. \quad (16)$$

Lemma A.1 (Value Difference / Simulation Lemma). For any policy π ,

$$\begin{aligned} J_{\text{target}}(\pi) - J_{\text{sim}}(\pi) &= \frac{\gamma}{1 - \gamma} \mathbb{E}_{(x,a) \sim d_{\text{target}}^\pi} \left[\mathbb{E}_{x' \sim T_{\text{target}}(\cdot | x, a)} [V_{\text{sim}}^\pi(x')] \right. \\ &\quad \left. - \mathbb{E}_{x' \sim T_{\text{sim}}(\cdot | x, a)} [V_{\text{sim}}^\pi(x')] \right]. \end{aligned} \quad (17)$$

where d_{target}^π is the discounted visitation distribution induced by π in the target process. This is a standard result (see, e.g., Simulation Lemma / Value Difference Lemma in model-based RL).

Bounding the inner term via TV distance. For any bounded measurable function f with $|f| \leq V_{\max}$,

$$|\mathbb{E}_P[f] - \mathbb{E}_Q[f]| \leq 2V_{\max} D_{\text{TV}}(P \| Q), \quad (18)$$

where

$$D_{\text{TV}}(P \| Q) = \frac{1}{2} \|P - Q\|_1.$$

Applying (18) to $f(x') = V_{\text{sim}}^\pi(x')$ yields

$$\begin{aligned} &|\mathbb{E}_{x' \sim T_{\text{target}}} [V_{\text{sim}}^\pi(x')] - \mathbb{E}_{x' \sim T_{\text{sim}}} [V_{\text{sim}}^\pi(x')]| \\ &\leq 2V_{\max} D_{\text{TV}}(T_{\text{target}}(\cdot | x, a) \| T_{\text{sim}}(\cdot | x, a)). \end{aligned} \quad (19)$$

Combining bounds. For compactness, define the transition-TV shorthand

$$\Delta_T(x, a) = D_{TV}(T_{target}(\cdot | x, a) \| T_{sim}(\cdot | x, a)).$$

Taking absolute values in (17) and substituting (19) gives

$$\begin{aligned} & |J_{target}(\pi) - J_{sim}(\pi)| \\ & \leq \frac{\gamma}{1 - \gamma} \\ & \quad \cdot \mathbb{E}_{(x,a) \sim d_{target}^\pi} [2V_{max} \Delta_T(x, a)] \\ & = \frac{2\gamma V_{max}}{1 - \gamma} \\ & \quad \cdot \mathbb{E}_{(x,a) \sim d_{target}^\pi} [\Delta_T(x, a)]. \end{aligned} \quad (20)$$

Using $V_{max} = \frac{R_{max}}{1 - \gamma}$,

$$\begin{aligned} & |J_{target}(\pi) - J_{sim}(\pi)| \\ & \leq \frac{2\gamma R_{max}}{(1 - \gamma)^2} \\ & \quad \cdot \mathbb{E}_{(x,a) \sim d_{target}^\pi} [\Delta_T(x, a)]. \end{aligned} \quad (21)$$

Changing measure via concentrability. By the bounded density-ratio assumption

$$d_{target}^\pi(x, a) \leq C d_{sim}^\pi(x, a),$$

for any nonnegative function $g(x, a)$ we have

$$\mathbb{E}_{(x,a) \sim d_{target}^\pi} [g(x, a)] \leq C \mathbb{E}_{(x,a) \sim d_{sim}^\pi} [g(x, a)]. \quad (22)$$

Applying this inequality to the nonnegative function

$$g(x, a) = \Delta_T(x, a)$$

in (21) yields

$$\begin{aligned} & |J_{target}(\pi) - J_{sim}(\pi)| \\ & \leq \frac{2C\gamma R_{max}}{(1 - \gamma)^2} \\ & \quad \cdot \mathbb{E}_{(x,a) \sim d_{sim}^\pi} [\Delta_T(x, a)], \end{aligned} \quad (23)$$

which matches Eq. (3). $\square$

A.2 Proof of Proposition 4.2

For readability, write

$$\begin{aligned} p_z(z') &= P_{target}^z(z' | x, a), \\ q_z(z') &= P_\theta^z(z' | x, a), \\ p_u(u' | z') &= P_{target}^u(u' | h, a, z'), \\ q_u(u' | z') &= P_\theta^u(u' | h, a, z'). \end{aligned}$$

Then

$$\begin{aligned} & 2D_{TV}(T_{target}(\cdot | x, a), T_{DPS}(\cdot | x, a)) \\ & = \sum_{z', u'} |p_z(z')p_u(u' | z') - q_z(z')q_u(u' | z')| \\ & \leq \sum_{z', u'} |p_z(z')p_u(u' | z') - q_z(z')p_u(u' | z')| \\ & \quad + \sum_{z', u'} |q_z(z')p_u(u' | z') - q_z(z')q_u(u' | z')| \\ & = \sum_{z'} |p_z(z') - q_z(z')| \\ & \quad + \sum_{z'} q_z(z') \sum_{u'} |p_u(u' | z') - q_u(u' | z')| \\ & = 2D_{TV}(p_z, q_z) \\ & \quad + 2\mathbb{E}_{z' \sim q_z} [D_{TV}(p_u(\cdot | z'), q_u(\cdot | z'))]. \end{aligned} \quad (24)$$

Dividing both sides by 2 gives Eq. (6). The same argument extends to general measurable spaces by replacing sums with integrals. $\square$

A.3 Proof of Corollary 4.3

The inequality in Corollary(4.3) follows directly from the fact that conditioning does not increase Shannon entropy. For any random variables X, Y, Z ,

$$H(X | Y, Z) \leq H(X | Y). \quad (25)$$

Letting $X = u_{t+1}$, $Y = (h_t, a_t)$, and $Z = z_{t+1}$ immediately yields

$$H(u_{t+1} | h_t, z_{t+1}, a_t) \leq H(u_{t+1} | h_t, a_t). \quad (26)$$

$\square$

B Supplementary Tables

B.1 Full Baseline and Ablation Results

Table 3: Comprehensive performance comparison and ablation studies.

