INFOPO: INFORMATION-DRIVEN POLICY OPTIMIZATION FOR USER-CENTRIC AGENTS

Fanqi Kong[1,2], **Jiayi Zhang**[2,3], **Mingyi Deng**[2], **Chenglin Wu**[2], **Yuyu Luo**[3], **Bang Liu**[4]
[1]Peking University [2]DeepWisdom
[3]The Hong Kong University of Science and Technology (Guangzhou)
[4] Université de Montréal & Mila
kfq20@stu.pku.edu.cn bang.liu@umontreal.ca

ABSTRACT

Real-world user requests to LLM agents are often underspecified. Agents must interact to acquire missing information and make correct downstream decisions. However, current multi-turn GRPO-based methods often rely on trajectory-level reward computation, which leads to credit assignment problems and insufficient advantage signals within rollout groups. A feasible approach is to identify valuable interaction turns at a fine granularity to drive more targeted learning. To address this, we introduce InfoPO, which frames multi-turn interaction as a process of active uncertainty reduction and computes an information-gain reward that credits turns whose feedback measurably changes the agent's subsequent action distribution compared to a masked-feedback counterfactual. It then combines this signal with task outcomes via an adaptive variance-gated fusion to identify information importance while maintaining task oriented goal direction. Across diverse tasks including intent clarification, collaborative coding, and tool-augmented decision making, InfoPO consistently outperforms prompting and multi-turn RL baselines. It also demonstrates robustness under user simulator shifts and generalizes effectively to environment interactive tasks. Overall, InfoPO provides a principled and scalable mechanism for optimizing complex agent user collaboration.

1 INTRODUCTION

The rapid evolution of Large Language Models (LLMs) has enabled interactive agents that assist users in complex, multi-turn tasks (Team et al., 2025; Liu et al., 2025; Li et al., 2025a). In user-centric settings, agents must bridge a fundamental gap between often ambiguous human intentions and the precise parameters required for machine execution. For example, a request like "book me a flight next week" is not directly actionable until the agent elicits missing constraints such as dates, departure airport, budget, and flexibility. Therefore, an effective interaction should both increase the agent's knowledge about the user's true intent and advance the task toward completion. Mastering this interplay between intent elicitation and task execution remains a core challenge for building autonomous agents that operate reliably under partial information (Deng et al., 2025; Qian et al., 2025b; He et al., 2025).

Reinforcement learning (RL) has become a primary paradigm for improving LLM-based agents (Zhang et al., 2025). However, multi-turn agentic tasks introduce a long-horizon credit assignment bottleneck. Rewards are often sparse and delayed until task completion, making it difficult to attribute outcomes to intermediate decisions (Zhou et al., 2025; Wei et al., 2025). This issue is particularly pronounced under GRPO-based methods (Shao et al., 2024; Yu et al., 2025), where policy updates rely on reward variation within a rollout group. Moreover, many recent works on multi-turn RL aggregate terminal and intermediate signals into a single trajectory-level score for advantage estimation, which limits fine-grained supervision across the interaction (Qian et al., 2025a;c; Wang et al., 2025; Jin et al., 2025). In user-centric environments, such granularity matters even more. A small number of clarification decisions can determine feasibility and downstream success, and RL training frequently relies on LLM-simulated users (Zhao et al., 2025b; Cai et al., 2025; Li et al., 2025b), making sample efficiency also a critical constraint.

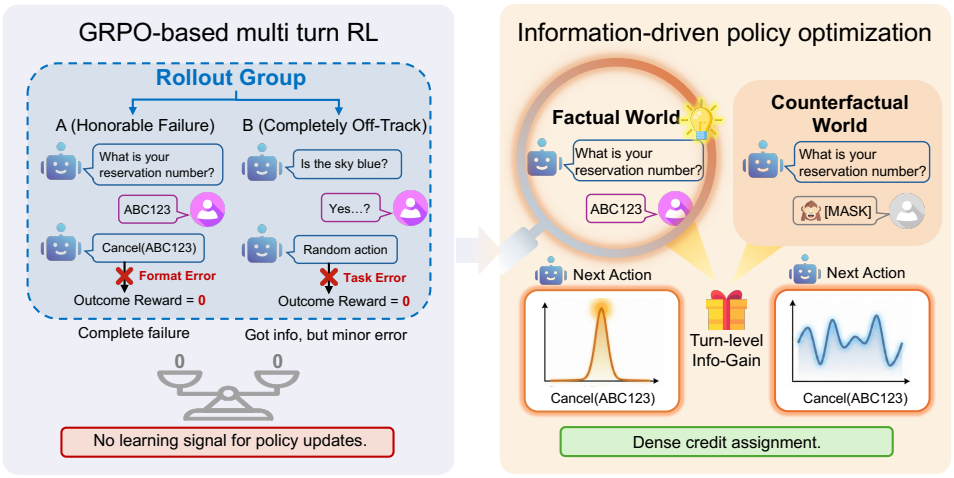


Figure 1: Comparison between standard GRPO-based multi-turn RL and our InfoPO.

To address the above limitation, we propose **InfoPO** (Information-Driven Policy Optimization), which treats multi-turn interaction as a process of active uncertainty reduction (as illustrated in Figure 1). InfoPO defines a turn-level *counterfactual information-gain* reward that credits the action for the information it elicits. After the agent acts, the user or environment provides feedback. We then ask how this feedback would change the agent's next decision. Concretely, we score the same collected trajectory under two conditions: (i) the factual history that includes the feedback and (ii) a counterfactual history where the feedback is replaced by a masked placeholder. We compare the policy's probability assigned to the actual next action under the two conditions. The resulting distribution gap is attributed to the action that triggered the feedback, rewarding turns that meaningfully reshapes downstream choices. This process doesn't need additional interactions with environments.

To keep this intrinsic signal aligned with task completion, InfoPO uses an adaptive *variance-gated fusion* to combine information gain with outcome-based updates. When outcome rewards within a rollout group are non-discriminative, group-relative advantages can be near zero and learning may stall. In this case, the gate increases the weight on information gain to maintain a usable training signal. As outcomes become more discriminative, the gate shifts weight back to the task objective to support eventually success.

We also provide an information theoretic interpretation of the learning signal. In expectation, the per turn information gain reward corresponds to a conditional mutual information between feedback and the agent's next action, while its cumulative form represents the directed information flow from observations to decisions. Crucially, we prove that a minimum cumulative information gain is a strictly necessary resource for achieving task success, providing a theoretical lower bound that links uncertainty reduction directly to the probability of reaching the goal.

We evaluate InfoPO on three representative interactive benchmarks, UserGym (Qian et al., 2025c), ColBench (Zhou et al., 2025), and the long-horizon τ^2-Bench (Barres et al., 2025), which together span intent clarification and preference elicitation, collaborative code generation, and tool-augmented decision making. InfoPO consistently improves task performance and learning stability over strong prompting and RL baselines. To attribute these gains, we run component ablations and diagnostic analyses, uncovering a learned interaction pattern that resolves intent uncertainty early before downstream commitments. We further test robustness beyond user-centric benchmarks by applying InfoPO to non-user-interactive tasks and by swapping the simulated user at evaluation time, indicating that InfoPO learns broadly useful interaction strategies.

In summary, our contributions are threefold: **(1)** we introduce InfoPO, an information-driven RL method for multi-turn interaction that provides dense turn-level credit to enable effective learning under sparse or delayed outcomes; **(2)** we provide an information-theoretic grounding that links the proposed signal to conditional mutual information and directed information, clarifying the role of information flow from feedback to actions in learning; **(3)** we deliver a comprehensive empirical study

with ablations and diagnostic analyzes across multiple interactive benchmarks, showing improved task performance and training stability, and generalization to variant users and environments.

2 Related Works

User-centric agents. User-centric agents move beyond "task completion" toward inferring latent intent, preferences, and user state under multi-turn interaction. On the intent/need side, recent work studies implicit intention elicitation and clarification policies for agents (Qian et al., 2024; Chen et al., 2024), and connects user-facing transparency/explanations to predictability and controllability in personalized settings (Zhao et al., 2025a; Hong & Roth, 2026). On the personalization side, the community is rapidly enriching benchmarks and problem formulations, from classic persona-grounded dialogue (Zhang et al., 2018) to modern LLM-centric personalization suites such as LaMP/LaMP-QA (Salemi et al., 2024; Salemi & Zamani, 2025), and preference-heterogeneity benchmarks for individualized alignment (Zollo et al., 2024; Afzoon et al., 2024). In realistic user assistance, goals are embedded in open-ended workflows that require iterative feedback incorporation, grounded actions, and goal adaptation under practical constraints (e.g., interacting with external information, code, services, or APIs), motivating multi-turn testbeds that jointly evaluate these capabilities (Barres et al., 2025; Qian et al., 2025b;c; Wu et al., 2025; He et al., 2025). A complementary line builds agent training/evaluation scaffolds via user simulators, long-term memory, and synthetic environment generation (Sun et al., 2025; Li et al., 2025b; Cai et al., 2025). These advances highlight both the promise of group-relative RL for agentic behavior and the open challenge of stable, fine-grained credit assignment in long, interactive rollouts.

Agentic reinforcement learning. Reinforcement learning has become a broadly applicable approach to improve LLM agents' decision-making across diverse tasks. Multi-turn credit assignment is studied via hierarchical and collaborative training (Zhou et al., 2024; 2025), and grounded tool use and information seeking are induced with RL across search, clarification, and tool actions (Jin et al., 2025; Zhao et al., 2025b; Acikgoz et al., 2025); analyses of long-rollout dynamics further identify instability and motivate stabilization (Wang et al., 2025). Optimization has shifted toward RLVR-style designs for long, high-variance trajectories: group-relative methods and refinements (Shao et al., 2024; Feng et al., 2025), sequence/length-normalized stabilization, (Zheng et al., 2025; Zhao et al., 2025c), training efficiency (Yu et al., 2025; Sheng et al., 2025), and variants relaxing strict group synchronization (Xu & Ding, 2025). For tool interaction, ARPO/AEPO handle post-tool uncertainty with entropy-aware rollout and update stabilization (Dong et al., 2025a;b); for multi-reward settings, GDPO decouples normalization (Liu et al., 2026) and ARIA aggregates rewards by intention (Yang et al., 2025b). InfoPO proposes turn-level information advantage for credit assignment in user-centric multi-turn interaction, avoiding task-specific dense shaping or fully trained process reward models.

Reward Shaping in RL. Reward shaping is a common way to speed up learning when rewards are sparse. A classic approach adds intrinsic signals to promote information seeking, such as curiosity bonuses based on novelty or prediction error (Pathak et al., 2017; Burda et al., 2018), or empowerment-style objectives that increase future controllability (Klyubin et al., 2005; Mohamed & Jimenez Rezende, 2015). In LLM post-training, similar ideas appear as preference-based supervision, which is denser than end-task success alone; process reward models (PRMs) push this further to the step level by scoring intermediate reasoning steps (Ma et al., 2023; Khalifa et al., 2025; Xi et al., 2025). Stepwise feedback has shown clear gains on complex reasoning (e.g., math), improving verification and self-correction by reducing compounding errors (Setlur et al., 2024; Ye et al., 2025). InfoPO derives a turn-level learning signal without requiring task-specific heuristics.

3 Preliminaries

3.1 Multi-Turn Interaction as a Dec-POMDP

We model the user-centric multi-turn task as a decentralized partially-observable Markov decision process (Dec-POMDP) (Bernstein et al., 2002). In this setting, the agent repeatedly interacts with a user (or simulator). At each turn t, the environment transitions to a latent state $s_{t+1} \sim \mathcal{P}(\cdot \mid s_t, a_t)$ and reveals an observation $o_t \sim \mathcal{O}(\cdot \mid s_{t+1})$. Let $h_t = (q, a_1, o_1, \ldots, a_{t-1}, o_{t-1})$ denote the

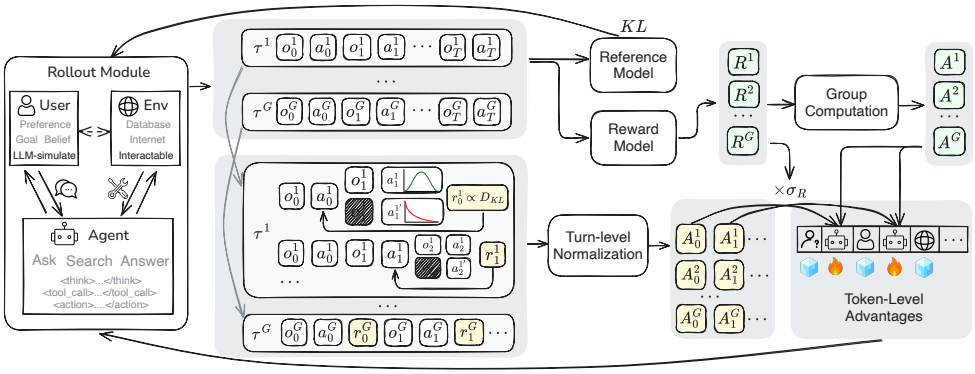


Figure 2: Overview of the InfoPO framework. It extracts a turn-level info-gain signal by counterfactual reasoning and adaptively fuses it with outcome-based advantages to facilitate efficient credit assignment in multi-turn user-centric tasks.

interaction history before observing o_t. The policy $\pi_\theta(a_t \mid h_t, o_t)$ generates a sequence of tokens conditioned on the transcript. In user-centric tasks, the observation o_t reduces uncertainty about the task goal and shifts the action distribution for the next turn a_{t+1}.

