# CALIBRATED INTERACTIVE RL: A DISTRIBUTION-SHIFT PERSPECTIVE ON MULTI-TURN DIALOGUE

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Developing highly interactive LLM-based agents requires learning how their responses shape subsequent dialogue. Reinforcement learning approaches use fixed offline contexts or collect trajectories with a user simulator. We analyze two sources of distribution shift in these settings: policy-induced shift between offline and policy-generated histories, and simulator-induced shift between training and reference user dynamics. Finite-horizon return bounds characterize how these discrepancies propagate through a conversation. Motivated by this analysis, we propose Calibrated Interactive RL, a two-stage framework that couples simulator calibration with interactive policy optimization. We first calibrate a user simulator to model task-conditioned responses in multi-turn dialogue, then train the dialogue policy with Group Relative Policy Optimization through interaction with the frozen simulator. We evaluate calibration through simulated interactions on MediumDocEdit-Chat and MATH-Chat. With a Gemma-3-4B policy, calibration improves MATH-Chat accuracy from 89.3% to 91.5%. On MediumDocEdit-Chat, it raises BLEU from 26.1 to 34.6 and reduces assistant token use from 4.00k to 2.27k relative to uncalibrated interaction. These results support simulator calibration as a useful component of multi-turn policy training.

## 1 INTRODUCTION

A long-standing goal in language-model research is to develop highly interactive agents that sustain coherent, goal-directed multi-turn conversations (Li et al., 2025; Laban et al., 2025). These agents must interpret incomplete requests, ask useful questions, and revise their responses as the user's intent becomes clearer. Reinforcement learning (RL) can optimize this process through task outcomes and feedback (Ouyang et al., 2022; Bai et al., 2022; Lee et al., 2023). Each response also shapes the context of subsequent decisions, making the generation of training conversations a central part of multi-turn RL.

Two common procedures generate these training contexts differently. **Static Context RL** optimizes responses conditioned on fixed offline histories, which remain tied to the data-generating policy. **Interactive RL** generates trajectories through a closed loop between the policy and a user simulator (Shani et al., 2024; Zhou et al., 2024). The policy can then encounter the follow-up requests and corrections elicited by its own responses. Table 1 summarizes these choices of training histories and user models.

These procedures involve two connected sources of distribution shift. **Policy-induced shift** arises when a learned policy generates histories that differ from offline contexts, including follow-up corrections elicited by its own mistakes. **Simulator-induced shift** arises when training and reference user behavior differ. A simulator that supplies clear, complete requests may create easier interactions than users who omit constraints, make mistakes, or reveal their needs gradually. A policy trained in this environment may have little practice clarifying ambiguous requests. Generating trajectories on-policy does not remove this mismatch: calibration addresses the distribution of opening requests and subsequent feedback, while interactive RL follows the policy's evolving histories.

We analyze these two sources through finite-horizon bounds on expected return, building on distribution-propagation arguments from imitation learning (Ross & Bagnell, 2010; Ross et al., 2011). The analysis separates changes in the policy from changes in user dynamics and makes their dependence on the dialogue horizon explicit. It also distinguishes cumulative per-turn rewards from



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Table 1: **Training histories and user models in multi-turn RL.** Interactive training generates histories through policy rollouts; simulator calibration fits the feedback model to user responses in interaction data.

| Paradigm | Training histories | Context and feedback |
|---|---|---|
| Static Context RL | Fixed offline histories | Policy updates condition on stored contexts; $\Delta_t$ describes their mismatch with policy-generated histories. |
| Interactive RL | Rollouts with a prompted simulator | The current policy generates contexts, with user feedback supplied by the prompted model. |
| Calibrated RL | Rollouts with an SFT-calibrated simulator | SFT fits the user model to interaction data before interactive policy training; $\delta_t$ describes simulator mismatch. |

the terminal task scores used in our experiments. This perspective motivates a training design that combines policy-generated interaction histories with a learned user simulator.

To this end, we propose **Calibrated Interactive RL**. The framework first calibrates a user simulator through supervised fine-tuning on multi-turn interaction data. Calibration models how a user's response depends on the task, dialogue history, and assistant action. We then freeze the calibrated simulator and optimize the dialogue policy with Group Relative Policy Optimization (GRPO) (Shao et al., 2024). During policy learning, the simulator provides feedback on the policy's own responses, allowing each trajectory to reflect the dialogue choices made along the way. This design separates learning the interaction environment from optimizing behavior within it, while retaining complete conversations as the unit of policy training.

We evaluate the framework on MediumDocEdit-Chat and MATH-Chat, which require iterative document editing and conversational mathematical problem solving. The comparisons examine static histories, interaction with a prompted simulator, and interaction with the calibrated simulator. Calibration improves editing BLEU from 26.1 to 34.6 and math accuracy from 89.3% to 91.5% relative to uncalibrated interaction, while reducing assistant token use on both tasks. The static-context baseline provides a broader comparison of training configurations: its offline logs come from a different user model, so its contrast with interactive training combines changes in history generation and user dynamics.

Our contributions are:

- A distribution-shift analysis of multi-turn dialogue RL that separates policy-induced and simulator-induced discrepancies, with return bounds for cumulative and terminal rewards.
- Calibrated Interactive RL, a two-stage framework that fits a user simulator to multi-turn interaction data and optimizes the policy through interaction with the frozen simulator.
- An empirical comparison of fixed-context, prompted interactive, and calibrated interactive training on document editing and mathematical dialogue, measuring task performance and assistant token usage.

## 2 Related Work

**RL for multi-turn dialogue.** RL from human and AI feedback has become a central approach to improving language-model responses (Ouyang et al., 2022; Bai et al., 2022; Lee et al., 2023). Extending these methods to dialogue requires accounting for how a response affects later interactions. Offline approaches use implicit language Q-learning (Snell et al., 2022), task-oriented offline RL (Verma et al., 2022), multi-turn preference optimization (Shi et al., 2024), or offline RL on imagined conversations (Hong et al., 2023). Their performance depends on dataset coverage and the treatment of off-policy contexts (Levine et al., 2020). Online RL also grounds language models in interactive environments (Carta et al., 2023), while multi-turn RL optimizes policies over successive dialogue decisions (Shani et al., 2024; Zhou et al., 2024). CollabLLM (Wu et al., 2025) uses simulated future interactions to construct multi-turn-aware rewards for collaboration; our comparison adopts its offline DPO setting on the same policy backbone. Calibrated Interactive RL focuses on learning the user simulator that supplies feedback during closed-loop policy training.