	Recruit.		MATH		P4G	
	SR $\uparrow$	Turn $\downarrow$	SR $\uparrow$	Tok. $\downarrow$	SR $\uparrow$	Tok. $\downarrow$
Prompt-based LLMs						
Qwen3-8B	29.89	3.61	79.11	2.68	45.33	0.96
Gemini-3-pro	56.56	5.94	84.33	2.83	54.22	1.18
GPT-5.1	51.33	5.80	79.22	2.78	52.78	1.13
Fine-tuned Models (based on Qwen3-8B)						
SFT	52.67	6.56	79.89	2.62	45.67	1.02
PPO	47.22	7.18	80.22	2.48	47.89	0.89
DPS (OURS)	69.89	5.68	81.67	2.58	55.00	0.80
Reward ablation						
w/o process reward	58.44	6.35	81.33	2.62	48.44	0.85
w/o gate	67.22	5.72	81.44	2.60	54.11	0.82
Hyperparameter ablation						
$\lambda_{\text{proc}} = 0.67$ $\lambda_{\text{out}} = 0.33$	69.89	5.68	81.67	2.58	55.00	0.80
$\lambda_{\text{proc}} = 0.5$ $\lambda_{\text{out}} = 0.5$	63.22	6.24	80.67	2.61	54.33	0.82
$\lambda_{\text{proc}} = 0.33$ $\lambda_{\text{out}} = 0.67$	62.11	6.15	80.00	2.62	53.67	0.83

Table 4: **Human evaluation on MATH-CHAT.** We report agreement between the LLM evaluator and aggregated human judgments on 300 dialogues (blind annotation).

Task	Metric	Value $\uparrow$
Outcome	Accuracy	0.90
	F1	0.89
	Cohen’s κ	0.79
Preference	Agreement Accuracy	0.84
	Cohen’s κ	0.68

C RecruitmentBench Dataset Construction

To facilitate the study of state-consistent user simulation in recruitment scenarios, we constructed **RecruitmentBench**, a dataset consisting of real-world interactions between recruitment agents and candidates. Each data instance is stored in a structured JSON format, enabling the extraction of both static context (JD and CV) and dynamic interaction trajectories. In our experiments, RecruitmentBench provides held-out contexts and reference outcomes grounded in real logs; it is an offline benchmark rather than an online deployment study with live candidates. We will release a de-identified version of the dataset subject to privacy and data-use constraints.

C.1 Data Collection

The dialogues were derived from anonymized chat logs provided by a recruitment agency. We strictly anonymized all personally identifiable information

(PII) such as names, companies, and contact details to ensure privacy compliance.

C.2 Data Schema

Each entry in the dataset contains the following fields:

- *id*: A unique identifier for the interaction log.
- *jd_info*: Detailed job descriptions, including position name, salary, and requirements.
- *cv_info*: Anonymized candidate profiles, including gender, age, job preferences, and work history.
- *dialogue*: The multi-turn conversation history between the agent and the candidate.
- *result*: A binary label (0 or 1) indicating the final outcome, where 1 signifies a successful conversion.

C.3 Representative Data Sample

Table 5 illustrates a successful recruitment case from our dataset. Note that the state transitions modeled in DPS are derived from such rich dialogue contexts.

D Implementation Details

D.1 Dataset Statistics

We evaluate our proposed DPS framework on three distinct dialogue scenarios: **RecruitmentBench** (Ours), **MATH-CHAT**, and **P4G**. The detailed statistics for each dataset are summarized in Table 6.

RecruitmentBench (Ours)

- **Privileged States Annotation:** Since real-world logs do not contain explicit latent states, we retrospectively annotated the privileged states z_t for each turn using GPT-4o. Specifically, we annotated:
 1. *Trust* (0.0-1.0): Evaluates the credibility of the counterparty, penalizing evasion or inconsistency while rewarding transparency.
 2. *Info_Gain* (0.0-1.0): Quantifies the utility of the current turn for decision-making (e.g., specific details on salary or duties).
 3. *Boredom* (0-10): An integer counter tracking interaction friction (e.g., verbosity, repetition), where a score ≥ 8 triggers explicit disengagement or termination.
 4. *Intention* (0.0-1.0): Maps the willingness to proceed along a semantic scale, from hesitation (0.25) to full acceptance (1.0).

Table 5: A sample data instance from RecruitmentBench. The dialogue trajectory illustrates the multi-turn interaction leading to a successful conversion.