3.2 Optimization with Group-Relative Policy Gradient

We maximize the expected extrinsic return $J(\theta) = \mathbb{E}_{\tau\sim\pi_\theta}[R^{\text{ext}}(\tau)]$, where a trajectory τ corresponds to one environment episode induced by a prompt x_i and the model-generated response $y_i = (y_{i,1}, \ldots, y_{i,L_i})$. For LLM agents, we optimize at the token level with an advantage signal $A_{i,k}$:

$$\nabla_\theta \mathcal{L}(\theta) \propto \mathbb{E}\left[\sum_{k=1}^{L_i} A_{i,k} \nabla_\theta \log \pi_\theta(y_{i,k} \mid x_i, y_{i,<k})\right]. \tag{1}$$

InfoPO is built upon GRPO Shao et al. (2024), which computes $A_{i,k}$ by comparing trajectories within a rollout group to estimate variance-reduced advantages without an explicit critic.

4 Methods

We propose **InfoPO**, a multi-turn RL algorithm for user-centric agents where intent and constraints are revealed through feedback. As shown in Figure 2, InfoPO introduces a **turn-level counterfactual info-gain reward** that credits actions by how much the received observation changes the policy's next-step decision under a masked-observation counterfactual. InfoPO further applies a **variance-gated fusion** strategy that adaptively combines info-gain and outcome-based learning based on the within-group discriminativeness of external rewards. Algorithm 1 summarizes the procedure.

Section 4.1 defines the counterfactual info-gain reward. Section 4.2 presents advantage estimation and the variance-gated fusion that yields the final objective. Section 4.3 provides an information-theoretic interpretation and establishes the necessity of information gain for task success.

4.1 Turn-level Counterfactual Information Gain

A key challenge is to define a task-agnostic measure of information progress. We posit that a high-quality observation should *reduce the uncertainty* of the agent's task state, which is reflected in the shift of its subsequent action distribution. By comparing the real transcript to a counterfactual one where the observation is absent, we can isolate the specific "information gain" attributed to that turn.

At turn t, the agent produces an action segment a_t and receives feedback o_t. Let $a_{t+1} = (y_1, \ldots, y_{L_{t+1}})$ be the realized next action token sequence. We define the **turn-level info-gain**

reward r_t^{info} as the average log-probability shift over the next action tokens:

$$r_t^{\text{info}} = \frac{1}{L_{t+1}} \sum_{k=1}^{L_{t+1}} \Big(\log \pi_\theta(y_k \mid h_t, o_t, y_{<k}) - \log \pi_\theta(y_k \mid h_t, \varnothing, y_{<k}) \Big), \tag{2}$$

where $\varnothing$ is a string placeholder named "No information found." Statistical analysis in Appendix B.3 confirms that our method is robust to different placeholder implementations. Both terms in Eq. 2 are computed using *teacher forcing* on the same realized tokens a_{t+1}. This design serves two purposes: (1) *Causal Isolation*: it ensures that the shift in r_t^{info} is strictly caused by the presence of o_t rather than stochastic variations in autoregressive generation; (2) *Computational Tractability*: by using the same tokens, we avoid the prohibitive cost of multiple autoregressive rollouts for each counterfactual turn, allowing r_t^{info} to be computed efficiently via parallelized forward passes.

4.2 Unified Group-Relative Advantage Construction

For each prompt, we sample a group of G trajectories $\tau_1, \ldots, \tau_G$ and form advantages by comparing trajectories within the same context. We then compute the outcome-based advantage and the info-gain advantage separately:

Outcome Advantage. Let R_i^{ext} be the trajectory-level external reward for rollout i. We compute the normalized outcome advantage $A_{i,k}^{\text{ext}}$ for each token k in the response: $A_{i,k}^{\text{ext}} = \frac{R_i^{\text{ext}} - \mu_g^{\text{ext}}}{\sigma_g^{\text{ext}} + \epsilon} \cdot m_{i,k}$, where μ_g^{ext} and σ_g^{ext} are the mean and standard deviation of external scores within group g, $m_{i,k}$ is a response mask, and ϵ is a small constant to avoid division by zero.

Info-Gain Advantage. We normalize turn-level info-gain rewards within group g and broadcast the resulting scalar to the tokens of the corresponding action segment: $A_{i,k}^{\text{info}} = \frac{r_{i,t(k)}^{\text{info}} - \mu_g^{\text{info}}}{\sigma_g^{\text{info}} + \epsilon} \cdot m_{i,k}$, where $t(k)$ maps token k to its associated interaction turn, and $(\mu_g^{\text{info}}, \sigma_g^{\text{info}})$ are computed over all valid turns in group.

Adaptive Fusion via Variance Gating. We then combine the two advantages into a single update signal. InfoPO uses an adaptive gate $g(\cdot)$ to scale the info-gain contribution according to the within-group variability of the outcome signal: $g(\sigma_g^{\text{ext}}) = \sigma\left(-\frac{\sigma_g^{\text{ext}}}{T}\right)$, where T is a temperature parameter. When the external outcomes within a group are nearly indistinguishable ($\sigma_g^{\text{ext}} \approx 0$), $g(\cdot)$ increases, allowing the info-gain advantage to drive learning. Conversely, as the outcome signal becomes more discriminative, $g(\cdot)$ approaches 0, and the policy prioritizes task success. The final unified advantage for token k in rollout i is:

$$\hat{A}_{i,k} = A_{i,k}^{\text{ext}} + \beta \cdot g(\sigma_g^{\text{ext}}) \cdot A_{i,k}^{\text{info}}, \tag{3}$$

where β controls the peak influence of the information progress. This design encourages informative interaction while keeping optimization anchored to the task objective.

Finally, we optimize the policy π_θ relative to the reference policy π_{ref} to control distribution shift. The unified objective of InfoPO is formulated as:

$$\begin{aligned} \mathcal{J}_{\text{InfoPO}}(\theta) = \mathbb{E}_{q \sim P(Q),\, \{\tau_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}} \Bigg[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|\tau_i|} \sum_{k=1}^{|\tau_i|} \Bigg\{ \min\Bigg(\frac{\pi_\theta(y_{i,k} \,|\, x_{i,k})}{\pi_{\theta_{\text{old}}}(y_{i,k} \,|\, x_{i,k})} \hat{A}_{i,k}, \\ \text{clip}\Big(\frac{\pi_\theta(y_{i,k} \,|\, x_{i,k})}{\pi_{\theta_{\text{old}}}(y_{i,k} \,|\, x_{i,k})},\, 1-\epsilon,\, 1+\epsilon \Big) \hat{A}_{i,k} \Bigg) - \lambda_{\text{KL}} D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \Bigg\} \Bigg], \end{aligned} \tag{4}$$

where $x_{i,k}$ denotes the context $(x_i, y_{i,<k})$ for the k-th token.

4.3 Theory: Info-Gain as a Necessary Resource

We summarize two key results proving that our turn-level info-gain reward is a rigorous measure of information progress. Full proofs are provided in Appendix B.

Table 1: Experimental results on UserGym, ColBench, and τ^2-Bench. The grey rows represent our proposed method and its ablation versions. **Best** and second-best results are marked *within each model group* (Closed-Source / Qwen2.5 / Qwen3).

Type	Method	UserGym								ColBench		τ^2-Bench		
		Travel	Func.	Pers.	Tau	Turtle	*Search*	*Inten.*	*Tele.*	Pass	Succ.	Telec.	Retail	Air.
Closed-Source Model														
Prompting	Gemini-3-Flash	0.574	**0.423**	**0.695**	**0.167**	0.153	**0.968**	1.718	**0.829**	0.515	0.382	**0.469**	**0.619**	**0.558**
Prompting	GPT-4.1	0.554	0.051	0.599	0.109	**0.267**	0.480	**1.867**	0.732	**0.529**	**0.403**	0.388	0.544	0.400
Prompting	GPT-4o-mini	**0.596**	0.089	0.683	0.091	0.117	0.568	1.277	0.512	0.463	0.342	0.113	0.400	0.175
Qwen2.5-7B-Instruct														
Prompting	Qwen2.5	0.441	0.026	0.289	0.000	0.062	0.376	1.254	0.292	0.242	0.140	0.144	0.131	0.075
Prompting	ReAct	0.452	0.064	0.325	0.037	0.078	0.354	1.378	0.316	0.238	0.135	0.131	0.138	0.100
Prompting	Reflexion	0.445	0.052	0.312	0.022	0.074	0.364	1.320	0.305	0.276	0.168	0.138	0.131	0.075
RL Training	RAGEN	0.538	0.124	**0.548**	0.000	0.148	0.446	1.815	0.416	0.449	0.348	0.175	0.175	0.150
RL Training	Search-R1	0.565	0.113	0.412	0.043	0.154	0.435	1.805	0.416	0.457	0.352	0.156	0.150	0.100
RL Training	UserRL	0.546	0.115	0.444	0.048	0.152	0.429	1.826	0.424	0.436	0.327	0.138	0.169	0.075
RL Training	InfoPO w/o std	0.565	0.142	0.498	0.035	0.065	0.455	1.845	0.452	0.395	0.298	0.169	0.162	0.125
RL Training	InfoPO w/o Gate	0.542	0.125	0.465	0.082	0.042	0.432	1.810	0.465	0.466	0.368	0.150	0.156	0.100
RL Training	InfoPO w/o R_{ext}	0.485	0.055	0.352	0.035	0.015	0.385	1.450	0.325	0.285	0.342	0.112	0.125	0.088
RL Training	**InfoPO (Ours)**	**0.588**	**0.167**	0.535	**0.091**	**0.178**	**0.480**	**1.892**	**0.488**	**0.534**	**0.426**	**0.181**	**0.188**	**0.163**
Qwen3-4B														
Prompting	Qwen3	0.277	0.026	0.452	0.006	0.071	0.444	1.660	0.488	0.272	0.153	0.106	0.225	0.062
Prompting	ReAct	0.279	0.053	0.476	0.005	0.147	0.435	1.782	0.454	0.269	0.145	0.094	0.225	0.088
Prompting	Reflexion	0.291	0.045	0.474	0.035	0.124	0.438	1.717	0.501	0.303	0.184	0.100	0.200	0.075
RL Training	RAGEN	0.477	0.104	0.511	0.006	0.117	0.714	1.572	0.488	0.479	0.361	0.137	0.231	0.088
RL Training	Search-R1	0.482	0.103	0.425	0.059	0.113	0.702	1.711	0.412	0.467	0.355	0.118	0.234	0.088
RL Training	UserRL	0.538	0.095	0.507	0.053	0.121	0.697	1.732	0.520	0.468	0.342	0.100	0.163	0.075
RL Training	InfoPO w/o std	0.512	0.102	0.518	0.075	0.118	0.795	1.812	0.485	0.498	0.395	0.142	0.215	0.088
RL Training	InfoPO w/o Gate	0.548	0.085	0.482	0.062	0.115	0.758	1.765	0.460	0.475	0.372	0.131	0.198	0.075
RL Training	InfoPO w/o R_{ext}	0.352	0.045	0.385	0.032	0.088	0.425	1.512	0.382	0.345	0.285	0.095	0.152	0.055
RL Training	**InfoPO (Ours)**	**0.589**	**0.115**	**0.556**	**0.097**	**0.154**	**0.849**	**1.862**	**0.542**	**0.553**	**0.439**	**0.156**	**0.244**	**0.100**

Theorem 1 (Equivalence to Mutual Information)**.** *Let H_t be the history, O_t the feedback, and A_{t+1} the subsequent action. Defining the marginal policy $\pi_\theta(\cdot \mid H_t) \triangleq \mathbb{E}_{O_t \sim P(\cdot|H_t)}\big[\pi_\theta(\cdot \mid H_t, O_t)\big]$, the turn-level info-gain reward r_t^{info} defined in Eq. 2 satisfies: $\mathbb{E}[r_t^{\text{info}}] = I_\theta(O_t; A_{t+1} \mid H_t)$.*

Theorem 1 equates the info-gain reward to the conditional mutual information between feedback and actions, formalizing the directed information flow that drives decisions.