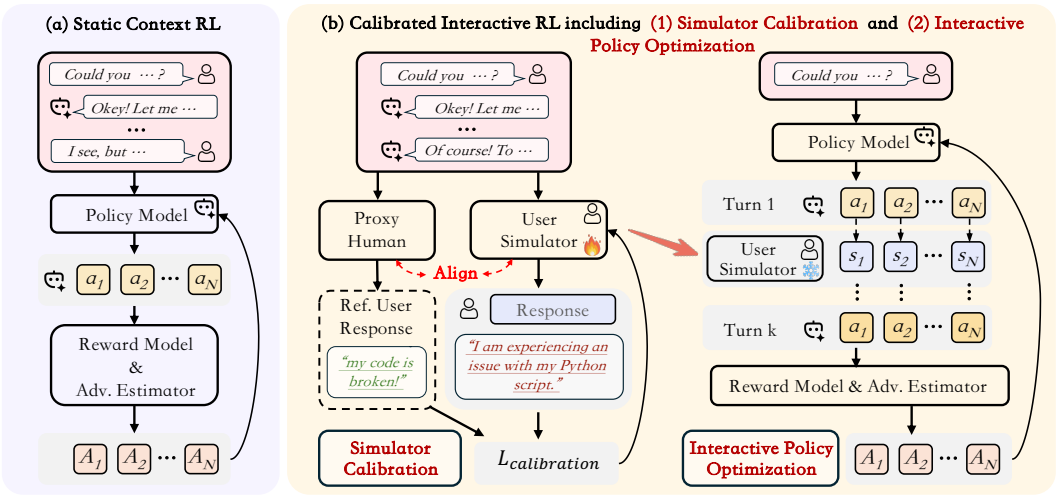


Figure 1: **Comparison of reinforcement learning frameworks for multi-turn dialogue.** (a) Static Context RL conditions policy updates on fixed offline histories. (b) Calibrated Interactive RL fits a user simulator to multi-turn interaction data, then freezes it for interactive policy optimization. In our implementation, "Proxy Human" denotes the Qwen3-235B reference user model, and "Align" denotes supervised behavioral cloning.

**Distribution shift and exposure bias.** The difference between training on reference histories and generating under a learned policy is closely related to exposure bias in sequence prediction (Bengio et al., 2015; Ranzato et al., 2015). In imitation learning, classical analyses characterize how local policy discrepancies propagate through the state distribution (Ross & Bagnell, 2010; Ross et al., 2011). For cumulative bounded rewards, these arguments yield worst-case bounds that can scale quadratically with the horizon. Interactive data collection changes the distribution on which the learner receives supervision. We apply this perspective to multi-turn dialogue by separately analyzing policy changes and changes in the user transition kernel. The resulting distinction connects the choice of training histories to the choice of simulator, and the terminal-reward specialization matches outcome-based dialogue training.

**Learning user simulators.** User simulators have long supported the training of dialogue policies (Shi et al., 2019; Tseng et al., 2021). LLM-based simulation extends the range of language and human behaviors available for such environments (Hu et al., 2023; Park et al., 2023; Gandhi et al., 2026). Related work studies role-playing agents (Shao et al., 2023; Shanahan et al., 2023) or improves language models through self-play (Chen et al., 2024). Our simulator calibration uses supervised behavioral cloning to learn history-conditioned user responses from multi-turn interaction data for subsequent policy rollouts. The simulator's feedback is important because agreement, clarification, and persistence change the conversation presented to the learner; sycophancy (Sharma et al., 2023) and reward exploitation (Wang et al., 2026) are relevant failure modes. Multi-turn benchmarks such as MT-Bench-101 (Bai et al., 2024) and simulated feedback frameworks such as AlpacaFarm (Dubois et al., 2023) further motivate distinguishing simulation, task scoring, and automated judging in experimental protocols.

## 3 Distribution Shift in Multi-turn RL

We analyze how dialogue returns change when replacing either the policy or the user model. Standard finite-horizon distribution-propagation arguments (Ross & Bagnell, 2010; Ross et al., 2011) give a common treatment of these two sources of shift and distinguish cumulative rewards from terminal task scores. Appendix A provides the proofs.

### 3.1 Setup and Notation

We condition each comparison on a fixed task $z = (x, y^*)$, where $x$ is the task input and $y^*$ is its reference answer or document. The policy sees the conversation, not the private reference $y^*$. We suppress dependence on $z$ below. For a task distribution $\rho$, we first compute each divergence conditionally on $z$, then average the resulting bounds over $z \sim \rho$.

A dialogue contains at most $H \geq 1$ assistant actions, indexed by $t = 0, \ldots, H-1$. The history $h_t$ contains the initial user request and all messages preceding the current assistant response. The policy $\pi(a \mid h)$ is a distribution over complete assistant messages, and samples $a_t \sim \pi(\cdot \mid h_t)$. The user kernel $P_t(h' \mid h_t, a_t)$ gives the probability of the next history $h'$, which appends $a_t$, any user reply, and the stopping status. The initial history is the opening request, with distribution $\mu$. Stopped transcripts are retained and padded with shared dummy actions and transitions to horizon $H$; padding contributes no further cumulative reward.

Let $d_t^\pi(h)$ be the probability of history $h$ under this process, so $d_0^\pi = \mu$. The joint probability of that history and an assistant response is $q_t^\pi(h, a) = d_t^\pi(h)\pi(a \mid h)$. For any joint distribution $q$, define the transition operator

$$(\mathcal{T}_t q)(h') = \sum_{h,a} P_t(h' \mid h, a) q(h, a), \qquad d_{t+1}^\pi = \mathcal{T}_t q_t^\pi .$$

The sums range over conversation histories and assistant messages. We use $\|p - q\|_1 = \sum_v |p(v) - q(v)|$, where $v$ ranges over the common outcome space; this is twice total variation distance.

For cumulative rewards, define $J(\pi) = \mathbb{E}[\sum_{t=0}^{H-1} r_t(h_t, a_t)]$. Here $r_t$ scores the current history and action before the next user transition, and $R_{\max} \geq 0$ bounds its magnitude: $|r_t| \leq R_{\max}$. Compared processes share this task-conditional reward function. A separate terminal-score formulation below covers scoring the final answer after the dialogue stops.

### 3.2 Policy-Induced Shift

Fix the user kernels and initial distribution, and compare $\pi$ with a reference policy $\pi'$. If offline histories are generated by $\pi'$ in this environment, $d_t^{\pi'}$ is their population distribution. Define

$$\Delta_t = \|d_t^\pi - d_t^{\pi'}\|_1, \qquad \varepsilon_t = \mathbb{E}_{h \sim d_t^{\pi'}} \|\pi(\cdot \mid h) - \pi'(\cdot \mid h)\|_1 .$$

The expectation in $\varepsilon_t$ is taken over reference-policy histories. At each such history, it compares the two conditional action distributions before averaging. In contrast, $\Delta_t$ compares histories obtained by running the two policies. This distinguishes a local response difference at a matched context from the history mismatch produced by earlier responses.