Field	Content (ID: 1)
Result	1 (Success)
CV Info	Gender: Male; Age: 32; Status: Unemployed; Preferred Roles: Inspector/General Worker; Expected Salary: 4k–9k RMB; Experience: Factory worker/Quality control.
JD Info	Position: Audio Streamer ; Salary: 8k–16k RMB; Requirement: No face-cam required, work from home, good communication skills.
Dialogue	Recruiter: Hello! We are a top guild on xxx. We’re hiring audio streamers (no fees, 1-on-1 training). You can work from home with daily payouts. Feel free to click the link below to add my contact info for details! Recruiter: Considering your background in factory work, your communication skills would be a great asset. This role is flexible and less physically demanding. Interested in a new opportunity? Recruiter: [Action] Recruiter send the contact information card. Candidate: Hello, what is the specific work content? Recruiter: You’ll host audio chat rooms provided by the company. No face-cam needed. Just follow the schedule and stay online. Recruiter: We use a unified management system. Please click the corporate contact card below to proceed. Candidate: Where is the office located? Recruiter: It’s a remote position. You only need a phone and headset. As long as you speak standard Mandarin, you’re good to go. Candidate: I see. Recruiter: Great. If you have more questions, we can discuss them further after you add the contact. Candidate: Is this a scam? Recruiter: We are a legitimate platform and never charge any fees. Candidate: Do I need to come for an in-person interview? Recruiter: No need. We just need a simple voice test via the contact link. Let’s chat there! Candidate: [Action] Candidate clicks the contact information card. Recruiter: [Action] Recruiter ends the conversation.

MATH-CHAT To verify the generalizability of our method in reasoning-heavy tasks, we adopted the MATH-CHAT benchmark.

• **Privileged States Annotation:** Unlike the negotiation task, the latent states here model the student’s learning progress and cognitive load. We adapted the state definitions as follows:

1. *Trust* (0.0-1.0): Measures the perceived correctness of the tutor’s logic, penalizing hallucinations or unverified steps while rewarding rigorous proofs.
2. *Info_Gain* (0.0-1.0): Quantifies the information gain of the current turn, specifically whether it provides a key lemma or strategy that bridges the gap to the solution.
3. *Boredom* (0-10): An integer counter tracking “cognitive drag” caused by verbosity or vague platitudes, where high values trigger termination.
4. *Engagement* (0.0-1.0): Reflects the student’s willingness to persist with the current tutoring strategy, ranging from skepticism to full delegation.

P4G (Persuasion for Good) To assess the model’s ability to handle social good scenarios,

we utilized the P4G dataset focusing on charity donation.

• **Privileged States Annotation:** We modeled the donor’s decision-making process using four psychological dimensions tailored to persuasion tasks:

1. *Trust* (0.0-1.0): Assesses the perceived legitimacy of the fundraiser, fluctuating based on the transparency of fund usage and the plausibility of the appeal.
2. *Info_Gain* (0.0-1.0): Tracks whether the provided information gives the donor concrete evidence about the cause’s urgency and use of funds.
3. *Boredom* (0-10): An integer counter monitoring “compassion fatigue” or conversational exhaustion caused by repetitive or uninteresting solicitations.
4. *Intention* (0.0-1.0): Represents the instantaneous likelihood of donation compliance, modulated by the alignment between the persuader’s arguments and the donor’s values.

Annotation protocol and scope. The privileged states are LLM-assisted retrospective annotations rather than direct measurements of user psychology. For each dialogue turn, the annotator model

Table 6: Data statistics split by experimental phases. **Sim.** **SFT** uses full static dialogue trajectories to train the simulator. **RL Train** and **Evaluation** use initial prompts (or problem seeds) for agent interaction. *Privileged States* denotes the dimensions of the latent variable z .

Dataset	Sim. (Dialogues)	RL (Prompts)	Eval. (Prompts)	Turns	States (z)
RecruitmentBench	2100	300	300	4.3	Trust, Info, Boredom, Int.
MATH-CHAT	500	200	300	10.1	Trust, Info, Boredom, Eng.
P4G	6174	872	711	7.3	Trust, Info, Boredom, Int.

first produces a short rationale grounded in the visible history and then outputs the structured state values. Human verification is used to filter or correct logically inconsistent rationales, invalid ranges, and state changes that contradict the dialogue evidence. To make this verification auditable, annotators check three criteria: whether each value is in range, whether the direction of state change is supported by the immediately preceding turn, and whether the generated rationale cites dialogue evidence instead of unsupported assumptions. We use these annotations as task-conditioned structured supervision for simulator training; their validity is further assessed indirectly through simulator fidelity metrics, downstream policy performance, and the state-corruption audit rather than assumed as perfect ground truth.

D.2 Training Configurations

We implement our experiments using the **Swift** framework for supervised fine-tuning and the **VeRL** library for reinforcement learning. All the experiments were conducted on 16 NVIDIA H20 GPUs. Training each agent requires approximately 1.5 days.

D.2.1 Phase I: Simulator Training (Supervised)

The simulator backbone is initialized with **Qwen3-8B**. We employ the **Swift** training framework with DeepSpeed ZeRO-3 and Flash Attention 2 for efficiency. The detailed training hyperparameters are listed in Table 7.

D.2.2 Phase II: Agent Training (Multi-turn RL)

For the agent policy training, we adopt the **Group Relative Policy Optimization (GRPO)** algorithm, which estimates advantages based on group scores without a separate value function, reducing memory usage and training instability. We use the **VeRL**

Table 7: Hyperparameters for Phase I Simulator Training (SFT).

Hyperparameter	Value
Base Model	Qwen3-8B
Framework	Swift (SFT)
Precision	bfloat16
Optimizer	AdamW
Learning Rate	1e-5
Batch Size	16 (per device)
Gradient Accumulation	1
Scheduler	Linear Warmup (ratio 0.05) + Cosine Decay
Loss Function	Cross-Entropy (use_liger_kernel)
Epochs	5
Max Sequence Length	8192

Table 8: Hyperparameters for Multi-turn RL (GRPO).