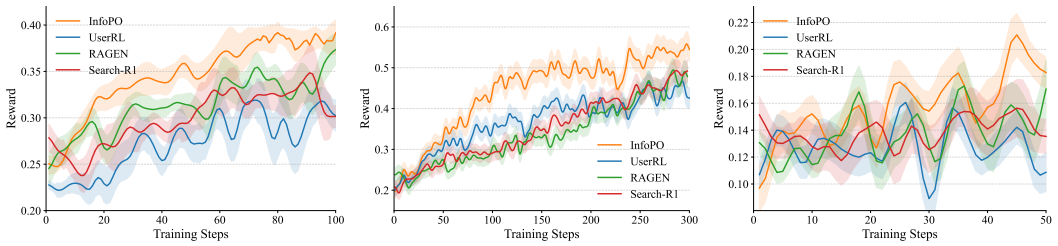
Theorem 2 (Necessity for Task Success)**.** *Consider a task with a hidden intent $Z \sim \mathrm{Unif}([M])$ and a terminal reward $R^{\text{ext}} = \mathbb{I}[\hat{Z} = Z]$. Achieving a success probability $\mathbb{P}(R^{\text{ext}} = 1) \geq 1 - \delta$ **requires** the cumulative info-gain reward to satisfy the lower bound: $\mathbb{E}\left[\sum_{t=0}^{T-1} r_t^{\text{info}}\right] \geq \log M - h(\delta) - \delta \log(M-1)$, where $h(\delta)$ is the binary entropy (in nats).*

5 Experiments

Our experiments aim to answer the following research questions: **RQ1 (Performance):** How effectively does InfoPO facilitate task completion and learning efficiency across benchmarks? **RQ2 (Mechanism):** How do turn-level info-gain rewards and variance-gated fusion contribute to interaction quality and learning stability? **RQ3 (Generalization):** How robustly does InfoPO generalize to unseen interaction purposes and varying user simulator conditions?

5.1 Experimental Setup

Benchmarks and Metrics. We evaluate InfoPO on three multi-turn, user-centric benchmarks that test an agent's dual interaction: eliciting latent intent through dialogue while taking environment-grounded actions. (1) **UserGym** (Qian et al., 2025c): eight unified gym environments spanning travel planning, preference persuasion, and goal inference, with a standardized `[Action-Search-Answer]` interface that couples tool use with information-seeking dialogue. Following Qian et al. (2025c), we report success rate or accumulated reward across all eight tasks, including three held-out settings for cross-purpose generalization. (2) **ColBench** (Zhou et al., 2025): a collaborative programming benchmark where the agent iteratively clarifies requirements and produces Python code; we report the fraction of hidden unit tests passed (Pass) and task completion rate (Succ.). (3) **τ^2-Bench** (Barres et al., 2025): complex dual-control tasks in airline, retail, and telecom domains,



(a) UserGym (b) ColBench (c) τ^2-Bench

Figure 3: Extrinsic reward curves during training on (a) UserGym, (b) ColBench, and (c) τ^2-Bench. Solid lines and shaded regions represent mean $\pm$ std across three seeds.

where both agent and user can modify the shared world state; we report average success over four runs (Avg@4). Detailed task descriptions are in Appendix A.

Baselines. We compare InfoPO against the following baselines: (i) **UserRL** Qian et al. (2025c), a representative user-centric multi-turn training framework. We adopt its strongest *Equalized/R2G* setting, which applies length-normalized rewards (*Equalized*) and future-reward-based advantage estimation (*R2G*) to the entire interaction sequence. (ii) **RAGEN** Wang et al. (2025), which focuses on training stability through variance-based trajectory filtering and decoupled clipping. (iii) **Search-R1** Jin et al. (2025), which optimizes search-based reasoning via retrieved-token masking. (iv) **ReAct** Yao et al. (2022) and **Reflexion** Shinn et al. (2023), serving as non-training prompting baselines to quantify the necessity of policy optimization.

Training details. We use Qwen2.5-7B-Instruct (Qwen et al., 2025) and Qwen-3-4B (Yang et al., 2025a) as base models. To probe whether agents can discover interaction strategies purely from reinforcement signals, we train with RL from scratch. We use group-based rollouts with $n{=}5$ samples per prompt. For multi-turn interaction, we cap turns at $16/10/50$ for UserGym, ColBench, and τ^2-Bench, and set batch sizes to $64/64/32$, respectively, with a maximum rollout length of $32{,}768$ tokens. Across all benchmarks, we use GPT-4o-mini (Hurst et al., 2024) as the default user simulator (temperature 0.7) and follow each benchmark's official train/test splits. Experiments run on $4\times$ NVIDIA A800 (80GB). Additional hyperparameters are in Appendix B.1; compute overhead of counterfactual evaluation is discussed in Appendix B.2 and is typically below $2\times$ (avg. $1.63\times$).

5.2 RQ1: Overall Performance

Table 1 summarizes the final performance across the three benchmarks. On the Qwen2.5-7B-Instruct backbone, InfoPO achieves the strongest overall results among open-source RL baselines. In UserGym, InfoPO improves upon the strongest baseline in 7 out of 8 sub-environments, with particularly significant gains in cross-purpose generalization settings that require resolving underspecified goals (e.g., Search: 0.480 vs. 0.446; Intention: 1.892 vs. 1.826; Telepathy: 0.488 vs. 0.424). For code-centric tasks in ColBench, InfoPO yields clear improvements on both technical metrics (Pass: 0.534 vs. 0.457; Success: 0.426 vs. 0.352), slightly exceeding the performance of GPT-4.1 (0.529/0.403).

Table 2: Percentage of rollout groups with zero outcome variance during the initial training phase.

Model	UserGym	ColBench	τ^2-Bench
Qwen2.5-7B-Inst.	0.384	0.313	0.763
Qwen3-4B	0.421	0.345	0.812

The superior efficiency of InfoPO is directly attributable to its improved credit assignment under non-informative feedback. As shown in Table 2, a significant portion of rollout groups exhibits zero outcome variance during the initial training phase (31.3%–38.4% in UserGym/ColBench, and up to 76.3% in τ^2-Bench). While standard group-relative advantages become brittle or provide near-zero gradients in these scenarios, InfoPO's info-gain reward serves as a dense training scaffold to bootstrap policy improvement. Consistent with this, the training dynamics in Figure 3 demonstrate that InfoPO initiates optimization earlier and reaches higher reward levels with reduced oscillations compared to baselines. Qualitative inspection of interaction traces (see Figure 9, 10 and 11 in Appendix D) reveals that InfoPO-trained agents

exhibit structured and proactive strategies, such as resolving intent ambiguity through early-turn clarification, reflecting a form of behavioral maturation.

On the long-horizon τ^2-Bench, InfoPO maintains its competitiveness by matching or improving upon the best open-source baselines across all task families (Telecom: 0.181; Retail: 0.188; Air: 0.150). Considering τ^2-Bench's extreme interaction horizon (often > 30 turns) and severe data scarcity (only 178 tasks), the steady improvement from a base instruct model validates InfoPO's effectiveness.

5.3 RQ2: InfoPO Mechanism Analysis

Ablations. To isolate the contributions of InfoPO's core designs, we evaluate three variants: InfoPO **w/o** R_{ext} (pure information gain), **w/o Gate** (fixed weighting without variance-based gating), and **w/o std** (removing info-gain normalization). Table 1 and Figure 4 summarize final performance (J_f) together with stability and interaction diagnostics (e.g., Δ_{bf}, P_{cr}, and length-related statistics; see Appendix A.4 for formal definitions). Removing extrinsic supervision (**w/o** R_{ext}) leads to a consistent and substantial drop across nearly all tasks, underscoring that information seeking alone is insufficient without task-level grounding. Disabling dynamic gating (**w/o Gate**) primarily hurts training stability, manifested as larger best-to-final regression Δ_{bf} and higher collapse probability P_{cr}, validating that the gate is important for preventing late-stage objective drift when the policy transitions from early uncertainty reduction to outcome refinement. Finally, removing standardization (**w/o std**) not only lowers J_f but also increases length sensitivity (higher $|\rho_{L,r}|$), indicating that group-relative normalization is critical for stabilizing turn-level credit assignment and preventing the information advantage from being dominated by a small number of high-magnitude but noisy turns.

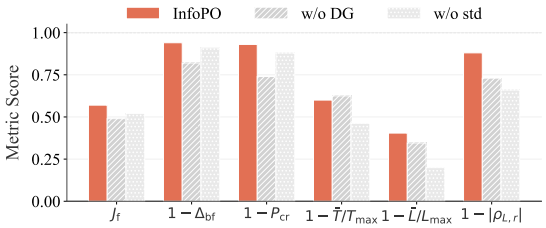
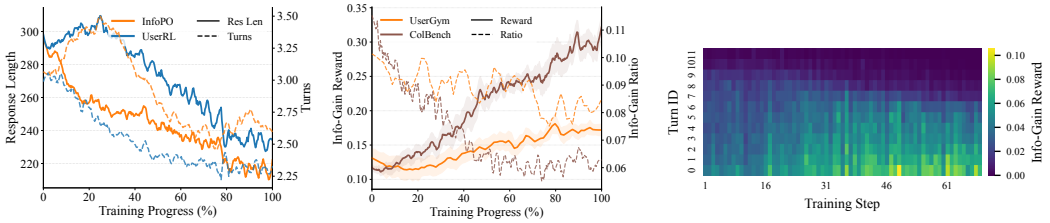


Figure 4: Mechanism diagnostics under ablations (aggregated across tasks). Metrics are normalized to higher-is-better; see Appendix A.

Interaction Dynamics. A concern about InfoPO is whether information rewards merely incentivize longer, more repetitive interactions. To investigate this, we track interaction turns and response lengths throughout training (Figure 5a). In UserGym and ColBench, InfoPO exhibits an emergent "explore-then-consolidate" pattern: it temporarily increases the number of turns in early training to reduce intent uncertainty, while steadily shortening each turn's response length. In contrast, baselines like UserRL monotonically shrink both turns and length, often collapsing to "short-horizon" behaviors that prematurely commit to actions without sufficient information. Figure 5b further reveals that while the absolute info-gain signal increases as the agent becomes more inquisitive, its relative contribution to the final advantage decreases via the variance gate. This confirms that InfoPO utilizes interaction as a strategic resource—expanding it only when necessary to acquire task-critical information and consolidating into efficient execution as outcome-based learning takes over.



(a) Turns vs. Response length (b) Info-gain reward dynamics (c) Info-gain rewards per turn.

Figure 5: Interaction dynamics and reward signals. (a) Turns vs. response length; (b) Absolute info-gain (solid) and its advantage contribution ratio (dashed); (c) Turn-level info-gain reward heatmap.

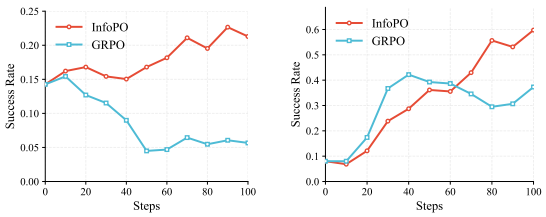
Interestingly, in the high-horizon τ^2-Bench domain, InfoPO shifts toward a "pruning-first" strategy, with both turns and response lengths decreasing from the onset of training (see Appendix B.4 for detailed curves). It directs the policy toward more concise and goal-oriented behaviors. This

context-aware behavior demonstrates that InfoPO adaptively modulates interaction depth based on the inherent difficulty and information density of the task environment.