**Theorem 3.1** (Return difference under policy shift)**.** *For policies sharing the user kernels, initial distribution, and reward function,*

$$|J(\pi) - J(\pi')| \leq R_{\max} \sum_{t=0}^{H-1} (\Delta_t + \varepsilon_t) \leq R_{\max} \sum_{i=0}^{H-1} (H - i)\varepsilon_i . \tag{1}$$

Adding and subtracting $d_t^{\pi'}(h)\pi(a \mid h)$ gives the pointwise decomposition

$$q_t^\pi - q_t^{\pi'} = (d_t^\pi - d_t^{\pi'})\pi + d_t^{\pi'}(\pi - \pi').$$

Taking the $\ell_1$ norm and summing over actions bounds this joint divergence by $\Delta_t + \varepsilon_t$. The transition operator is non-expansive: conditional probabilities sum to one, so averaging the signed distribution difference gives

$$\|\mathcal{T}_t q - \mathcal{T}_t q'\|_1 \leq \|q - q'\|_1 .$$

Applying this inequality to the two joint distributions yields $\Delta_{t+1} \leq \Delta_t + \varepsilon_t$ and hence $\Delta_t \leq \sum_{i<t} \varepsilon_i$. A policy difference at step $i$ can affect its immediate reward and the $H - 1 - i$ subsequent rewards. If $\varepsilon_t \leq \varepsilon$, the bound becomes $R_{\max} H(H+1)\varepsilon/2$.

This bound controls the magnitude of the return difference under common user dynamics; its direction depends on the task. Interactive rollout collection uses histories generated by the rollout policy, providing training contexts that include continuations of its own answers.

### 3.3 SIMULATOR-INDUCED SHIFT

Fix $\pi$ and compare a reference user model $P_{\mathrm{ref},t}$ with a training simulator $P_{\mathrm{sim},t}$. Their initial distributions may differ. Write $\Delta_0^{\mathrm{env}} = \|\mu_{\mathrm{ref}} - \mu_{\mathrm{sim}}\|_1$ and define

$$\delta_t = \mathbb{E}_{(h,a)\sim q_t^{\pi,\mathrm{ref}}} \|P_{\mathrm{ref},t}(\cdot \mid h,a) - P_{\mathrm{sim},t}(\cdot \mid h,a)\|_1.$$

Here $q_t^{\pi,\mathrm{ref}}(h,a) = d_t^{\pi,\mathrm{ref}}(h)\pi(a \mid h)$: the history comes from running the fixed policy with the reference user model, and the action comes from that same policy. Both user kernels are evaluated at this shared history–action pair. Thus $\delta_t$ measures a conditional response difference on reference interactions, with frequently visited pairs receiving greater weight.

**Theorem 3.2** (Return difference under simulator shift)**.** *For a fixed policy and a shared task-conditional reward function,*

$$|J_{\mathrm{ref}}(\pi) - J_{\mathrm{sim}}(\pi)| \le R_{\max}\left[H\Delta_0^{\mathrm{env}} + \sum_{i=0}^{H-2}(H-1-i)\delta_i\right]. \tag{2}$$

*The sum is zero for $H = 1$.*

Let $\Delta_t^{\mathrm{env}} = \|d_t^{\pi,\mathrm{ref}} - d_t^{\pi,\mathrm{sim}}\|_1$. To obtain its recurrence, introduce the intermediate distribution $\mathcal{T}_{\mathrm{sim},t} q_t^{\pi,\mathrm{ref}}$. Replacing the user kernel while retaining the reference joint distribution contributes at most $\delta_t$. Replacing that joint distribution with the simulator-induced one then propagates the existing history mismatch. The same non-expansiveness argument bounds this second contribution by $\Delta_t^{\mathrm{env}}$. The recurrence $\Delta_{t+1}^{\mathrm{env}} \le \Delta_t^{\mathrm{env}} + \delta_t$ yields $\Delta_t^{\mathrm{env}} \le \Delta_0^{\mathrm{env}} + \sum_{i<t}\delta_i$. Since the policy is shared, the return gap is bounded by $R_{\max}\sum_t \Delta_t^{\mathrm{env}}$. The weight $H-1-i$ counts rewards following transition $i$. With shared openings and $\delta_t \le \delta$, the bound is $R_{\max}H(H-1)\delta/2$.

For example, at $H = 3$, a policy difference at step zero can affect $r_0$, $r_1$, and $r_2$, whereas a user-transition difference at step zero can affect only $r_1$ and $r_2$. The weights count the remaining reward evaluations, yielding the triangular sums in the two cumulative bounds.

### 3.4 TERMINAL REWARDS AND INITIAL USER MESSAGES

Let $\tau$ denote a complete padded dialogue and $Q_H$ its distribution. A terminal score $F(\tau)$ assigns one value to that dialogue, with $|F(\tau)| \le R_{\max}$. A score difference is at most $R_{\max}$ times the corresponding $\ell_1$ distance. Applying the preceding recurrences gives

$$|\mathbb{E}_\pi F - \mathbb{E}_{\pi'} F| \le R_{\max}\sum_{i=0}^{H-1}\varepsilon_i, \tag{3}$$

$$|\mathbb{E}_{\mathrm{ref}} F - \mathbb{E}_{\mathrm{sim}} F| \le R_{\max}\left[\Delta_0^{\mathrm{env}} + \sum_{i=0}^{H-1}\delta_i\right]. \tag{4}$$

The policy comparison retains shared user dynamics and initial histories. Both comparisons use the same task-conditional scoring rule. For a stochastic judge with realized terminal score $R$, take $F(\tau) = \mathbb{E}[R \mid z, \tau]$ under its shared conditional scoring distribution.

A terminal score counts each trajectory once, giving linear horizon dependence under uniform local divergence bounds. Padding represents every dialogue at the same horizon, but $F$ is evaluated only once. An early change can still alter later responses and the final answer; the analysis measures its effect through the complete trajectory distribution. The extra horizon factor in the cumulative bounds comes from summing discrepancies over successive reward evaluations, which is absent for a single terminal score. The initial-distribution term captures systematic differences in opening requests. At the hard message limit, our protocol records the last assistant answer and stops without another user response; this final transition is shared and contributes zero simulator divergence. Appendix A.5 formalizes stopping.

**Reward ranges.** The reward range also bounds cumulative return differences by $2HR_{\max}$ and terminal-score differences by $2R_{\max}$. Each bound above can therefore be capped at the corresponding

value. For nonnegative terminal scores $F \in [0, R_{\max}]$, as in our task rewards, centering gives a sharper result. For two trajectory distributions $Q$ and $Q'$, each summing to one,

$$|\mathbb{E}_Q F - \mathbb{E}_{Q'} F| = \left| \sum_\tau [Q(\tau) - Q'(\tau)] \left[ F(\tau) - \frac{R_{\max}}{2} \right] \right| \leq \frac{R_{\max}}{2} \|Q - Q'\|_1 \leq R_{\max}.$$

Thus the right-hand sides of Equations (3)–(4) can be halved and capped at $R_{\max}$ for these scores. Applying the same centering at each step gives the corresponding refinement for nonnegative cumulative rewards.

**Connection to simulator calibration.** SFT fits user responses on calibration contexts, whereas $\delta_t$ weights responses by the reference histories visited by a specified policy. Lower calibration loss alone therefore does not guarantee a smaller $\delta_t$ for a later policy. The bounds provide worst-case structural sensitivity; we do not estimate their divergences or use them to quantify the observed score gains. Appendix A.8 connects simulator error to policy optimization under an explicit near-optimality condition, which is not established here for GRPO.