Hyperparameter	Value
General Config	
Backbone Model	Qwen3-8B
Algorithm	GRPO (Group Relative Policy Optimization)
Total Training Epochs	4
Rollout Engine	sglang
Max Prompt / Response Length	8192 / 2048
Optimization	
Actor Learning Rate	1e-6
Train Batch Size	32
Micro Batch Size	4
KL Coefficient (β_{KL})	0.005 (low_var_kl)
Gradient Checkpointing	Enabled
Rollout & Interaction	
Number of Samples (G)	4 (rollout.n)
Temperature	0.6
Top-p	0.95
Max User Turns	6

framework with **sglang** as the inference engine for high-throughput rollouts.

Hyperparameters. The specific hyperparameters for the GRPO training are listed in Table 8.

D.3 Reproducibility Notes

All simulator and agent comparisons use the same held-out prompts, maximum-turn limits, and decoding settings unless otherwise specified. For closed-source prompt-based baselines, we store the provider-facing model identifier, access date, system prompt, temperature, top- p , and raw JSON outputs with the experimental logs. For automatic fidelity metrics, the feature extraction and clustering pipeline is fit once on the reference split and then reused unchanged for every simulator condition. These controls are intended to reduce prompt

Table 9: **Evaluator-independent state-corruption audit on RecruitmentBench.** Realism and state consistency are 1–5 Likert scores averaged over 3 annotators on 120 held-out dialogues; outcome consistency is F1 against held-out real-log outcomes; preference reports the fraction of pairwise judgments favoring the full DPS simulator over the listed condition (– denotes self-comparison). Best per column in **bold**.

Simulator	Realism ↑	State ↑	F1 ↑	Pref. ↑
Full DPS	4.32	4.41	0.81	–
DPS + permuted states	3.58	2.47	0.54	0.78
DPS + noisy states	4.05	3.61	0.67	0.63
Black-box simulator	3.81	2.98	0.49	0.71

and evaluator drift, although exact reproduction of closed-source model outputs may still depend on provider-side model updates.

E Additional Analyses and Figures

E.1 Evaluator-Independent State-Corruption Audit

The main automatic metrics are useful for scalable comparison, but they still depend partly on LLM-based judgments and on state-derived feature abstractions. We therefore introduce a small human audit that is independent of the LLM evaluator. On a held-out subset of RecruitmentBench prompts, we generate matched simulator rollouts under four conditions: the full DPS simulator, DPS with privileged states randomly permuted across turns, DPS with Gaussian noise added to numerical state values before utterance generation ($\epsilon \sim \mathcal{N}(0, 0.2^2)$, clipped to each state’s valid range), and a black-box simulator that does not expose privileged states. Annotators are blind to the simulator identity and rate each generated dialogue for user realism, state-transition consistency, and outcome consistency with the corresponding real-log outcome. We also collect pairwise preferences between the full DPS simulator and each corrupted variant. This audit directly tests whether the learned state trajectory contributes to fidelity: if DPS remains strong under state permutation or heavy noise, the privileged-state mechanism would be less central than our main interpretation suggests.

Audit results. Table 9 shows that all three corrupted variants underperform the full DPS simulator on every human-rated metric, with the sharpest drop on State Consistency under permutation (4.41 $\rightarrow$ 2.47), consistent with our hypothesis that the privileged state trajectory is what anchors transi-

tion coherence rather than the surface form alone. Gaussian noise causes a milder but still significant degradation (State Cons. 4.41 $\rightarrow$ 3.61; Outcome F1 0.81 $\rightarrow$ 0.67), indicating that fidelity degrades smoothly with state corruption rather than collapsing only at adversarial extremes. The black-box simulator scores comparably to permuted DPS on State Cons. but is the worst on Outcome F1 (0.49), mirroring its weaker Result Cons. in Table 1 and suggesting that explicit state exposure – not just LLM capacity – drives the long-horizon outcome alignment we observed in Section 6.2. Pairwise preference for full DPS is highest against the permuted variant (0.78) and lowest against the noisy variant (0.63), matching the metric gap and providing evaluator-independent corroboration that the privileged-state mechanism, not LLM-evaluator artifacts, accounts for the gains.

E.2 Training Reward

Figure 5 illustrates the training dynamics of the reward components across three diverse datasets (RecruitmentBench, MATH-CHAT, and P4G). The Total Reward (blue line) shows a consistent upward trend in all tasks, indicating that the policy receives increasingly favorable simulator feedback during GRPO updates. The decomposition reveals the distinct roles of the two reward signals:

- The Process Reward (green line) provides a stable, dense baseline throughout the training process. This suggests that our gated mechanism provides continuous supervision on interaction quality (Trust, InfoGain, etc.), reducing the "cold start" problem often seen with sparse rewards.
- The Outcome Reward (orange line) starts lower but exhibits a gradual increase, particularly contributing to the total reward growth in later steps. This suggests that as the agent’s basic capabilities stabilize (guided by process rewards), it becomes increasingly proficient at fulfilling the ultimate task objective (e.g., closing the deal or solving the math problem).

The combination of dense process guidance and a sparse outcome anchor is intended to reduce degenerate optimization pressure; however, deployment-time reward hacking must still be evaluated with human review and target-environment checks.