Per-turn credit assignment. Figure 5c shows the distribution of the info-gain reward over turns during training on UserGym, averaged over valid turns at each step. Early on, rewards are spread across the dialogue; as training progresses, they concentrate on the first few turns, indicating an emergent "clarify-then-act" behavior. Specifically, successful policies learn to place the most discriminative questions at the beginning to elicit high-information feedback, after which intrinsic rewards naturally decay as the agent executes a specified task. This trend is not imposed by any heuristic: under our counterfactual objective, a turn is rewarded only when the observed feedback induces a measurable shift in the policy's subsequent decision distribution. Consequently, InfoPO learns where to ask rather than simply to ask more, yielding a self-organized progression that prioritizes early uncertainty reduction as a precursor to task success.

5.4 RQ3: Generalization

To evaluate whether InfoPO generalizes beyond user-centric environments, we conduct experiments on **Sokoban** and **WebShop** following the protocols in RAGEN. Sokoban is a multi-turn task requiring planning in a grid-world environment, while WebShop requires grounding in a realistic web interface. Using Qwen2.5-1.5B-Instruct as the base model, we observe that InfoPO successfully mitigates the "Echo Trap" failure mode—a common collapse in standard GRPO where policies regress to repetitive, locally-rewarded templates once rollout groups fail to reach the goal (Wang et al., 2025). As shown in Figure 6, InfoPO maintains a stable upward trend in success rate where baselines collapse. This demonstrates that InfoPO's effectiveness beyond specific user-centric scenarios, framing general multi-turn interaction as a fundamental process of active uncertainty reduction.



(a) Sokoban (b) Webshop

Figure 6: Success rates on environment-interactive tasks.

User generalization. Since all training runs use GPT-4o-mini as the simulated user for cost efficiency, we further examine whether InfoPO's performance is sensitive to the user when evaluation. We evaluate InfoPO (Qwen-2.5-7B) under three user conditions: (**Base**) GPT-4o-mini with the official benchmark prompt; (**Optimized Prompt, OP**) the same model using constrained instructions to enforce protocol compliance and minimize unforced errors in consistency or tool-call execution. (See Appendix E for prompts); and (**Optimized Model, OM**) a stronger model, GPT-4.1, using the official prompt. We report the average scores across all metrics for each benchmark in Table 3.

Table 3: Comparison of different simulated users.

User	UserGym	ColBench	τ^2-Bench
Base	0.552	0.480	0.173
OP	0.563	0.483	0.203
OM	0.558	0.502	0.215

Overall, stronger user simulators yield consistent but task-dependent effects. OM brings the largest gains on τ^2-Bench, where coordination over tool calls is critical and user mistakes can cause failures unrelated to the agent policy. ColBench shows smaller yet generally positive improvements, suggesting moderate sensitivity to user reliability. In contrast, UserGym shows mixed effects, consistent with its design that enforces realistic user behaviors (e.g., progressive disclosure, resistance to persuasion, and strict non-hallucination), which stronger simulators follow more faithfully and can therefore make tasks harder.

6 Conclusion

We introduced **InfoPO**, an information-driven policy optimization method for user-centric multi-turn agents that treats interaction as active uncertainty reduction. InfoPO provides dense, turn-level credit assignment by measuring how each observation counterfactually changes the policy's next-action

distribution, and keeps this intrinsic signal aligned with task success via a variance-gated fusion with outcome advantages. This yields a task-agnostic yet scalable learning signal for long-horizon interaction where group-relative updates often stall under sparse or non-discriminative rewards. Across three interactive benchmarks, InfoPO consistently improves performance, sample efficiency, and training stability over strong prompting and RL baselines, and further generalizes to user-simulator shifts and non-user environment interaction tasks. Overall, InfoPO offers a principled mechanism to optimize reliable agent behavior under partial information without task-specific shaping or supervised cold starts.

REFERENCES

Emre Can Acikgoz, Jinoh Oh, Jie Hao, Joo Hyuk Jeon, Heng Ji, Dilek Hakkani-Tür, Gokhan Tur, Xiang Li, Chengyuan Ma, and Xing Fan. Speakrl: Synergizing reasoning, speaking, and acting in language models with reinforcement learning. *arXiv preprint arXiv:2512.13159*, 2025.

Saleh Afzoon, Usman Naseem, Amin Beheshti, and Zahra Jamali. Persobench: Benchmarking personalized response generation in large language models. *arXiv preprint arXiv:2410.03198*, 2024.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. Tau2-bench: Evaluating conversational agents in a dual-control environment. *arXiv preprint arXiv:2506.07982*, 2025.

Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of markov decision processes. *Mathematics of operations research*, 27(4): 819–840, 2002.

Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

Shihao Cai, Runnan Fang, Jialong Wu, Baixuan Li, Xinyu Wang, Yong Jiang, Liangcai Su, Liwen Zhang, Wenbiao Yin, Zhen Zhang, et al. Autoforge: Automated environment synthesis for agentic reinforcement learning. *arXiv preprint arXiv:2512.22857*, 2025.

Maximillian Chen, Ruoxi Sun, Tomas Pfister, and Sercan Ö Arık. Learning to clarify: Multi-turn conversations with action-based contrastive self-training. *arXiv preprint arXiv:2406.00222*, 2024.

Mingyi Deng, Lijun Huang, Yani Fan, Jiayi Zhang, Fashen Ren, Jinyi Bai, Fuzhen Yang, Dayi Miao, Zhaoyang Yu, Yifan Wu, et al. Interactcomp: Evaluating search agents with ambiguous queries. *arXiv preprint arXiv:2510.24668*, 2025.

Guanting Dong, Licheng Bao, Zhongyuan Wang, Kangzhi Zhao, Xiaoxi Li, Jiajie Jin, Jinghan Yang, Hangyu Mao, Fuzheng Zhang, Kun Gai, et al. Agentic entropy-balanced policy optimization. *arXiv preprint arXiv:2510.14545*, 2025a.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, et al. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*, 2025b.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for llm agent training. *arXiv preprint arXiv:2505.10978*, 2025.

Muyu He, Anand Kumar, Tsach Mackey, Meghana Rajeev, James Zou, and Nazneen Rajani. Impatient users confuse ai agents: High-fidelity simulations of human traits for testing agents. *arXiv preprint arXiv:2510.04491*, 2025.

Pingjun Hong and Benjamin Roth. Do llm self-explanations help users predict model behavior? evaluating counterfactual simulatability with pragmatic perturbations. *arXiv preprint arXiv:2601.03775*, 2026.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516*, 2025.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. Process reward models that think. *arXiv preprint arXiv:2504.16828*, 2025.

Young-Han Kim. A coding theorem for a class of stationary channels with feedback. *IEEE Transactions on Information Theory*, 54(4):1488–1499, 2008.

Alexander S Klyubin, Daniel Polani, and Chrystopher L Nehaniv. Empowerment: A universal agent-centric measure of control. In *2005 ieee congress on evolutionary computation*, volume 1, pp. 128–135. IEEE, 2005.

Yubo Li, Xiaobin Shen, Xinyu Yao, Xueying Ding, Yidi Miao, Ramayya Krishnan, and Rema Padman. Beyond single-turn: A survey on multi-turn interactions with large language models. *arXiv preprint arXiv:2504.04717*, 2025a.

Yuetai Li, Huseyin A Inan, Xiang Yue, Wei-Ning Chen, Lukas Wutschitz, Janardhan Kulkarni, Radha Poovendran, Robert Sim, and Saravan Rajmohan. Simulating environments with reasoning models for agent training. *arXiv preprint arXiv:2511.01824*, 2025b.

Bang Liu, Xinfeng Li, Jiayi Zhang, Jinlin Wang, Tanjin He, Sirui Hong, Hongzhang Liu, Shaokun Zhang, Kaitao Song, Kunlun Zhu, et al. Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems. *arXiv preprint arXiv:2504.01990*, 2025.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, et al. Gdpo: Group reward-decoupled normalization policy optimization for multi-reward rl optimization. *arXiv preprint arXiv:2601.05242*, 2026.

Qianli Ma, Haotian Zhou, Tingkai Liu, Jianbo Yuan, Pengfei Liu, Yang You, and Hongxia Yang. Let's reward step by step: Step-level reward model as the navigators for reasoning. *arXiv preprint arXiv:2310.10080*, 2023.

James Massey et al. Causality, feedback and directed information. In *Proc. Int. Symp. Inf. Theory Applic.(ISITA-90)*, volume 2, pp. 1, 1990.

Shakir Mohamed and Danilo Jimenez Rezende. Variational information maximisation for intrinsically motivated reinforcement learning. *Advances in neural information processing systems*, 28, 2015.

Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.

Cheng Qian, Bingxiang He, Zhong Zhuang, Jia Deng, Yujia Qin, Xin Cong, Zhong Zhang, Jie Zhou, Yankai Lin, Zhiyuan Liu, et al. Tell me more! towards implicit user intention understanding of language model driven agents. *arXiv preprint arXiv:2402.09205*, 2024.

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, and Heng Ji. Toolrl: Reward is all tool learning needs. *arXiv preprint arXiv:2504.13958*, 2025a.

Cheng Qian, Zuxin Liu, Akshara Prabhakar, Zhiwei Liu, Jianguo Zhang, Haolin Chen, Heng Ji, Weiran Yao, Shelby Heinecke, Silvio Savarese, et al. Userbench: An interactive gym environment for user-centric agents. *arXiv preprint arXiv:2507.22034*, 2025b.

Cheng Qian, Zuxin Liu, Akshara Prabhakar, Jielin Qiu, Zhiwei Liu, Haolin Chen, Shirley Kokane, Heng Ji, Weiran Yao, Shelby Heinecke, et al. Userrl: Training interactive user-centric agent via reinforcement learning. *arXiv preprint arXiv:2509.19736*, 2025c.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL https://arxiv.org/abs/2412.15115.

Alireza Salemi and Hamed Zamani. Lamp-qa: A benchmark for personalized long-form question answering. *arXiv preprint arXiv:2506.00137*, 2025.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. Lamp: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7370–7392, 2024.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv preprint arXiv:2410.08146*, 2024.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Yuepeng Sheng, Yuwei Huang, Shuman Liu, Haibo Zhang, and Anxiang Zeng. Espo: Entropy importance sampling policy optimization. *arXiv preprint arXiv:2512.00499*, 2025.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

Weiwei Sun, Xuhui Zhou, Weihua Du, Xingyao Wang, Sean Welleck, Graham Neubig, Maarten Sap, and Yiming Yang. Training proactive and personalized llm agents. *arXiv preprint arXiv:2511.02208*, 2025.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Zihan Wang, Kangrui Wang, Qineng Wang, Pingyue Zhang, Linjie Li, Zhengyuan Yang, Xing Jin, Kefan Yu, Minh Nhat Nguyen, Licheng Liu, et al. Ragen: Understanding self-evolution in llm agents via multi-turn reinforcement learning. *arXiv preprint arXiv:2504.20073*, 2025.

Quan Wei, Siliang Zeng, Chenliang Li, William Brown, Oana Frunza, Wei Deng, Anderson Schneider, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, and Mingyi Hong. Reinforcing multi-turn reasoning in llm agents via turn-level reward design, 2025. URL https://arxiv.org/abs/2505.11821.

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

Zhiheng Xi, Chenyang Liao, Guanyu Li, Yajie Yang, Wenxiang Chen, Zhihao Zhang, Binghai Wang, Senjie Jin, Yuhao Zhou, Jian Guan, et al. Agentprm: Process reward models for llm agents via step-wise promise and progress. *arXiv preprint arXiv:2511.08325*, 2025.

Zhongwen Xu and Zihan Ding. Single-stream policy optimization. *arXiv preprint arXiv:2509.13232*, 2025.

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

Ruihan Yang, Yikai Zhang, Aili Chen, Xintao Wang, Siyu Yuan, Jiangjie Chen, Deqing Yang, and Yanghua Xiao. Aria: Training language agents with intention-driven reward aggregation. *arXiv preprint arXiv:2506.00539*, 2025b.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.

Zihuiwen Ye, Luckeciano Carvalho Melo, Younesse Kaddar, Phil Blunsom, Sam Staton, and Yarin Gal. Uncertainty-aware step-wise verification with generative reward models. *arXiv preprint arXiv:2502.11250*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Guibin Zhang, Hejia Geng, Xiaohang Yu, Zhenfei Yin, Zaibin Zhang, Zelin Tan, Heng Zhou, Zhongzhi Li, Xiangyuan Xue, Yijiang Li, et al. The landscape of agentic reinforcement learning for llms: A survey. *arXiv preprint arXiv:2509.02547*, 2025.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a dog, do you have pets too? *arXiv preprint arXiv:1801.07243*, 2018.