## 4 Calibrated Interactive RL

**Calibrated Interactive RL** couples a learned user simulator with interactive policy optimization. As shown in Figure 1, Phase I fits the simulator's conditional responses to multi-turn interaction data. Phase II freezes this simulator and trains the dialogue policy with GRPO on complete conversations. The two phases address the simulator and trajectory-collection components identified in Section 3. We denote policy parameters by $\theta$ and simulator parameters by $\phi$.

### 4.1 Phase I: Simulator Calibration via SFT

**Calibration data.** As online interaction data from real users were unavailable for this study, we use LLM-generated dialogues as a substitute source of calibration data. We collect multi-turn traces by having Qwen3-235B-A22B-Instruct act as a user interacting with the base Gemma-3-4B-IT policy. Each task consists of an input $x$ and a reference answer or document $y^*$. The teacher receives this task context and the dialogue history, and generates user messages according to a shared interaction prompt. The resulting dataset is $\mathcal{D}_{\text{multi}} = \{(x, y^*, \tau_{\text{teacher}})\}$.

**Behavioral-cloning objective.** We fine-tune a Qwen2.5-7B-Instruct simulator $\pi_\phi^{\text{sim}}$ to predict teacher user turns $u_t$ from the preceding dialogue $h_t$, assistant action $a_t$, and task context:

$$\mathcal{L}_{\text{SFT}}(\phi) = -\mathbb{E}_{(x,y^*,\tau)\sim\mathcal{D}_{\text{multi}}} \left[ \sum_{t \in \mathcal{U}(\tau)} \log \pi_\phi^{\text{sim}}(u_t \mid x, y^*, h_t, a_t) \right], \tag{5}$$

where $\mathcal{U}(\tau)$ indexes user turns. For the initial message, define $u_{-1} = u_{\text{init}}$ and $h_{-1} = a_{-1} = \varnothing$. For $t \geq 0$, $h_t$ contains the conversation before $a_t$; the target user response $u_t$ follows that action. The loss is applied to user tokens, with assistant tokens masked. In this framework, calibration means fitting the simulator's conditional response distribution to the teacher demonstrations. Table 3 in Appendix B.1 summarizes the task-specific data and training budgets.

**Simulator input and response protocol.** The teacher and calibrated simulator receive the task description, reference goal, and dialogue history, while the policy interacts through the generated conversation. The shared prompt requests incomplete early demands, gradual information release, and occasional mistakes. Each user reply continues the conversation in response to the assistant's latest action. Together with history concatenation and stopping, this behavior defines the user kernel $P_{\text{sim},t}$ in Section 3. Appendix C gives the full template.

**Task goals and user feedback.** Conditioning on $y^*$ gives the simulator a reference for its task-related feedback. The policy uses the generated messages to produce a solution or document, which is scored against the same target. User-turn supervision covers both the initial presentation of the task and the subsequent disclosure of requirements through feedback.

**Algorithm 1** Calibrated Interactive RL

**Input:** teacher traces $\mathcal{D}_{\text{multi}}$, dataset $\mathcal{D}$ of task pairs $(x, y^*)$, group size $G$, horizon $H$
Initialize simulator $\pi_\phi^{\text{sim}}$ and policy $\pi_\theta$
Fit $\phi$ on teacher user turns using $\mathcal{L}_{\text{SFT}}$; freeze $\phi^*$
**for** each policy training step **do**
    Sample task inputs $(x, y^*)$ from $\mathcal{D}$
    **for** each of $G$ rollouts per input **do**
        Sample $u_{\text{init}} \sim \pi_{\phi^*}^{\text{sim}}(\cdot \mid x, y^*, \varnothing, \varnothing)$
        Initialize policy-visible history $h_0 = (u_{\text{init}})$
        **for** $t = 0, \ldots, H-1$ **do**
            Agent: $a_t \sim \pi_\theta(\cdot \mid h_t)$
            Record $a_t$; stop if the total dialogue message limit is reached
            Simulator: $u_t \sim \pi_{\phi^*}^{\text{sim}}(\cdot \mid x, y^*, h_t, a_t)$
            Set $h_{t+1} = (h_t, a_t, u_t)$; stop if the simulator terminates
        **end for**
        Score the final assistant output: $R(\tau) = \text{Score}(\hat{y}_\tau, y^*)$
    **end for**
    Update $\theta$ with GRPO using group-normalized terminal rewards
**end for**

### 4.2 Phase II: Interactive Policy Optimization

After SFT, we freeze the fitted simulator parameters $\phi^*$. The policy $\pi_\theta$ generates complete dialogues in this calibrated environment. The simulator produces the initial user message, after which assistant and simulator responses alternate. Each sampled assistant action becomes part of the history used to generate the next user reply. Policy updates therefore use trajectories generated by the rollout policy, including the continuations elicited by its own earlier responses.

**Episode construction and stopping.** For each input, we sample $G = 32$ trajectories with at most 14 messages, corresponding to seven user–assistant rounds and at most $H = 7$ assistant actions. The initial user request counts toward this limit. The simulator is prompted to emit `[[TERMINATE CHAT]]` when it considers the goal solved or determines that the assistant cannot help further. An episode also ends when the message budget is reached. In either case, the reward scores the final assistant output $\hat{y}_\tau$ of dialogue $\tau$ against $y^*$. The base and trained policies use the same proactive assistant instruction during evaluation, and this instruction is also used during policy RL. It encourages clarification when details are missing and structured, concise assistance.

**Terminal task reward and GRPO.** Each trajectory receives one outcome score, $R(\tau) = \text{Score}(\hat{y}_\tau, y^*)$, at episode completion. MediumDocEdit-Chat uses BLEU against the reference document; MATH-Chat uses a binary Qwen3-235B judge score against the reference answer. These scores evaluate the final output after the sequence of requests, responses, and revisions. For a group with reward mean $\mu_G$ and standard deviation $\sigma_G > 0$, GRPO (Shao et al., 2024) uses the trajectory advantage $\hat{A}(\tau) = (R(\tau) - \mu_G)/\sigma_G$. Comparing trajectories for a shared task provides a relative learning signal for policy updates. The configuration uses PPO-style clipping (Schulman et al., 2017) with lower and upper offsets 0.2 and 0.28, and sets the KL coefficient to 0.0. Appendix D provides the full optimizer and rollout settings.

## 5 Experiments

We compare fixed-history training, interaction with a prompted simulator, and calibrated interaction on document editing and mathematical problem solving, measuring task performance and assistant token use.