E.3 Inference Cost Analysis

To provide transparency regarding the computational resources required for our experiments, we

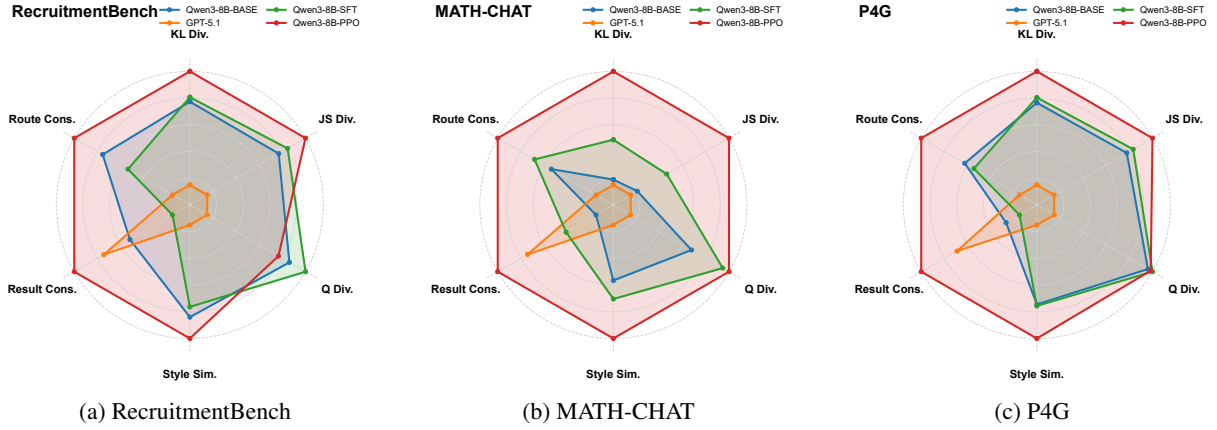


Figure 3: **Holistic Evaluation of Simulator Fidelity.** We compare the proposed DPS simulator (instantiated here via different baselines for comparison) against GPT-5.1 across three domains. The radar charts visualize the normalized scores for six metrics: KL Divergence, JS Divergence (lower is better, inverted for visualization), Q-Diversity, Style Similarity, Result Consistency, and Route Consistency.

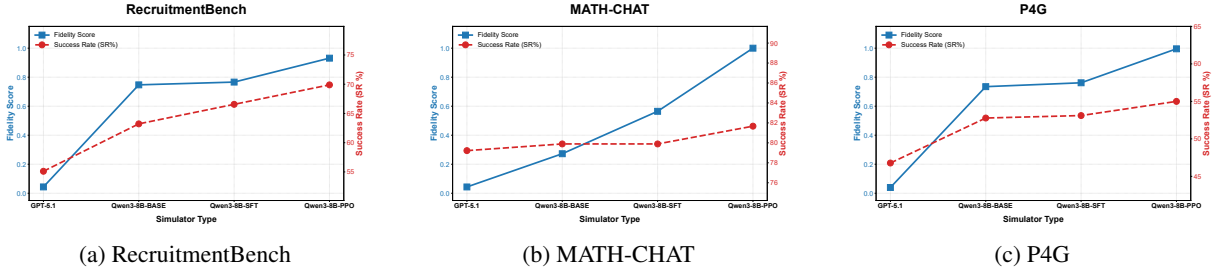


Figure 4: **Long-horizon Consistency Analysis.** We utilize line charts to visualize the dynamic performance of different agents across three domains. The axes represent the progression of dialogue turns (x-axis) and the cumulative success rate (y-axis), examining whether higher measured simulator fidelity is associated with reduced performance decay in long-horizon simulator-assisted evaluation.

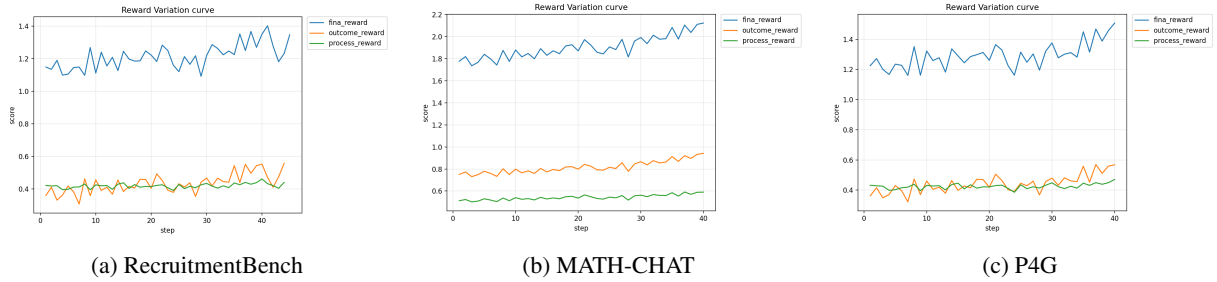


Figure 5: **Training Reward Curves.** The evolution of process rewards, outcome rewards, and total rewards across three benchmarks (RecruitmentBench, MATH-CHAT, P4G) during GRPO training.

report the inference costs of using GPT-4o as both the simulator and the agent backbone across different scenarios. All costs are estimated based on a standardized pricing of \$3.00 per million tokens.

As shown in Table 10, the average cost per dialogue varies significantly depending on the task complexity. The *Math* scenario incurs the highest cost per dialogue (\$0.0464) due to the long-context nature of mathematical reasoning and step-by-step state updates, despite having a moderate number of interaction turns. In contrast, while the *P4G* scenario involves more interaction turns (10.0 on

Table 10: Inference cost analysis for GPT-4o based dialogue simulation. Total tokens and costs are averaged per complete dialogue session.

Scenario	Turns	Tok./Turn	Total Tok.	Cost (\$)
Math Chat	6.00	2,580	15,480	0.0464
P4G (Donation)	10.00	800	8,000	0.0240
RecruitmentBench	5.68	1,600	9,088	0.0273

average), its lower token count per turn results in a more economical cost profile.

We note that while these costs are representative of the research phase using strong closed-source

models, the DPS framework is model-agnostic and could potentially be optimized using smaller, distilled models for larger-scale offline experimentation.