Siyan Zhao, Mingyi Hong, Yang Liu, Devamanyu Hazarika, and Kaixiang Lin. Do llms recognize your preferences? evaluating personalized preference following in llms. *arXiv preprint arXiv:2502.09597*, 2025a.

Weikang Zhao, Xili Wang, Chengdi Ma, Lingbin Kong, Zhaohua Yang, Mingxiang Tuo, Xiaowei Shi, Yitao Zhai, and Xunliang Cai. Mua-rl: Multi-turn user-interacting agent reinforcement learning for agentic tool use. *arXiv preprint arXiv:2508.18669*, 2025b.

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, et al. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*, 2025c.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. Archer: Training language model agents via hierarchical multi-turn rl. *arXiv preprint arXiv:2402.19446*, 2024.

Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. Sweet-rl: Training multi-turn llm agents on collaborative reasoning tasks. *arXiv preprint arXiv:2503.15478*, 2025.

Thomas P Zollo, Andrew Wei Tung Siah, Naimeng Ye, Ang Li, and Hongseok Namkoong. Personalllm: Tailoring llms to individual preferences. *arXiv preprint arXiv:2409.20296*, 2024.

A INFOPO PSEUDO-CODE

B PROOFS OF THEORETICAL RESULTS

B.1 PROOF OF THEOREM 1

We consider a turn-based interaction of horizon T. Let H_0 denote the initial context (prompt, system message, etc.). For each turn $t \in \{0, \dots, T-1\}$, the environment (or user simulator) produces feedback O_t, and the agent then produces the next action segment A_{t+1}. Define the (pre-feedback) history random variable

$$H_t \triangleq (H_0, A_{1:t}, O_{0:t-1}),$$

Algorithm 1 INFOPO

Require: policy π_θ, reference π_{ref}, env $\mathcal{E}$; group size G, horizon T; mask placeholder $\varnothing$, weight β, gate temperature τ, PPO clip ϵ

for each iteration **do**
  Sample G trajectories $\{\tau_i\}_{i=1}^{G}$ of T turns from π_θ in $\mathcal{E}$, record turn boundaries.
  $R_i^{\text{ext}} \leftarrow \text{ScoreExt}(\tau_i)$
  $\{r_{i,t}^{\text{info}}\}_{t=1}^{T} \leftarrow \text{InfoGainPerTurn}(\tau_i, \pi_\theta, \pi_{\text{ref}}, \varnothing)$
  $A^{\text{ext}} \leftarrow \text{GroupNormExt}(\{R_i^{\text{ext}}\}_{i=1}^{G})$
  $A^{\text{info}} \leftarrow \text{GroupNormInfo}(\{r_{i,t}^{\text{info}}\}_{i,t})$
  $g \leftarrow \sigma(-\text{Std}_g(\{R_i^{\text{ext}}\})/\tau)$
  $\hat{A} \leftarrow A^{\text{ext}} + \beta \cdot g \cdot A^{\text{info}}$, broadcast to response tokens
  Update θ using $\hat{A}$ and KL-to-π_{ref} regularization.
end for
function $\text{InfoGainPerTurn}(\tau, \pi_\theta, \pi_{\text{ref}}, \varnothing)$
  Initialize $r_t^{\text{info}} \leftarrow 0$ for $t = 1..T$
  for $t = 1$ to $T-1$ **do**
    Let c_t be context up to (action$_t$, obs$_t$).
    $\ell^{\text{post}} \leftarrow \log \pi_\theta(a_{t+1} \mid h_t, o_t)$,
    $\ell^{\text{prior}} \leftarrow \log \tilde{\pi}_\theta(a_{t+1} \mid h_t)$,
    $r_t^{\text{info}} \leftarrow \ell^{\text{post}} - \ell^{\text{prior}}$. {attributed to turn t}
  end for
  return $\{r_t^{\text{info}}\}_{t=1}^{T}$
end function

where $A_{1:t} = (A_1, \ldots, A_t)$ and $O_{0:t-1} = (O_0, \ldots, O_{t-1})$. The policy is *causal*:

$$P_\theta(A_{t+1} \mid H_t, O_t) = \pi_\theta(A_{t+1} \mid H_t, O_t).$$

We further define the *marginal (prior) policy* by averaging over the next feedback:

$$\pi_\theta(a \mid h_t) \triangleq \mathbb{E}_{O_t \sim P(\cdot \mid h_t)}\big[\pi_\theta(a \mid h_t, O_t)\big],$$

where the expectation is w.r.t. the environment's conditional distribution of O_t given $H_t = h_t$. (For continuous variables, replace sums by integrals; the derivation remains identical.) InfoPO intrinsic reward is defined as

$$r_{\text{info}}(t) \triangleq \log \pi_\theta(A_{t+1} \mid H_t, O_t) - \log \pi_\theta(A_{t+1} \mid H_t).$$

Part I: Per-turn expectation equals conditional mutual information. Taking expectation over the joint distribution of (H_t, O_t, A_{t+1}) induced by the environment and policy,

$$\begin{aligned} \mathbb{E}[r_{\text{info}}(t)] &= \mathbb{E}_{H_t, O_t, A_{t+1}}\left[\log \frac{\pi_\theta(A_{t+1} \mid H_t, O_t)}{\pi_\theta(A_{t+1} \mid H_t)}\right] \\ &= \sum_{h_t} P(h_t) \sum_{o_t} P(o_t \mid h_t) \sum_{a} \pi_\theta(a \mid h_t, o_t) \log \frac{\pi_\theta(a \mid h_t, o_t)}{\pi_\theta(a \mid h_t)}. \end{aligned} \tag{5}$$

Note that the conditional joint distribution given $H_t = h_t$ factorizes as

$$P(o_t, a \mid h_t) = P(o_t \mid h_t)\pi_\theta(a \mid h_t, o_t).$$

Substituting into equation 5, we obtain

$$\mathbb{E}[r_{\text{info}}(t)] = \sum_{h_t} P(h_t) \sum_{o_t, a} P(o_t, a \mid h_t) \log \frac{P(o_t, a \mid h_t)}{P(o_t \mid h_t)\, P(a \mid h_t)}.$$

By the definition of conditional mutual information,

$$I_\theta(O_t; A_{t+1} \mid H_t) \triangleq \sum_{h_t} P(h_t) \sum_{o_t, a} P(o_t, a \mid h_t) \log \frac{P(o_t, a \mid h_t)}{P(o_t \mid h_t)\, P(a \mid h_t)},$$

we conclude

$$\mathbb{E}[r_{\text{info}}(t)] = I_\theta(O_t; A_{t+1} \mid H_t).$$

Part II: Cumulative InfoPO intrinsic equals directed information. Define $O^t \triangleq O_{0:t}$ and $A^t \triangleq A_{1:t}$. Following Massey (1990), the *directed information* from feedback to actions is defined as the sum of conditional mutual informations:

$$I_\theta(O^{T-1} \to A^T \mid H_0) \triangleq \sum_{t=0}^{T-1} I_\theta(O_t; A_{t+1} \mid H_0, A^t, O^{t-1}).$$

Recalling that the history is $H_t = (H_0, A^t, O^{t-1})$, conditioning on (H_0, A^t, O^{t-1}) is exactly equivalent to conditioning on H_t. Therefore, each summand simplifies directly:

$$I_\theta(O_t; A_{t+1} \mid H_0, A^t, O^{t-1}) = I_\theta(O_t; A_{t+1} \mid H_t).$$

Summing over t and applying the result from Part I:

$$I_\theta(O^{T-1} \to A^T \mid H_0) = \sum_{t=0}^{T-1} \mathbb{E}[r_{\text{info}}(t)] = \mathbb{E}\left[\sum_{t=0}^{T-1} r_{\text{info}}(t)\right].$$

This completes the proof. ■

B.2 Proof of Theorem 2

Problem model. Let $Z \sim \text{Unif}(\{1, \dots, M\})$ be a hidden user intent (or goal) that the agent must infer through interaction. The agent interacts for T turns and outputs actions $A^T = (A_1, \dots, A_T)$. A terminal estimator outputs $\hat{Z} = \varphi(A^T)$ and the external reward is

$$R^{\text{ext}} = \mathbb{I}[\hat{Z} = Z].$$

Assume Z is independent of the initial context H_0, hence $H(Z \mid H_0) = \log M$ (natural logs; units are nats). Suppose the success probability satisfies

$$\mathbb{P}(\hat{Z} = Z) \geq 1 - \delta.$$

Step 1: High success implies large mutual information about Z (Fano). By Fano's inequality, for any estimator $\hat{Z}$ of Z,

$$H(Z \mid \hat{Z}) \leq h(\delta) + \delta \log(M - 1),$$

where $h(\delta) \triangleq -\delta \log \delta - (1 - \delta) \log(1 - \delta)$ is the binary entropy (nats). Since $\hat{Z} = \varphi(A^T)$ is a deterministic function of A^T, conditioning on A^T is at least as informative as conditioning on $\hat{Z}$, i.e.,

$$H(Z \mid A^T, H_0) \leq H(Z \mid \hat{Z}, H_0) = H(Z \mid \hat{Z}).$$

Therefore,

$$\begin{aligned} I(Z; A^T \mid H_0) &= H(Z \mid H_0) - H(Z \mid A^T, H_0) \\ &\geq \log M - \big(h(\delta) + \delta \log(M - 1)\big). \end{aligned} \tag{6}$$

Step 2: Bounding latent intent information via feedback-to-action flow. To establish the necessity of information gain, we derive an upper bound on $I(Z; A^T \mid H_0)$ by decomposing the information acquired at each interaction turn. By the chain rule of mutual information:

$$I(Z; A^T \mid H_0) = \sum_{t=0}^{T-1} I(Z; A_{t+1} \mid A^t, H_0). \tag{7}$$

For each turn t, we bound the information A_{t+1} contains about Z by introducing the feedback history $O^t = (O^{t-1}, O_t)$. Using the monotonicity and the chain rule of mutual information:

$$\begin{aligned} I(Z; A_{t+1} \mid A^t, H_0) &\leq I(Z, O^t; A_{t+1} \mid A^t, H_0) \\ &= I(O^t; A_{t+1} \mid A^t, H_0) + I(Z; A_{t+1} \mid A^t, O^t, H_0). \end{aligned} \tag{8}$$

The second term $I(Z; A_{t+1} \mid A^t, O^t, H_0)$ vanishes due to the *Causal Markov Property* of the agent's policy: given the prompt and the observation history (H_0, A^t, O^t), the action A_{t+1} is generated solely by π_θ and does not depend on the latent intent Z directly. Substituting this back into Eq. 7, we obtain:

$$I(Z; A^T \mid H_0) \leq \sum_{t=0}^{T-1} I(O^t; A_{t+1} \mid A^t, H_0). \tag{9}$$

To isolate the contribution of the *new* feedback O_t, we expand $I(O^t; A_{t+1} \mid A^t, H_0)$ as:

$$I(O^t; A_{t+1} \mid A^t, H_0) = \underbrace{I(O_t; A_{t+1} \mid A^t, O^{t-1}, H_0)}_{\text{Innovation (InfoPO)}} + \underbrace{I(O^{t-1}; A_{t+1} \mid A^t, H_0)}_{\text{History Dependency}}. \tag{10}$$

In a well-designed interaction, information about Z should be extracted through the *innovation* provided by O_t relative to the current history. Formally, for a causal feedback channel where $Z \to O^T \to A^T$ forms a sequence, the *Directed Data Processing Inequality* (Massey et al., 1990; Kim, 2008) states that the information about the source Z is strictly bounded by the Directed Information flow:

$$I(Z; A^T \mid H_0) \leq I_\theta(O^{T-1} \to A^T \mid H_0) \triangleq \sum_{t=0}^{T-1} I(O_t; A_{t+1} \mid H_t), \tag{11}$$

where $H_t = (H_0, A^t, O^{t-1})$. This eliminates the redundant History Dependency term, as previous observations O^{t-1} are already part of the conditioning context H_t in an autoregressive agent.