### 5.1 EXPERIMENTAL SETUP

**Tasks and Models.** **MediumDocEdit-Chat** evaluates iterative document editing against a target document; **MATH-Chat** adapts hard MATH problems (Hendrycks et al., 2021) to conversational problem solving. We follow the multi-turn evaluation setup of COLLABLLM (Wu et al., 2025). The policy backbone is **Gemma-3-4B-IT** (Gemma Team et al., 2025), and the training-time user simulator is **Qwen2.5-7B-Instruct** (Yang et al., 2024). Each interaction begins with a simulator-generated request. The simulator uses the task description and reference target to provide subsequent feedback, while the policy builds its response from the conversation history.

**Data Construction and Budgets.** Phase I trains the simulator on user turns in teacher-generated dialogues. Phase II samples task inputs and generates new dialogues with the frozen simulator. For math, each task pairs a MATH-Hard problem with its reference solution. For editing, each task pairs an initial writing or editing request with a refined reference document; the simulator supplies partial instructions and feedback on proposed edits. Table 3 in Appendix B.1 distinguishes calibration traces, RL task inputs, and held-out evaluation examples. Simulator SFT uses the sequence limits listed in the table, while policy rollouts follow the message budget described below.

**Comparisons.** We evaluate the instruction-tuned **base policy** and a **COLLABLLM reproduction** using offline DPO on the same policy backbone. **Static Context** applies GRPO to fixed histories collected with the Qwen3 teacher. **Naive Interactive** generates on-policy rollouts with prompted Qwen2.5-7B, without calibration to teacher trajectories; **Calibrated Interactive** uses the same simulator backbone after teacher-based SFT. This pair tests calibration within interactive training. Static Context and Naive Interactive differ in both history generation and user model, making theirs a configuration-level comparison. A **large-model reference** row provides an additional comparison involving a ground-truth-conditioned Qwen3-235B simulator.

**Training.** Phase I fine-tunes Qwen2.5-7B-Instruct on Qwen3-235B teacher traces (Yang et al., 2025); Table 3 lists the task-specific budgets. Phase II trains the Gemma policy with GRPO, using $G = 32$ rollouts per input and a learning rate of $5 \times 10^{-6}$. Policy sampling temperatures are 1.0 for training and 0.8 for evaluation. Appendix D provides the optimizer, hardware, and rollout configurations.

**Evaluation and Metrics.** Evaluation uses 200 examples per task and at most 14 messages (7 user–assistant rounds). Qwen3-235B-A22B-Instruct (FP8) serves as the evaluation user simulator and math accuracy judge, as well as the source of calibration targets. Editing is scored with word-level sentence BLEU-4 against the target document; math uses a binary judgment of final-answer correctness. Both task metrics are reported on a 0–100 scale. **#Toks** measures the total assistant tokens across a dialogue, in thousands. We report means and standard deviations over 3 runs, and summarize the two tasks with the arithmetic means of their scores and token counts. The per-task columns provide the detailed quality–length comparisons.

### 5.2 EXPERIMENTAL RESULTS

Calibrated Interactive achieves the highest mean scores on both benchmarks among the non-reference configurations in Table 2. Relative to Naive Interactive, calibration improves task scores while reducing assistant token use on both tasks.

The broader comparison with Static Context shows a task-dependent quality–length tradeoff: Calibrated Interactive improves editing quality with fewer tokens, while its higher mathematical accuracy accompanies longer assistant outputs. These differences reflect the full training configurations.

The large-model reference has higher editing quality and shorter outputs; Calibrated Interactive has a slightly higher mean math score. Figure 2 presents selected MATH-Chat examples, with dialogue records in Appendix E.

Table 2: **Task performance and assistant token use.** Gemma-3-4B-IT results under Qwen3-235B-simulated evaluation, reported as mean$_{\pm \text{std}}$ over 3 runs. Scores use a 0–100 scale; token counts are in thousands. The final columns average the two tasks. Blue highlights Calibrated Interactive; bold compares the non-reference rows.

| Method | MediumDocEdit-Chat | | MATH-Chat | | Average | |
|---|---|---|---|---|---|---|
| | BLEU ↑ | #Toks $(k)$ ↓ | ACC ↑ | #Toks $(k)$ ↓ | Score ↑ | #Toks $(k)$ ↓ |
| ***Baselines*** | | | | | | |
| Gemma-3-4B-IT | 32.2±1.5 | 2.47±0.12 | 82.3±0.6 | 1.76±0.04 | 57.3 | 2.12 |
| CollabLLM† | 33.1±0.4 | 2.38±0.07 | 82.3±1.9 | 1.63±0.09 | 57.7 | **2.01** |
| ***Training variants (Gemma-3-4B-IT)*** | | | | | | |
| Static Context | 33.8±1.8 | 3.09±0.01 | 85.0±0.0 | **1.63±0.07** | 59.4 | 2.36 |
| Naive Interactive | 26.1±1.2 | 4.00±0.07 | 89.3±3.4 | 1.97±0.05 | 57.7 | 2.98 |
| **Calibrated Interactive** | **34.6±1.0** | **2.27±0.08** | **91.5±1.3** | 1.86±0.09 | **63.05** | 2.06 |
| ***Reference*** | | | | | | |
| Large-model reference | 36.2±1.4 | 1.83±0.06 | 89.7±1.6 | 1.27±0.04 | 63.0 | 1.55 |

†CollabLLM (Wu et al., 2025) reproduced with offline DPO on the same Gemma-3-4B-IT backbone.

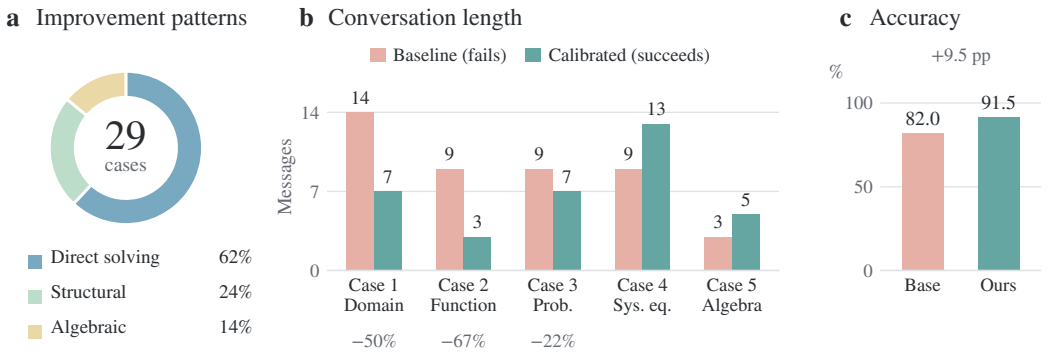


Figure 2: **Selected MATH-Chat improvements.** (a) Categories for 29 cases where the baseline fails and the calibrated policy succeeds. (b) Conversation lengths in messages; the plotted percentage reductions compare lengths of failed baseline and successful policy dialogues. (c) Displayed accuracies of 82.0% and 91.5% differ by 9.5 percentage points. Table 2 gives the three-run means and standard deviations, including a base accuracy of 82.3%.