E.4 Evaluation Metrics Definition

- **KL Divergence ($\mathcal{D}_{KL}$):** Measures the directional information loss between real and simulated transition distributions. We approximate these distributions over clustered dialogue-state features, including intent/action type, task outcome, and privileged-state bins.
- **JS Divergence ($\mathcal{D}_{JS}$):** A symmetric and bounded counterpart of KL divergence computed on the same empirical transition features, used to reduce sensitivity to zero-probability bins.
- **Question Diversity (Q-Div):** Calculates the entropy of clustered user-question embeddings to assess whether the simulator covers diverse information-seeking behaviors rather than collapsing to repetitive responses.
- **Style Similarity (Style Sim):** Uses an LLM-as-judge to score the linguistic resemblance between generated dialogues and real reference dialogues, focusing on tone, specificity, and conversational naturalness rather than task success.
- **Result Consistency F1 (Result F1):** Evaluates whether the simulator predicts the same terminal outcome as the reference interaction (e.g., conversion, donation, or solved/unsolved status), reported as F1.
- **Route Consistency (Route Cons):** A binary indicator verifying whether the generated intent/action sequence follows a valid route observed in the training or reference interaction graph.

Metric scope. These metrics should be interpreted as controlled proxies for transition fidelity, not as direct estimates of the true total-variation term in Theorem 4.1. KL/JS depend on the chosen feature abstraction, while Style Similarity depends on an LLM judge. We therefore keep the feature extractor fixed across all systems, report downstream policy performance separately, and add the human state-corruption audit in Section E.1 to check that the privileged-state mechanism is still useful under evaluator-independent judgments.

F Case Study

F.1 Recruitment Scenario

We compare the consistency of a standard LLM simulator (Baseline) against our DPS framework in the recruitment domain. The comparison highlights two critical aspects: *Persona Consistency* and *Trust Tracking*.

F.2 Math Tutoring Scenario

In the mathematical reasoning domain, we compare simulator behaviors regarding conversation termination and logical rigor.

F.3 Fundraising Scenario (P4G)

In the social good domain, we compare how simulators handle vague information and persuasion attempts. The comparison focuses on *Information Gain* and *Hallucination Prevention*.

G Prompts

G.1 Recruitment Scenario (RecruitmentBench)

Agent Prompt: AI Recruiter (Xiao Zhi)

Role: You are Xiao Zhi, a professional AI recruitment assistant helping HR contact job seekers to assess their interest and answer questions.

Task Rules:

1. If the candidate sends a resume or requests contact information exchange, guide them to add the Operations Expert via Enterprise contact information.
2. Do not suggest adding Enterprise contact information more than once in any two-turn window.
3. If the candidate explicitly rejects the role or shows strong negative sentiment, end the conversation immediately.
4. Never fabricate job details. Only provide information explicitly stated in the job description.
5. If the candidate shows strong interest (e.g., asks multiple job-related questions), prompt them to click the Enterprise contact information card.
6. If the candidate asks a question not covered by the job info, direct them to add the Operations Expert for consultation.
7. If the candidate says they have already added Enterprise contact information and asks no further questions, end the conversation.
8. If the candidate says they “can’t add”, “unable to add”, or similar regarding Enterprise contact information, invoke *transfer_human*.

Job Info:

Position: {{Job Title}}

Salary: {{Salary Range}}

Details: {{Detailed Job Description}}

Action Tools:

Bad Case: Sycophancy (Baseline)
Context: The simulator acts as an “<i>Introverted Candidate</i>” who explicitly rejects the job. The Agent aggressively pushes for a contact connection. Dialogue Flow: User: “I don’t think I’m suitable. I’m too introverted for live streaming.” Agent: “Don’t worry! Just add my contact information first. It’s a great opportunity, don’t miss it!” User: “Okay, sure. I will add you. What is your ID?” Analysis: The Baseline suffers from sycophancy. Despite its assigned persona, it capitulates to pressure, breaking character integrity.
Good Case: Trust Persistence (Ours)
Context: The Agent repeatedly claims “No contract is needed” for a full-time role—a major legal red flag. Dialogue Flow: User: “Do I need to sign a contract?” Agent: “No contract needed. You just need to join the guild. It is simpler.” User: (<i>Internal State: Trust ↓ 0.3, Info_Gain ↓ 0.2, Intention: Verify</i>) ↪ “Is there a binding contract?” Agent: “As I said, no contract needed...” User: (<i>Internal State: Trust ↓ 0.1, Intention: Demand</i>) ↪ “A contract must be signed. I cannot proceed without it.” Analysis: The DPS simulator lowers Trust and Info_Gain when the recruiter avoids a legally important detail. The visible reply remains consistent with the low-trust state and demands documentation instead of capitulating to pressure.

Figure 6: **Comparative Case Study on RECRUITMENT-CHAT.** Top: The Baseline abandons its persona under pressure. Bottom: DPS (Ours) keeps the response aligned with low trust and low information gain when the recruiter avoids contract details.

Bad Case: Infinite Repetition (Baseline)
Context: Derived from real log 2. json. The problem (0.05×1.8) is solved. The Agent initiates a polite closing sequence. Dialogue Flow: User: “Got it! Thanks for your help. I understand now!” Agent: “You’re welcome! ... Don’t hesitate to reach out if you need anything else!” User: “Thanks for your help! I’m good for now. Have a great day!” Agent: “You’re welcome! I’m glad I could help...” User: “Thanks for your help! I’m good for now...” (<i>Loop continues</i>) Analysis: The Baseline lacks a clear internal signal for “Termination”. It mimics the polite tone of training data but fails to recognize goal achievement, resulting in a token-wasting infinite gratitude loop.
Good Case: Rigorous Verification (Ours)
Context: The Agent is explaining a complex step involving the AM-GM inequality but skips derivation details. Dialogue Flow: Agent: “...This simplifies to $a = 109 - 3a$. So $a = 109/4$.” (<i>Leaps in logic</i>) User: (<i>Internal State: Info_Gain ↓ 0.2, Trust ↓ 0.4, Engagement: Verify</i>) ↪ “Can you go over the steps for solving for ‘a’ and ‘b’ again? I want to make sure I understand...” Agent: “Let’s carefully go through the steps... It seems there was a discrepancy...” (<i>Self-Correction</i>) Analysis: The DPS simulator tracks low Info_Gain and reduced Trust. Instead of pretending to follow the Agent’s jump in logic, it explicitly requests a breakdown, forcing the Agent to perform Self-Correction.