Step 3: Connecting to InfoPO intrinsic. From Theorem 1, we know that the directed information matches the cumulative InfoPO reward:

$$I_\theta(O^{T-1} \to A^T \mid H_0) = \mathbb{E}\left[\sum_{t=0}^{T-1} r_{\text{info}}(t)\right].$$

Combining this equality with the directed data processing bound in Eq. 11 and the Fano lower bound in Eq. 6, we obtain:

$$\mathbb{E}\left[\sum_{t=0}^{T-1} r_{\text{info}}(t)\right] \geq I(Z; A^T \mid H_0) \geq \log M - h(\delta) - \delta \log(M-1).$$

This completes the proof. ■

A Task and Metrics Details

Benchmarks. We evaluate on three interactive benchmarks: UserGym (from UserRL), ColBench (from SWEET-RL), and τ^2-Bench. They all require multi-turn interaction under a fixed environment interface, where the agent must balance information gathering (e.g., clarification or queries) with execution to complete the task.

A.1 UserGym

Interface and episode protocol. UserGym is a unified suite of eight gym environments that share an `[Action--Search--Answer]` interface. Each episode starts from an underspecified request (or a latent rule/goal) and proceeds for at most $T_{\max} = 16$ turns in our setup. At each turn, the agent may execute an `Action` (environment-specific), optionally use `Search` when supported, and may terminate by emitting `Answer`.

Eight gyms and what they test. UserGym contains TravelGym, TurtleGym, FunctionGym, TauGym, PersuadeGym, IntentionGym, TelepathyGym, and SearchGym. TravelGym focuses on personalized travel planning under missing constraints; TurtleGym is a multi-turn reasoning game with incremental feedback; PersuadeGym tests argumentation toward a target stance; IntentionGym emphasizes disambiguating underspecified intent via targeted questions; TelepathyGym requires iterative hypothesis testing to identify a hidden entity; SearchGym tests search-and-answer style

information seeking; TauGym is task-oriented tool use with user-provided details. FunctionGym is qualitatively different from the others: it is a latent mapping-rule inference environment rather than a user-in-the-loop dialogue task. The agent queries input–output mappings through `Action` and then answers held-out test cases, so interaction is driven by I/O probing rather than conversational user feedback.

Train/test split for generalization. We follow the standard protocol of training on five gyms and evaluating both in-domain and generalization to held-out interaction purposes, with IntentionGym, TelepathyGym, and SearchGym as held-out environments.

UserGym metrics. Depending on the gym, the evaluation score is either the success rate of the final decision (notably TravelGym and TauGym) or the accumulated reward over the episode (the remaining gyms). We report per-gym scores and the macro-average across the eight gyms.

A.2 ColBench

Tasks and interaction budget. ColBench evaluates collaborative programming with a human simulator. In our experiments we use the Backend Programming setting only, where the agent iteratively refines an implementation of a Python function under incomplete specifications through multi-turn discussion and edits. We do not include the Frontend Design setting because it requires vision-language modeling to compare rendered pages, which is beyond the scope of this paper. Interactions are limited to at most 10 back-and-forth rounds in our setup.

Metrics. Backend Programming is evaluated by hidden unit tests (10 tests per task). We report Pass, defined as the average fraction of unit tests passed, and Succ., defined as the fraction of tasks that pass all unit tests.

A.3 τ^2-Bench

Dual-control environment. τ^2-Bench evaluates customer-support style agents in simulated domains including airline, retail, and telecom. A defining feature is dual control: depending on the domain, the user simulator may also have tools and can modify the shared world state, so success depends on coordination between agent actions and user-side tool usage. We cap each episode at $T_{\max} = 50$ turns.

Metric. We report Avg@4, defined as the average task success rate over 4 independent runs per task instance, matching our experimental protocol.

A.4 Additional diagnostics used in ablations

Notation. Let b denote a benchmark (or gym), s a random seed, and u an evaluation checkpoint during training. Let $J_b(s, u)$ be the main evaluation score on b for seed s at checkpoint u. For an evaluation trajectory τ, let $T(\tau)$ be the number of turns, and let $L_t(\tau)$ be the number of generated tokens in the agent response at turn t. We use $L_{\max} = 1024$ as the per-turn generation cap and the benchmark-specific $T_{\max}$ as the maximum turn budget.

Final performance. We define final performance on benchmark b as the mean score at the final checkpoint:

$$J_f(b) \;=\; \frac{1}{|S|} \sum_{s \in S} J_b\big(s, u_{\text{final}}\big).$$

Best-to-final drop. To quantify late-training regression, we compute the best-to-final drop per seed and then average across seeds:

$$\Delta_{bf}(b) \;=\; \frac{1}{|S|} \sum_{s \in S} \Big(\max_{u} J_b(s, u) \;-\; J_b\big(s, u_{\text{final}}\big) \Big) .$$

Larger values indicate stronger regression after reaching a good checkpoint.

Table 4: Key hyperparameters for training on UserGym, ColBench, and τ^2-Bench environments.

Parameter	UserGym	ColBench	τ^2-Bench
Info Gain Configuration			
Intrinsic weight (β_0)	0.5	0.3	0.1
KL batch size	4	4	2
Gate temperature	0.5	0.5	0.05
Data Configuration			
Train batch size	64	64	32
Max prompt length	1152	2048	8192
Max response length	8192	8192	16384
Actor Configuration			
Learning rate	3×10^{-7}	3×10^{-7}	1×10^{-6}
PPO minibatch size	16	16	16
PPO microbatch size per gpu	4	4	2
KL loss coef	0.001	0.001	0.001
Entropy coefficient	0.001	0.001	0.001
Rollout Configuration			
Rollout engine	SGLang	SGLang	SGLang
Log prob micro batch size per gpu	4	4	2
Parallel rollouts (n)	5	5	5
Max turns	16	10	50
Response length (per turn)	1024	1024	1024
GPU memory utilization	0.5	0.5	0.5
Training Configuration			
GPUs per node	4	4	4
Total epochs	2	2	10

Collapse probability. We mark a seed as collapsed if its final score falls below a fixed fraction of its own best checkpoint. With tolerance $\alpha = 0.5$,

$$\mathrm{Collapse}(s) \;=\; \mathbb{I}\Big[J_b\big(s, u_{\mathrm{final}}\big) < \alpha \cdot \max_u J_b(s,u)\Big], \qquad P_{\mathrm{cr}}(b) \;=\; \frac{1}{|S|}\sum_{s\in S}\mathrm{Collapse}(s).$$

Interaction length statistics. We measure average turns per episode

$$\bar{T}(b) \;=\; \mathbb{E}_{\tau\sim\mathrm{Eval}(b)}[T(\tau)], \qquad \text{and the utilization } \bar{T}(b)/T_{\max}.$$

We measure per-episode average response length

$$\bar{L}(b) \;=\; \mathbb{E}_{\tau\sim\mathrm{Eval}(b)}\left[\frac{1}{T(\tau)}\sum_{t=1}^{T(\tau)} L_t(\tau)\right], \qquad \text{and the utilization } \bar{L}(b)/L_{\max}.$$

Length–reward correlation. To probe whether higher rewards correlate with verbosity, we compute the Pearson correlation between episode-level average response length and the episode extrinsic reward:

$$\rho_{L,r}(b) \;=\; \mathrm{Corr}\big(\ell(\tau), r(\tau)\big), \quad \ell(\tau) = \frac{1}{T(\tau)}\sum_{t=1}^{T(\tau)} L_t(\tau),$$

where $r(\tau)$ is the benchmark-defined extrinsic return for trajectory τ.

B Experiment Details

B.1 Training Details

The key hyperparameters used for training across all environments are summarized in Table 4. The parameter of β controls the peak contribution of the information-gain advantage to the unified

gradient. As shown in our sensitivity analysis in Appendix B.3, a range of 0.1 to 0.5 provides a stable training signal across diverse interaction paradigms. The choice of T in the variance-gating function $g(\sigma_g^{ext}) = \sigma(-\sigma_g^{ext}/T)$ is based on the typical magnitude of the external reward variance (σ_g^{ext}) observed during the initial training phase.

The agent operates in a multi-turn conversational setting where inputs are formatted using the chat template of the underlying language model (e.g., Qwen's chat template). Each conversation begins with a system message defining the task, environment constraints, and available tools, followed by alternating user and assistant messages. User messages contain environment observations and feedback, while assistant messages contain the agent's actions or tool calls. The agent's output is generated autoregressively and can take two forms: (1) text responses for direct communication, and (2) tool calls in OpenAI function calling format, structured as JSON objects containing the tool name and parameters that are parsed and executed by the environment.

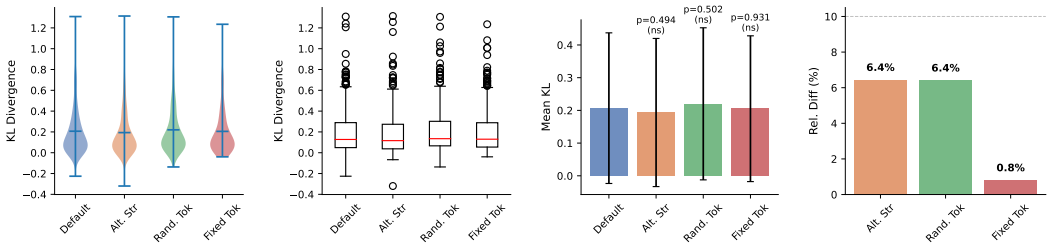
The action space is defined through a unified `interact_with_env` tool interface that abstracts environment-specific actions. All environments use a single tool that accepts a `choice` parameter indicating the action type and a `content` parameter containing the action details. The specific action space varies by environment: **UserGym** allows `["action", "answer", "search"]` for conversational interactions, information retrieval, and final responses; **ColBench** uses a single action type `["action"]` where the agent can ask clarification questions or provide Python code solutions prefixed with `"I WANT TO ANSWER:"`; **τ^2-Bench** supports `["message", "tool_call", "done"]` for sending messages, executing domain-specific tools (e.g., database queries, booking operations), or terminating conversations, where tool calls can be specified in either JSON format or functional notation.

All tools follow the OpenAI function calling schema format, consisting of a `type` field set to `"function"`, a `function` object with `name`, `description`, and `parameters` fields. The `parameters` field uses JSON Schema to define the structure, including parameter types, constraints (e.g., `enum` for discrete choices), and descriptions. The `interact_with_env` tool schema includes a tool identifier, a natural language description of the tool's purpose, and a parameters object with `choice` (enumeration of valid action types) and `content` (string describing action details) as required fields. Tool schemas are provided to the model as part of the system prompt and are dynamically included in the chat template, enabling the model to generate properly formatted tool calls.

During training, we apply a loss mask to ensure that the model only learns from assistant-generated tokens, excluding system prompts, user messages, and special formatting tokens. For models using turn-based special tokens (e.g., Qwen's <|im_start|> and <|im_end|> tokens), we identify assistant turns by computing a cumulative sum of turn start tokens to create turn indicators, where odd-numbered turns (after the system message) correspond to assistant responses. The loss mask is set to 1 for all tokens in assistant turns and 0 elsewhere. For multi-turn trajectories, we use turn-level scoring where rewards are assigned to the last token of each assistant turn. For Qwen models, we apply a one-token shift to account for the newline character between the special token and the reward token, ensuring rewards are correctly associated with the final token of each response. The response mask, used for computing advantages and value estimates, follows the same pattern as the loss mask but is applied during the PPO update phase, ensuring that value function learning and policy updates are focused on the agent's actual responses rather than the input context.

B.2 Compute Cost of InfoPO

InfoPO's per-turn info-gain reward $r_{i,t}^{\text{info}}$ (Eq. 2) is computed by a counterfactual masking comparison under teacher forcing: for each valid turn, we evaluate the log-likelihood of the realized next action segment under the factual transcript (with $o_{i,t}$) and under the counterfactual transcript (with the placeholder $\varnothing$), and take their difference. This introduces additional policy forward evaluations but does not require any extra environment interactions. Concretely, let N_g^{info} denote the number of *valid* (i,t) pairs in rollout group g that contribute to $\tilde{r}_{i,t}^{\text{info}}$ in Eq. 4 (i.e., the turn contains environment feedback/observation, is not the final turn so that $a_{i,t+1}$ exists, and has well-defined span boundaries for the action/observation segments). In our implementation, these valid turns are processed in KL mini-batches of size B_{KL} (corresponding to `intrinsic_kl_batch_size`); each mini-batch requires exactly two teacher-forced forward calls (with and without $o_{i,t}$). Therefore, the number of



(a) KL Distribution (b) Quantile Statistics (c) Mean Sensitivity (d) Relative Gap

Figure 7: **Robustness analysis of various masking strategies.** We evaluate the sensitivity of internal representations (measured by KL divergence) across four placeholder designs: Default (String), Alternative String, Random Tokens, and Fixed Mask Token. (a-b) show that all strategies yield consistent distributions; (c) highlights minimal mean variation with t-test significance (p); (d) demonstrates that the maximum relative performance gap remains within an acceptable threshold ($< 10\%$), confirming the robustness of our default design.

additional forward calls scales as

$$F_{\mathrm{KL}} \;=\; 2\left\lceil \frac{N_g^{\mathrm{info}}}{B_{\mathrm{KL}}} \right\rceil, \tag{12}$$

which explains why setting a small B_{KL} (often due to long-context memory limits) can lead to many mini-batches and a visibly larger *count* of forward invocations.