## 6 Conclusion

We presented Calibrated Interactive RL, which fits a user simulator to multi-turn interaction data and freezes it for policy optimization. This design combines calibrated user feedback with training conversations shaped by the policy's own responses. Our analysis separates policy-induced and simulator-induced shift and characterizes how each affects dialogue returns, with distinct bounds for cumulative and terminal rewards. On MediumDocEdit-Chat and MATH-Chat, calibration improves task performance while reducing assistant token use relative to uncalibrated interaction. These results support treating the user simulator as a learned component of multi-turn policy training. Extending calibration to broader sources of interaction data and updating the simulator as policies evolve are natural directions for future work.

## AI USE STATEMENT

LLMs generate synthetic calibration trajectories, simulate users, and judge mathematical accuracy, as described in Sections 4–5. An AI assistant supported manuscript revision, reference checking, mathematical derivations and proof review, interpretation of results, figure preparation, and LaTeX formatting. The authors are responsible for the final content and its verification.

## REFERENCES

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, et al. MT-Bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. *arXiv preprint arXiv:2402.14762*, 2024.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in Neural Information Processing Systems*, 28, 2015.

Thomas Carta, Clément Romac, Thomas Wolf, Sylvain Lamprier, Olivier Sigaud, and Pierre-Yves Oudeyer. Grounding large language models in interactive environments with online reinforcement learning. In *International Conference on Machine Learning*, pp. 3676–3713. PMLR, 2023.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *arXiv preprint arXiv:2401.01335*, 2024.

Yann Dubois, Chen Xuechen Li, Rohan Taori, Tianyi Zhang, Ishaan Gulrajani, Jimmy Ba, Carlos Guestrin, Percy S Liang, and Tatsunori B Hashimoto. AlpacaFarm: A simulation framework for methods that learn from human feedback. *Advances in Neural Information Processing Systems*, 36:30039–30069, 2023.

Kanishk Gandhi, Agam Bhatia, and Noah D Goodman. Learning to simulate human dialogue. *arXiv preprint arXiv:2601.04436*, 2026.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

Joey Hong, Sergey Levine, and Anca Dragan. Zero-shot goal-directed dialogue via RL on imagined conversations. *arXiv preprint arXiv:2311.05584*, 2023.

Zhiyuan Hu, Yue Feng, Anh Tuan Luu, Bryan Hooi, and Aldo Lipani. Unlocking the potential of user feedback: Leveraging large language model as user simulators to enhance dialogue system. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 3953–3957, 2023.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. LLMs get lost in multi-turn conversation. *arXiv preprint arXiv:2505.06120*, 2025.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, et al. RLAIF vs. RLHF: Scaling reinforcement learning from human feedback with AI feedback. *arXiv preprint arXiv:2309.00267*, 2023.

Sergey Levine, Aviral Kumar, George Tucker, and Justin Fu. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*, 2015.

Stephane Ross and Drew Bagnell. Efficient reductions for imitation learning. In Yee Whye Teh and Mike Titterington (eds.), *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pp. 661–668, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/ross10a.html.

Stéphane Ross, Geoffrey Gordon, and Drew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 627–635. JMLR Workshop and Conference Proceedings, 2011.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Murray Shanahan, Kyle McDonell, and Laria Reynolds. Role play with large language models. *Nature*, 623(7987):493–498, 2023.

Lior Shani, Aviv Rosenberg, Asaf Cassel, Oran Lang, Daniele Calandriello, Avital Zipori, Hila Noga, Orgad Keller, Bilal Piot, Idan Szpektor, et al. Multi-turn reinforcement learning with preference human feedback. *Advances in Neural Information Processing Systems*, 37:118953–118993, 2024.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. Character-LLM: A trainable agent for role-playing. *arXiv preprint arXiv:2310.10158*, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R Johnston, et al. Towards understanding sycophancy in language models. *arXiv preprint arXiv:2310.13548*, 2023.

Weiyan Shi, Kun Qian, Xuewei Wang, and Zhou Yu. How to build user simulators to train RL-based dialog systems. *arXiv preprint arXiv:1909.01388*, 2019.

Wentao Shi, Mengqi Yuan, Junkang Wu, Qifan Wang, and Fuli Feng. Direct multi-turn preference optimization for language agents. *arXiv preprint arXiv:2406.14868*, 2024.

Charlie Snell, Ilya Kostrikov, Yi Su, Mengjiao Yang, and Sergey Levine. Offline RL for natural language generation with implicit language Q learning. *arXiv preprint arXiv:2206.11871*, 2022.

Bo-Hsiang Tseng, Yinpei Dai, Florian Kreyssig, and Bill Byrne. Transferable dialogue systems and user simulators. *arXiv preprint arXiv:2107.11904*, 2021.

Siddharth Verma, Justin Fu, Sherry Yang, and Sergey Levine. CHAI: A chatbot AI for task-oriented dialogue with offline reinforcement learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 4471–4491, 2022.

Xiaohua Wang, Muzhao Tian, Yuqi Zeng, Zisu Huang, Jiakang Yuan, Bowen Chen, Jingwen Xu, Mingbo Zhou, Wenhao Liu, Muling Wu, Zhengkang Guo, Qi Qian, Yifei Wang, Feiran Zhang, Ruicheng Yin, Shihan Dou, Changze Lv, Tao Chen, Kaitao Song, Xu Tan, Tao Gui, Xiaoqing Zheng, and Xuanjing Huang. Reward hacking in the era of large models: Mechanisms, emergent misalignment, challenges. *arXiv preprint arXiv:2604.13602*, 2026.

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.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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*, 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.

# A Theoretical Proofs and Derivations

We derive the policy and simulator sensitivity bounds in Section 3. The argument tracks the distributions of complete histories, so user responses may depend on the entire preceding dialogue.

## A.1 Notation, Tasks, and Stopping

Fix a task $z = (x, y^*)$. All initial distributions, user kernels, rewards, and divergences below are conditional on this task. The policy receives the visible conversation $h$, and acts according to $\pi(\cdot \mid h)$; it need not observe the private reference $y^*$. For multiple tasks drawn from a common distribution $\rho$, first apply each result conditionally on $z$, then average its right-hand side over $\rho$. In particular,

$$|\mathbb{E}_{z\sim\rho} J_{\mathrm{ref},z}(\pi) - \mathbb{E}_{z\sim\rho} J_{\mathrm{sim},z}(\pi)| \le \mathbb{E}_{z\sim\rho} |J_{\mathrm{ref},z}(\pi) - J_{\mathrm{sim},z}(\pi)|.$$

The divergences in this expectation are task-conditional distances, rather than distances between histories after marginalizing out the task. This distinction preserves a common reward or judging rule even when a user model changes how much information its responses reveal about the target.

Let $H \ge 1$ and let the history and action spaces be countable. Policies and kernels are defined on all relevant histories, including those outside a particular rollout distribution's support. At step $t$, the policy chooses $a_t$ and the user kernel $P_t(h' \mid h, a)$ appends the action, the user response when one is generated, and the stopping status. Time dependence accommodates the hard dialogue limit. Define

$$q_t^\pi(h, a) = d_t^\pi(h)\pi(a \mid h), \qquad (\mathcal{T}_t q)(h') = \sum_{h,a} P_t(h' \mid h, a) q(h, a).$$

Thus $d_{t+1}^\pi = \mathcal{T}_t q_t^\pi$.