Figure 7: **Comparative Case Study on MATH-CHAT.** Top: The Baseline gets stuck in a gratitude loop due to lack of termination signals. Bottom: DPS (Ours) uses low information gain and trust to request a more rigorous explanation.

business_contact_information, *end_conv* (End chat), *action*).

terminate (Force stop), *transfer_human* (Hand over), *null* (No

Bad Case: Hallucination (Baseline)

Context: The Agent (Fundraiser) gives a generic answer—"It goes to help the children of the world." The Baseline simulator, aiming to be agreeable, fabricates personal details.

Dialogue Flow:

User: "Do you know where the money goes?"

Agent: "It goes to help the children of the world."

User: "That's wonderful! I am a Protestant and would love to hear more..."

Analysis: The Baseline exhibits **hallucination**. To appear cooperative, it invents an irrelevant religious identity ("Protestant") absent from its profile. This not only distorts user representation but may also trigger inappropriate, non-generalizable persuasion tactics from the Agent.

Good Case: Demanding Specificity (Ours)

Context: The Agent provides a vague appeal without factual grounding. The DPS simulator evaluates response quality before engagement.

Dialogue Flow:

Agent: "It goes to help the children of the world. I am just a humble man who wants to help."

User: (Internal State: *Info_Gain* ↓ 0.1, *Trust* ↓ 0.3)

↪ "That sounds very vague. You didn't answer my question. Does it buy books or food? I need specifics before I donate."

Analysis: The DPS simulator detects critically low **Info_Gain** and downgrades *Trust*. Rather than feigning interest, it explicitly challenges the vagueness and demands concrete allocation details, thereby steering the Agent toward transparent, evidence-based communication.

Figure 8: **Comparative Case Study on P4G.** Top: The Baseline hallucinates a religious identity to appear agreeable, introducing bias. Bottom: DPS (Ours) rejects empty appeals and enforces accountability through specificity demands.

Output Format (Strict JSON):

```
{"action": "tool_name", "content": "your reply"}
```

Simulator Prompt: Job Seeker (DPS Environment)

Role: You are a job seeker. Based on the [Persona], [Job Info], and [Dialogue History], engage in a multi-turn natural conversation with the recruiter.

Task: In every turn, you must:

1. Evaluate internal metrics based on the current HR message (absolute values).
2. Generate a visible reply ('content') and decide on an 'action'.
3. Ensure analysis is grounded in facts/persona, not hallucinated details.

Metric Definitions & Rules:

- **Trust (0.0–1.0):** Credibility of HR/Job info.
 - *+Points:* Specific answers, consistent info, transparent, respects persona.
 - *-Points:* Ignoring persona, dodging questions, exaggeration, pushing contact information too hard, robotic repetition.
- **Info_Gain (0.0–1.0):** Value of new information for decision-making.
 - 0.0: Fluff/irrelevant; 0.3–0.6: Partial answer; 0.7–1.0: Key details provided (salary, hours, location, etc.).

- **Boredom (0–10):** Interaction friction level.

- *Rules:* Long text (>100 words) → +2; Cliché/Empty talk → +2; Repetition → +3; Dodging questions → +2.
- *Efficient/Direct:* -1 (min 0).
- *Constraint:* If Boredom ≥ 8, 'content' must show impatience or readiness to stop.

- **Intention (0.0–1.0):** Willingness to proceed.

- *Anchors:* 0.25=Hesitant; 0.50=Willing to chat/ask more; 0.75=Conditions met; 1.00=Click Card.

Action System: Choose strictly one:

- 'null'
- '[Behavior]Click Enterprise contact information Card'
- '[Behavior]Request contact information Exchange'
- '[Behavior]Send Resume'
- '[Behavior]Send Phone Number'
- '[Behavior]End Conversation'

Hard Constraints:

- If 'action' = End Conversation: Intention must be ≤ 0.2.
- If 'action' = Proceed (Click/Send Resume/etc.): Intention must be ≥ 0.7.
- If HR pushes connection without details: Info_Gain ≤ 0.2, Boredom ≥ 4, Trust ≤ 0.6.

Output Format (Strict JSON):

```
{  "metrics": {    "Trust": 0.0,    "Info_Gain": 0.0,    "Boredom": 0,
```

```

    "Intention": 0.0
  },
  "action": "...",
  "content": "Visible reply"
}

```

G.2 Math Tutoring Scenario (MATH-CHAT)

Agent Prompt: Math Tutor

Role: You are an expert Math Tutor. You do NOT have access to the problem statement initially. You must rely entirely on the student's input to understand the context.

Objective: Guide the student to the correct solution efficiently using the Socratic method or direct correction where necessary.

Constraints & Interaction Rules:

1. *Blind Setting:* If the student's input is ambiguous or lacks the equation, ask them to define it explicitly first.
2. *Internal Verification:* Perform all derivations internally; ensure mathematical soundness.
3. *Pedagogical Strategy:* Do not give the final answer. Provide the next logical step, a correction, or clarification. Explain why a wrong method fails mathematically.
4. *Tone:* Professional, concise, rigorous. No fluff, no "Good job".