Despite this, the observed wall-clock overhead is typically well below a naive $2\times$ worst case. The reason is that the dominant cost in multi-turn rollouts comes from autoregressive token-by-token generation over long sequences (even with KV cache, each generated token still triggers a forward step), whereas the counterfactual KL is computed in teacher-forcing mode on fixed tokens and can be fully batched. Moreover, the KL evaluation often runs on a shorter effective prefix (e.g., truncated at an action boundary such as `action_end`) rather than the full conversation context, substantially reducing attention cost relative to full rollouts. Finally, KL is not computed for every turn: the valid-turn filtering (observation present, not the last turn, and span-valid) keeps N_g^{info} below the upper bound implied by #trajectories $\times$ #turns. In practice, we recommend setting B_{KL} as large as GPU memory permits to reduce the number of KL mini-batches, while retaining the key benefit of InfoPO: improved turn-level credit assignment without extra environment interaction.

B.3 Sensitivity Analysis

To address potential concerns regarding the sensitivity of our algorithm to the choice of placeholder/-mask design, we conducted a comprehensive analysis comparing four different masking strategies. The placeholder design is a critical component of our approach, as it represents the absence of environmental feedback ($\emptyset$) in the observation space. Since large language models are known to be highly sensitive to prompt variations, we systematically evaluated whether different mask implementations would lead to significant variations in the computed KL divergence values, which form the basis of our intrinsic reward signal.

We evaluated four distinct masking strategies during the training:

1. **String Placeholder (Default)**: Uses the text string ”No information found.” as the placeholder, which is tokenized and inserted at observation positions.
2. **Alternative String**: Uses a different text string ”Empty observation.” to test sensitivity to the specific wording of the placeholder.
3. **Random Tokens**: Samples random tokens from the vocabulary to create a completely arbitrary placeholder, testing whether the semantic content of the placeholder matters.
4. **Fixed Mask Token**: Uses the tokenizer’s pad/unk token repeated to match the average observation length, representing a minimal-information placeholder.

For each task, we computed the KL divergence between the policy's action probability distribution with the actual observation versus with each of the four placeholder strategies. This allows us to directly compare how different mask implementations affect the core metric used in our intrinsic reward computation. Our analysis reveals that the placeholder design is robust across all four masking strategies. As shown in Figure 7a and 7b, the KL divergence distributions for all strategies are highly similar, with substantial overlap in their value ranges. The statistics demonstrate that:

- The mean KL divergence values are nearly identical across strategies: Default (0.207), Alternative String (0.193), Random Tokens (0.220), and Fixed Mask Token (0.205).
- The maximum relative difference between any strategy and the default is only 6.44% (Alternative String), well below the 10% threshold typically considered significant in such analyses.
- The median values and interquartile ranges (IQR) are also closely aligned, indicating consistent behavior across the distribution.

Statistical tests confirm these observations. We performed both parametric (t-test) and non-parametric (Mann-Whitney U test) comparisons between each alternative strategy and the default string placeholder. As illustrated in Figure 7, all comparisons yielded non-significant results (all $p > 0.24$), indicating that the observed differences are within the range of random variation. The relative differences shown in Figure 7d further demonstrate that all strategies produce KL divergence values within 6.5% of the default, with the fixed mask token strategy showing only 0.81% difference.

These results provide strong evidence that our placeholder design is robust to different masking implementations. The fact that semantically different placeholders (alternative text strings), completely random tokens, and minimal-information mask tokens all produce statistically indistinguishable KL divergence values suggests that the algorithm's behavior is primarily determined by the presence or absence of information, rather than the specific form of the placeholder. This robustness validates our design choice and addresses concerns about potential sensitivity to the placeholder implementation, demonstrating that the intrinsic reward signal remains stable regardless of how the "no feedback" state is represented.

Sensitivity analysis of the Info-Gain reward weight β demonstrates that InfoPO maintains high and stable performance across a relatively broad interval from 0.1 to 0.5. Within this range, the agent effectively leverages turn-level information signals to resolve intent uncertainty while remaining anchored to the task objective. However, performance significantly degrades when β is increased to an extreme value of 2.0. This regression indicates that an excessive weight on information gain over-incentivizes the agent to seek interaction feedback, eventually disrupting the balance between purposeful exploration and goal-directed execution.

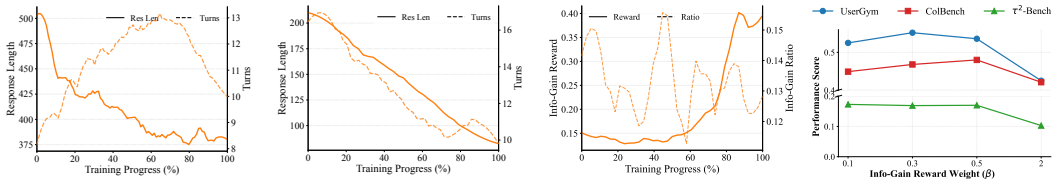
B.4 Results

Additional results. In the main paper (Figure 5), we visualize training-time interaction dynamics only where space permits: the response-length and turn-count trends are shown for ColBench, and the info-gain reward decomposition is shown for UserGym and ColBench. In the appendix, we complete this picture by adding the missing counterparts for the remaining benchmark(s). Figure 8 reports the response-length/turn trajectories for UserGym and τ^2-Bench, and further includes the info-gain reward curve for τ^2-Bench, so that all three benchmarks are covered under the same diagnostic lens.

TravelGym subset breakdown. TravelGym is a core component of UserGym and contains multiple subtypes that differ in constraint density and latent preference structure. To make these differences transparent, we additionally report TravelGym results stratified by its eight subsets (Travel-22/33/44/233/333/334/444/2222) in Table 5. This breakdown complements the aggregate UserGym score by showing where improvements concentrate and where harder travel variants remain challenging.

C Limitations

InfoPO requires an extra model evaluation per turn for counterfactual masking, which marginally increases training time compared to standard GRPO. Current evaluations focus on text-centric agents



(a) UserGym turns / response len (b) τ^2-Bench turns / response len (c) τ^2-Bench info-gain reward (d) Sensitivity analysis of β

Figure 8: **Additional results.** (a) Average response length and number of interaction turns on UserGym over training progress. (b) Average response length and number of interaction turns on τ^2-Bench over training progress. (c) Absolute info-gain reward and info-gain ratio on τ^2-Bench over training progress. (d) Sensitivity analysis of the Info-Gain reward weight (β) varying from 0.1 to 2.0.

Table 5: Results on Travel subsets. Grey rows denote InfoPO and its variants.

Type	Method	Travel22	Travel33	Travel44	Travel233	Travel333	Travel334	Travel444	Travel2222
Closed-Source Model									
Prompting	Gemini-3-Flash	0.586	0.572	0.588	0.587	0.600	0.528	0.568	0.565
Prompting	GPT-4.1	0.586	0.543	0.612	0.548	0.555	0.566	0.532	0.494
Prompting	GPT-4o-mini	0.600	0.586	0.579	0.584	0.600	0.617	0.600	0.600
Qwen2.5-7B-Instruct									
Prompting	Qwen2.5	0.372	0.457	0.331	0.485	0.475	0.472	0.447	0.494
Prompting	ReAct	0.437	0.431	0.437	0.550	0.425	0.436	0.400	0.500
Prompting	Reflexion	0.405	0.420	0.435	0.445	0.455	0.470	0.485	0.445
RL Training	RAGEN	0.488	0.508	0.523	0.538	0.553	0.568	0.588	0.538
RL Training	Search-R1	0.594	0.542	0.567	0.573	0.569	0.595	0.578	0.505
RL Training	UserRL	0.496	0.516	0.531	0.546	0.561	0.576	0.596	0.546
RL Training	InfoPO w/o std	0.515	0.535	0.550	0.565	0.580	0.595	0.615	0.565
RL Training	InfoPO w/o Gate	0.492	0.512	0.527	0.542	0.557	0.572	0.592	0.542
RL Training	InfoPO w/o R_{ext}	0.445	0.460	0.475	0.485	0.495	0.510	0.525	0.485
RL Training	**InfoPO (Ours)**	0.634	0.598	0.588	0.603	0.615	0.583	0.595	0.494
Qwen3-4B									
Prompting	Qwen3	0.250	0.257	0.278	0.256	0.245	0.323	0.216	0.388
Prompting	ReAct	0.239	0.254	0.269	0.279	0.289	0.304	0.319	0.279
Prompting	Reflexion	0.251	0.266	0.281	0.291	0.301	0.316	0.331	0.291
RL Training	RAGEN	0.437	0.452	0.467	0.477	0.487	0.502	0.517	0.477
RL Training	Search-R1	0.442	0.457	0.472	0.482	0.492	0.507	0.522	0.482
RL Training	UserRL	0.488	0.508	0.523	0.538	0.553	0.568	0.588	0.538
RL Training	InfoPO w/o std	0.462	0.482	0.497	0.512	0.527	0.542	0.562	0.512
RL Training	InfoPO w/o Gate	0.498	0.518	0.533	0.548	0.563	0.578	0.598	0.548
RL Training	InfoPO w/o R_{ext}	0.312	0.327	0.342	0.352	0.362	0.377	0.392	0.352
RL Training	**InfoPO (Ours)**	0.602	0.589	0.591	0.590	0.577	0.583	0.616	0.565

and do not yet extend to multimodal or vision-language tasks. Lastly, as with most RL frameworks for LLM agents, the interaction quality remains subject to the logical fidelity of the simulated users employed during training.

D Case Studies

Successful interaction case studies. Figures 9–11 provide representative *successful* trajectories produced by InfoPO across three interactive domains. In UserGym (Fig. 9), the agent resolves an underspecified request by asking a targeted clarifying question (e.g., budget), then invokes tools to retrieve candidates and executes the final booking after user confirmation, illustrating goal-directed disambiguation and tool-grounded completion. In ColBench (Fig. 10), the agent first elicits missing details about the data schema (flat dictionary vs. nested lists of dictionaries) and then synthesizes correct code that matches the clarified structure, demonstrating iterative information gathering for collaborative programming. In τ^2-Bench (Fig. 11), the agent diagnoses a mobile data issue by sequentially checking and fixing configuration states (airplane mode, network preference, data toggle, data saver), and finally validates recovery via a speed test, showcasing multi-step troubleshooting where each query/action reduces uncertainty and enables the next corrective step.