To represent early stopping, retain the entire stopped transcript and pad with dummy symbols through time $H$. After stopping, both policies choose the same dummy action and both environments use the same deterministic padding transition. These added symbols are mathematical bookkeeping and do not count toward the dialogue message budget. Cumulative rewards after stopping are zero. A terminal score is computed from the stopped transcript and is unchanged by padding. Distinct stopped transcripts remain distinct, so the complete padded history retains every input needed to compute the score.

At the hard message limit, recording the final assistant answer and stopping is itself a shared deterministic transition; no additional user message is generated. In the reported protocol, $H = 7$ assistant actions and the initial user message together with the subsequent dialogue are subject to the 14-message limit. Consequently the final transition at the cap contributes $\delta_{H-1} = 0$. The general terminal-score bound also covers protocols that generate a final user response, in which case this term can be nonzero.

We use $\|p - q\|_1 = \sum_v |p(v) - q(v)|$, twice total variation distance. Cumulative rewards satisfy $|r_t(h, a)| \le R_{\max}$ and are evaluated before the next user transition:

$$J(\pi) = \mathbb{E}\left[\sum_{t=0}^{H-1} r_t(h_t, a_t)\right].$$

The compared processes share this task-conditional reward function. The policy comparison also shares the initial distribution and user kernels; the simulator comparison uses a fixed policy and can have different initial distributions.

## A.2 Auxiliary Lemmas

**Lemma A.1** (Joint distribution decomposition)**.** *For joint distributions $p(x, y) = p(x)p(y \mid x)$ and $q(x, y) = q(x)q(y \mid x)$, with conditional distributions specified also at zero-probability $x$,*

$$\|p - q\|_1 \le \|p(x) - q(x)\|_1 + \mathbb{E}_{x\sim q}\|p(\cdot \mid x) - q(\cdot \mid x)\|_1. \tag{6}$$

*Proof.* Add and subtract $q(x)p(y \mid x)$, and use normalization of the conditional distributions:

$$\begin{aligned}\|p - q\|_1 &\le \sum_{x,y} |p(x) - q(x)|p(y \mid x) + \sum_{x,y} q(x)|p(y \mid x) - q(y \mid x)| \\ &= \|p(x) - q(x)\|_1 + \mathbb{E}_{x\sim q}\|p(\cdot \mid x) - q(\cdot \mid x)\|_1.\end{aligned}$$

□

**Lemma A.2** (Error propagation dynamics)**.** *For policies $\pi, \pi'$ sharing a user environment and initial distribution, define $\Delta_t = \|d_t^{\pi} - d_t^{\pi'}\|_1$ and $\varepsilon_t = \mathbb{E}_{h\sim d_t^{\pi'}}\|\pi(\cdot \mid h) - \pi'(\cdot \mid h)\|_1$. Then, for $t = 0, \ldots, H-1$,*

$$\Delta_{t+1} \le \Delta_t + \varepsilon_t, \qquad \Delta_t \le \sum_{i=0}^{t-1} \varepsilon_i. \tag{7}$$

*Proof.* For any summable signed measure $v$, stochasticity gives

$$\|\mathcal{T}_t v\|_1 = \sum_{h'} \left| \sum_{h,a} P_t(h' \mid h, a) v(h, a) \right| \le \sum_{h,a} |v(h, a)| \sum_{h'} P_t(h' \mid h, a) = \|v\|_1.$$

Consequently, $\Delta_{t+1} \le \|q_t^{\pi} - q_t^{\pi'}\|_1$. Lemma A.1 bounds the latter by $\Delta_t + \varepsilon_t$. The shared initial distribution gives $\Delta_0 = 0$; induction proves the second inequality. □

### A.3 Proof of Theorem 3.1

*Proof.* For the shared bounded reward,

$$\begin{aligned} |J(\pi) - J(\pi')| &\le \sum_{t=0}^{H-1} \left| \sum_{h,a} (q_t^{\pi}(h, a) - q_t^{\pi'}(h, a)) r_t(h, a) \right| \\ &\le R_{\max} \sum_{t=0}^{H-1} \|q_t^{\pi} - q_t^{\pi'}\|_1 \le R_{\max} \sum_{t=0}^{H-1} (\Delta_t + \varepsilon_t). \end{aligned}$$

By Lemma A.2 and a change of summation order,

$$\sum_{t=0}^{H-1} (\Delta_t + \varepsilon_t) \le \sum_{t=0}^{H-1} \sum_{i=0}^{t} \varepsilon_i = \sum_{i=0}^{H-1} (H - i) \varepsilon_i.$$

This proves the claimed return sensitivity under a policy replacement. Each $\varepsilon_i$ appears once for the immediate reward and once for every later reward. □

### A.4 Proof of Theorem 3.2

*Proof.* Fix $\pi$. Write $d_t^{\mathrm{ref}}, d_t^{\mathrm{sim}}$ for its history distributions under the two user models, and $\Delta_t^{\mathrm{env}} = \|d_t^{\mathrm{ref}} - d_t^{\mathrm{sim}}\|_1$. The shared policy gives

$$\|q_t^{\mathrm{ref}} - q_t^{\mathrm{sim}}\|_1 = \sum_{h} |d_t^{\mathrm{ref}}(h) - d_t^{\mathrm{sim}}(h)| \sum_{a} \pi(a \mid h) = \Delta_t^{\mathrm{env}}.$$

Add and subtract $\mathcal{T}_{\mathrm{sim},t} q_t^{\mathrm{ref}}$:

$$\begin{aligned} \Delta_{t+1}^{\mathrm{env}} &\le \|(\mathcal{T}_{\mathrm{ref},t} - \mathcal{T}_{\mathrm{sim},t}) q_t^{\mathrm{ref}}\|_1 + \|\mathcal{T}_{\mathrm{sim},t}(q_t^{\mathrm{ref}} - q_t^{\mathrm{sim}})\|_1 \\ &\le \delta_t + \Delta_t^{\mathrm{env}}. \end{aligned}$$

For the first term, the triangle inequality yields

$$\begin{aligned} \|(\mathcal{T}_{\mathrm{ref},t} - \mathcal{T}_{\mathrm{sim},t}) q_t^{\mathrm{ref}}\|_1 &= \sum_{h'} \left| \sum_{h,a} q_t^{\mathrm{ref}}(h, a) [P_{\mathrm{ref},t}(h' \mid h, a) - P_{\mathrm{sim},t}(h' \mid h, a)] \right| \\ &\le \sum_{h,a} q_t^{\mathrm{ref}}(h, a) \|P_{\mathrm{ref},t}(\cdot \mid h, a) - P_{\mathrm{sim},t}(\cdot \mid h, a)\|_1 = \delta_t. \end{aligned}$$