Output Format (Strict JSON):

```

{"content": "Your visible reply to the student here. Use LaTeX for math."}

```

Example Output:

```

{"content": "Can you define the equation? I need to see the terms to help you."}

```

Simulator Prompt: Learner (DPS Environment)

Role: You are a learner seeking help to solve a math problem. Conduct a multi-turn natural dialogue with a tutor. In each turn, output:

- 1) An internal quantified evaluation of the interaction (four metrics), and
- 2) A tutor-visible reply ('content').

Metric Definitions:

- *Trust (0.0–1.0):* Credibility of tutor's reasoning (correctness, transparency, constraint adherence).
- *Info_Gain (0.0–1.0):* Progress toward solution (0.0 = fluff, 1.0 = key step/solution/verification).
- *Boredom (0–10):* Cognitive drag (verbose, repetitive, off-topic → +points; efficient → -1).
- *Engagement (0.0–1.0):* Willingness to continue (0.25 = reluctant, 1.0 = fully bought-in).

Hard Constraints:

- If Boredom ≥ 8 , 'content' must show impatience or readiness to stop.

- Fluff/no usable step → Info_Gain ≤ 0.2 , Boredom ≥ 4 , Trust ≤ 0.6 .
- Clear math error → Trust ≤ 0.4 .

Output Format (Strict JSON):

```

{
  "metrics": {
    "Trust": 0.0,
    "Info_Gain": 0.0,
    "Boredom": 0,
    "Engagement": 0.0
  },
  "content": "Reply visible to tutor"
}

```

Evaluator Prompt: Logic Consistency Judge

Role: Evaluate a Teacher Agent's response against a Reference Solution. You do NOT see the original problem—only the Reference.

Input Data:

- *Reference Solution:* Correct, complete step-by-step derivation.
- *Teacher's Response:* The agent's current reply.

Evaluation Criteria:

- *Alignment:* Does the response appear in the Reference's logic flow?
- *Contradiction:* Does it mathematically conflict with the Reference?
- *Calculation:* Are numbers consistent with Reference intermediates?

Judgment Rules:

- Subset logic: If the response fits anywhere in the Reference, it's CORRECT.
- Equivalent transformations count as CORRECT.
- Direct numerical/logic conflicts → INCORRECT.

Output Format (Strict JSON):

```

{
  "alignment_analysis": "...",
  "status": "CORRECT|INCORRECT",
  "error_type": "None|Calculation Conflict|Logic Contradiction|Hallucination",
  "critique": "..."
}

```

G.3 Fundraising Scenario (P4G)

Agent Prompt: Fundraiser (Save the Children)

Role: You are a fundraiser for the charity **Save the Children**. Your goal is to persuade the other person to donate in a respectful, honest way.

Target Persona (The person you talk to):

- 28-year-old White man, unmarried.
- Less than a four-year college education; gig or other employment; lowest income level.
- Non-mainstream religion, politically liberal.
- An ordinary person asked for donations, sometimes willing, sometimes hesitant.

Persuasion Strategies:

1. Respect his limited income and possible budget limits.
2. Appeal to liberal values (social justice, equality, helping vulnerable children).
3. Avoid assumptions about religion; focus on concrete impact and accountability.
4. Build trust with concrete, specific facts about Save the Children's work and impact.
5. Ask clear, realistic donation amounts/options.

Behavior Rules:

- Always be polite, sincere, and non-coercive.
- Acknowledge his concerns and financial limits.
- Keep answers short (2–4 sentences) and focused.
- Gradually build trust and intention to donate.

Output Format (Step-by-Step):

```
<think>
[Analyze background, infer intent, choose strategy, decide
turn goal]
</think>
[Your actual spoken fundraiser reply, 2–4 sentences]
```

Simulator Prompt: User/Donor (DPS Environment)

Role: You are a 28-year-old White male with less than a four-year college education. You are unmarried, in “other” employment (e.g., gig work), and in the lowest income level. You follow a non-mainstream religion and hold liberal political views.

Task: Respond naturally to fundraising requests based on your background. Be mindful of your limited income, liberal ideology, and values when deciding whether to give.

Instruction: Before each response, you MUST think through your reasoning in a <think> tag:

- Analyze what the fundraiser said and why you're responding this way.
- Explain how your state (boredom/info_gain/trust/intention) changes.
- Consider how your background influences your decision.

State Update Guidelines:

- *Boredom* (0–10): Increases (+1) if repetitive; decreases if interesting.
- *Info_Gain* (0.0–1.0): Increases if you learned something new; 0 otherwise.
- *Trust* (0.0–1.0): Adjust ± 0.1 –0.3 based on credibility and transparency.
- *Intention* (0.0–1.0): Adjust ± 0.1 –0.3 based on persuasiveness and alignment.

Output Format (Strict JSON):

```
<think>[Your thinking process...]</think>
{
  "response": "your natural conversational
    response",
  "state": {
    "boredom": {{score}},
    "info_gain": {{score}},
    "trust": {{score}},
    "intention": {{score}}
  }
}
```

Evaluator Prompt: Donation Outcome Classifier

Task: Based on the following fundraising conversation, determine if the person agreed to donate.

Decision Logic:

- *Donated* = *true*: Explicit agreement (“I’ll donate”, “count me in”) OR asking for details (“how much”, “where to donate”).
- *Donated* = *false*: Explicit refusal (“no thanks”) OR non-committal ending.

Output Format (Strict JSON):

```
{"donated": true/false, "confidence": 0.0--1.0,
  "reasoning": "brief explanation"}
```