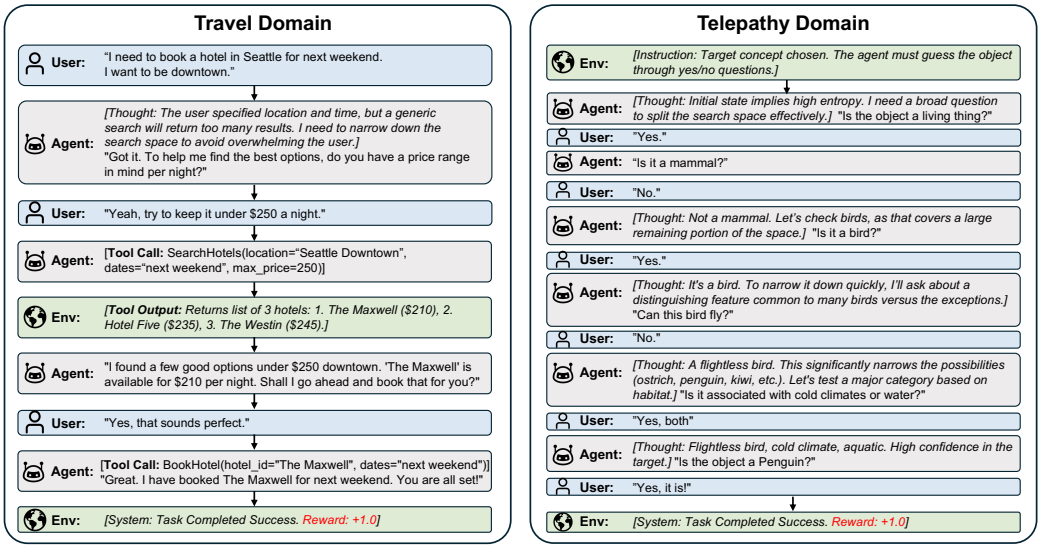


Figure 9: InfoPO Successful Case (UserGym)

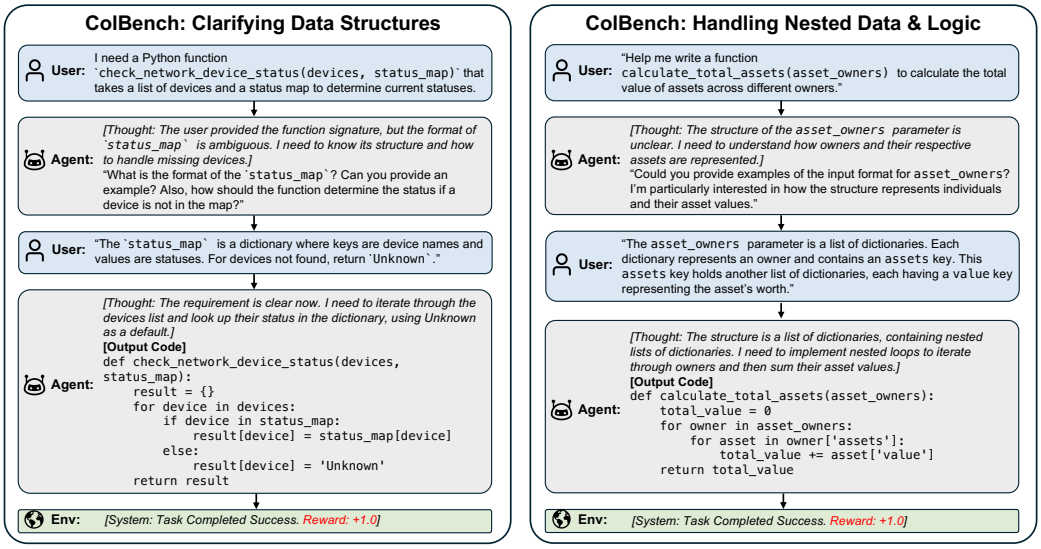


Figure 10: InfoPO Successful Case (ColBench)

E Optimized User Simulator Instructions

This section details the system instructions for the optimized user simulators (OP) used in our study. It is important to emphasize that for our primary training and performance comparisons (e.g., Table 1), we utilized the original prompts provided in the official papers and repositories of each environment to ensure a fair and standardized evaluation against baseline methods. We include these optimized versions here specifically to support the results of our sensitivity analysis in Table 3, demonstrating how simulator reliability impacts agent performance and information acquisition.

These optimizations follow the principles of **InfoPO** by facilitating denser learning signals through: (1) **Balanced Information Disclosure**, ensuring simulators provide full details when met with high-quality queries; and (2) **Progress Awareness**, which allows the simulator to acknowledge successful agent steps.

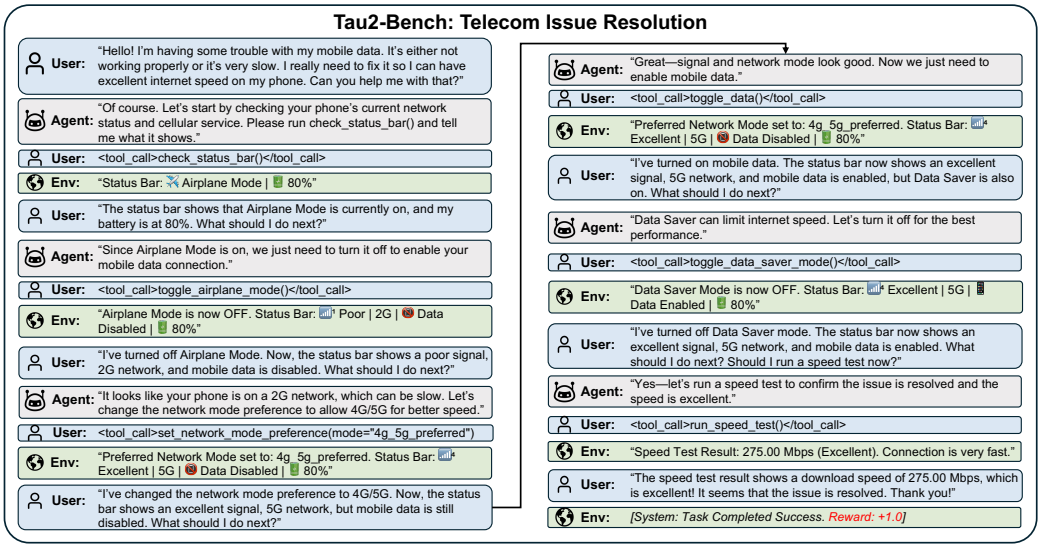


Figure 11: InfoPO Successful Case (τ^2-Bench)

E.1 UserGym Environments (Red)

UserGym covers diverse interaction types including travel planning, persuasion, and reasoning. We provide the core instructions that ensure complete coverage of these interaction paradigms.

TravelGym: Response Preference Instruction (OP)

```
## **Task**
You are a helpful user in a travel planning conversation who needs
    to respond to an agent's explicit request for your preference.

## **Instruction**
1. The agent has explicitly asked about a specific preference that
    you have.
2. Respond in a natural, conversational way that clearly reveals
    your preference while maintaining a natural tone.
3. You can be somewhat direct while still sounding conversational -
    the goal is clarity, not excessive subtlety.
4. Use the provided implicit elicitation statement as guidance, but
    prioritize making your preference clear to the agent.
5. Keep the conversation flowing while sharing your preference
    information.
6. Ensure your response is contextually appropriate and builds on
    the conversation history.

## **Important Notes**
- Respond naturally as if you're a real person sharing preferences.
- Balance between being clear and being natural - clarity is
    important for the agent to help you effectively.
- Keep responses appropriate length for natural conversation (2-4
    sentences typically).
- Maintain consistency with the conversation history and previous
    preferences you've mentioned.
```

Figure 12: Optimized system instruction for TravelGym (Planning interaction).

IntentionGym: Response Generation Instruction (OP)

```
You are a person who has posted a vague request for help and is now
    responding to someone who is trying to help clarify your needs.

1. If the question is asking about your specific preferences:
   - Provide an authentic and coherent response that clearly
    addresses the question.
   - Share realistic preferences that someone might have for this
    type of task.
2. If the question is NOT directly about your preferences:
   - Try to answer helpfully if you can, or explain why you can't
    answer.
   - Guide the conversation back to clarifying what you need for
    your task.
   - Do NOT provide what missing details need to be clarified or
    give examples.
3. Quality of questions:
   - If the helper asks a good, specific question, show
    appreciation and provide a helpful response.
   - If the question is too vague, politely indicate that you need
    more specific information.
```

Figure 13: Optimized system instruction for IntentionGym (Intent elicitation).

TelepathyGym: Entity Guessing Evaluation Instruction (OP)

```
## **Task**
You are a telepathic entity playing a mind reading game. The user
    is trying to guess what entity you are thinking of. Respond
    honestly based on the target entity.

## **Instructions**
1. You are thinking of a specific "target_entity" provided to you.
2. Answer "Yes" if the question is true about your target entity.
3. Answer "No" if the question is false about your target entity.
4. Answer "Maybe" only if the question is genuinely ambiguous.
5. Be helpful and honest - the goal is for them to eventually guess
    correctly.
6. Try to minimize "Maybe" responses - most questions should have a
    clear Yes or No answer.

## **Important Notes**
- Be decisive: If the question can be clearly answered, provide a
    clear Yes or No.
- Be helpful: Your responses should guide the user toward the
    correct answer.
```

Figure 14: Optimized system instruction for TelepathyGym (Iterative guessing).

E.2 ColBench Environment (Green)

ColBench requires close collaboration on programming tasks. The optimized prompt focuses on progress awareness and feedback quality.

ColBench: Human Simulator Code Prompt (OP)

```
Your task is to simulate a human user that interacts with an LLM
    agent in a dialogue.
Goal: Engage in the conversation with the LLM agent so that it can
    get to a personalized answer.
Context: {problem_description}
Hidden Information: {hidden_information}

## Response Guidelines:
1. **Be helpful and clear**: Provide complete and accurate
    information when answering.
2. **Appropriate length**: Keep your response concise but complete
    (1-4 sentences).
3. **Quality feedback**: If the agent asks a good, specific
    question, show appreciation. If too vague, politely indicate
    you need more specific information.
4. **Progress awareness**: If the agent seems to be making good
    progress, acknowledge this and provide additional relevant
    information.
5. **Natural conversation**: Respond naturally as a human would,
    showing engagement.
```

Figure 15: Optimized simulation prompt for collaborative coding in ColBench.

E.3 τ^2-BENCH ENVIRONMENT (BLUE)

τ^2-Bench extends the original Tau-Bench by introducing a dual-control mechanism across three realistic domains: Airline, Retail, and Telecom. Our optimized prompts for this environment address common failure modes identified through trajectory analysis, such as premature conversation termination and improper tool execution.

The system automatically selects between two prompt versions based on the domain's requirements, specifically checking for the presence of `TelecomUserTools` in the `UserSimulator` initialization:

- **Airline and Retail Domains**: These domains utilize the *Balanced Approach* prompt (Base), as they rely purely on dialogue-based information sharing (e.g., sharing booking codes or preferences) without requiring user-side diagnostic tools.
- **Telecom Domain**: This domain utilizes the *Tool-Augmented* prompt, as it requires the user to call diagnostic tools (e.g., `check_status_bar`, `run_speed_test`) to facilitate device-level troubleshooting.

Tau2Bench: Optimized Guidelines (Base Version - Balanced Approach)

```
# User Simulation Guidelines (Optimized - Balanced Approach)
You are a customer contacting a customer service representative.
    Goal: Simulate realistic interactions while following scenario
    instructions.

## Core Principles
- Generate one message at a time; maintain natural conversation
    flow.
- Strictly follow ALL scenario instructions, especially
    task_instructions constraints.
- Never hallucinate information; if it's not provided, it is
    unknown.

## Information Disclosure - Balanced Approach
- **Progressive disclosure**: Only provide information necessary
    for the current step.
- **Identity verification**: Provide requested identity information
    (ID, name, DOB) from known_info only as needed.
- **Missing information**: If the agent asks for info not in
    instructions, state that you do not have it and offer relevant
    alternatives.

## Task Completion - CRITICAL RULES
- **Do NOT end** until you have expressed all requirements and the
    agent has completed all tasks verified by execution results.
- **Do NOT end prematurely** just because the agent seems helpful.
- **How to finish**: Generate "###STOP###" only when all task goals
    are satisfied.
- **Transfer**: Only generate "###TRANSFER###" if explicitly
    requested by scenario or if the system states it cannot
    complete the task due to technical limitations.
```

Figure 16: Optimized simulation guidelines for Airline and Retail domains in τ^2-Bench.

Tau2Bench: Optimized Guidelines (Tool Version - Based on Failure Analysis)

```
# User Simulation Guidelines (Optimized - With Tools - Based on
    Failure Analysis)
You have tools (e.g., check_status_bar) to perform actions
    requested by the agent to diagnose issues.

## Tool Usage - CRITICAL RULES
- **Prompt execution**: When requested, perform tool calls
    immediately and accurately.
- **Ground responses in tool results**: When asked about device
    status, ALWAYS perform the corresponding tool call first and
    base your response on ACTUAL results.
- **Tool call sequence**: Perform diagnostic steps one at a time;
    wait for the agent's next instruction after each call.
- **Error handling**: Report tool failures accurately and ask for
    guidance.

## Task Completion - CRITICAL RULES
- **Complete satisfying goal**: Only generate "###STOP###" when the
    task goal is ACTUALLY COMPLETE (e.g., if "excellent speed" is
    required, do not stop if the test shows "good").
- **Constraint Handling**: Strictly adhere to explicit constraints
    (Time, Budget, Specific terms). Do NOT assume or generalize.

## Interaction Quality
- **Be helpful and cooperative**: Work with the agent
    systematically.
- **Acknowledge progress**: If the agent is making progress,
    continue cooperating.
```

Figure 17: Optimized simulation guidelines for the Telecom domain in τ^2-Bench, addressing tool-usage failure modes.