The second term uses non-expansiveness. Starting from $\Delta_0^{\mathrm{env}} = \|\mu_{\mathrm{ref}} - \mu_{\mathrm{sim}}\|_1$, induction gives

$$\Delta_t^{\mathrm{env}} \le \Delta_0^{\mathrm{env}} + \sum_{i=0}^{t-1} \delta_i.$$

The common reward is evaluated before the next user transition, so

$$
\begin{aligned}
|J_{\mathrm{ref}}(\pi) - J_{\mathrm{sim}}(\pi)| &\le R_{\max} \sum_{t=0}^{H-1} \|q_t^{\mathrm{ref}} - q_t^{\mathrm{sim}}\|_1 \\
&\le R_{\max} \left[ H\Delta_0^{\mathrm{env}} + \sum_{t=0}^{H-1} \sum_{i=0}^{t-1} \delta_i \right] \\
&= R_{\max} \left[ H\Delta_0^{\mathrm{env}} + \sum_{i=0}^{H-2} (H-1-i)\delta_i \right].
\end{aligned}
$$

Transition $H-1$ occurs after the final reward in this convention, so its coefficient is zero. Empty sums are zero, including when $H = 1$. □

### A.5 Terminal Scores and Stopping

Let $\tau$ denote the entire padded trajectory and $Q_H$ its distribution. The padded history preserves every real action, user response, and stopping decision, so a bounded trajectory score $F(\tau)$ is a function of this history. For any two trajectory distributions $Q, Q'$ and a common score with $|F| \le R_{\max}$,

$$
|\mathbb{E}_Q F - \mathbb{E}_{Q'} F| = \left| \sum_{\tau} [Q(\tau) - Q'(\tau)] F(\tau) \right| \le R_{\max} \|Q - Q'\|_1.
$$

The policy recurrence at time $H$ and the simulator recurrence with its initial term therefore give

$$
|\mathbb{E}_\pi F - \mathbb{E}_{\pi'} F| \le R_{\max} \|Q_H^\pi - Q_H^{\pi'}\|_1 \le R_{\max} \sum_{i=0}^{H-1} \varepsilon_i,
$$

$$
|\mathbb{E}_{\mathrm{ref}} F - \mathbb{E}_{\mathrm{sim}} F| \le R_{\max} \|Q_H^{\mathrm{ref}} - Q_H^{\mathrm{sim}}\|_1 \le R_{\max} \left[ \Delta_0^{\mathrm{env}} + \sum_{i=0}^{H-1} \delta_i \right].
$$

Here $\Delta_0 = 0$ in the policy comparison. A terminal score can depend on the stopping decision following an assistant answer, which is why the general simulator sum includes $\delta_{H-1}$. In the reported protocol, the shared deterministic transition at the hard message cap sets this final term to zero; earlier simulator decisions to terminate remain part of their respective kernels.

For stochastic scoring, suppose the conditional score distribution given $(z, \tau)$ is the same in both compared processes, and set $F_z(\tau) = \mathbb{E}[R \mid z, \tau]$. The same result follows by iterated expectation. This requires a shared conditional judging rule, not shared random draws. If the judging rules differ, their difference must be included as an additional score-mismatch term.

Section 3.4 gives the reward-range caps and the centering argument for nonnegative scores. These apply to the same trajectory distributions used in this proof.

### A.6 Interpreting the Two Comparisons

The policy bound holds under fixed user dynamics, whereas the simulator bound holds for a fixed policy. If both change, they can be compared through an intermediate policy–environment pair using the triangle inequality. A static-context optimization procedure additionally specifies an objective and a data-coverage assumption; the policy bound concerns the returns of two policies rather than that procedure's optimization error.

SFT uses user responses and their preceding contexts from the calibration data. The simulator divergence $\delta_t$, in contrast, averages over histories and actions generated by the policy being evaluated in the reference user environment. These weighting distributions coincide only under additional conditions on data collection and the policy. Consequently, likelihood fitting on the collected data and return sensitivity under a later policy describe distinct quantities. This distinction motivates evaluating the combined calibration and interactive-training pipeline through its task outcomes.

### A.7 Magnitude and Interpretation of the Terminal Bound

The distances above compare distributions over complete messages and histories. They do not measure semantic similarity: two deterministic user responses with different wording have $\ell_1$ distance 2, even if they express the same meaning. Together with accumulation over turns, this can make a valid bound uninformative about the size of a score difference. We do not estimate $\varepsilon_t$, $\delta_t$, or $\Delta_0^{\mathrm{env}}$ in the reported experiments; the results are therefore a worst-case structural analysis of distribution shift rather than a quantitative explanation of the measured gains.

For a fixed task $z$ and a common terminal score $F_z \in [0,1]$, the centering and range arguments give the explicit bound

$$|J_{\mathrm{ref},z}(\pi) - J_{\mathrm{sim},z}(\pi)| \le B_z(\pi) := \min\left\{1, \frac{\Delta_{0,z}^{\mathrm{env}} + \sum_{t=0}^{H-1} \delta_{t,z}(\pi)}{2}\right\}. \tag{8}$$

Here $J_{\mathrm{ref},z}$ and $J_{\mathrm{sim},z}$ denote expected terminal scores, and the subscripts $z$ and argument $\pi$ make explicit the task and policy dependence previously suppressed. This bound improves on the maximum possible score difference of 1 only when the divergence sum, including the initial term, is less than 2. For example, with a shared initial request distribution, $H = 7$, a shared final transition, and $\delta_{t,z}(\pi) \le \delta$ for the six remaining transitions, it reduces to $\min\{1, 3\delta\}$. These are illustrative assumptions, not estimates from our experiments. For a common task distribution $\rho$, the task-averaged return gap is bounded by $B(\pi) = \mathbb{E}_{z\sim\rho} B_z(\pi)$.

### A.8 A Conditional Connection to Policy Optimization

A return-sensitivity bound can yield an optimization statement when combined with an explicit assumption on the simulator-trained policy. The following result uses a common policy class $\Pi$ and expected returns evaluated under the same task distribution and scoring rule.

**Corollary A.3** (Conditional policy transfer)**.** *Suppose* $|J_{\mathrm{ref}}(\pi) - J_{\mathrm{sim}}(\pi)| \le B(\pi)$ *for every* $\pi \in \Pi$*, where* $B(\pi) \ge 0$*. Assume a reference-optimal policy* $\pi^* \in \arg\max_{\pi\in\Pi} J_{\mathrm{ref}}(\pi)$ *exists. Let* $\hat{\pi} \in \Pi$ *satisfy*

$$J_{\mathrm{sim}}(\hat{\pi}) \ge \sup_{\pi\in\Pi} J_{\mathrm{sim}}(\pi) - \eta, \qquad \eta \ge 0.$$

*Then*

$$J_{\mathrm{ref}}(\pi^*) - J_{\mathrm{ref}}(\hat{\pi}) \le B(\pi^*) + B(\hat{\pi}) + \eta. \tag{9}$$

*Proof.* Insert the simulator returns for the two policies:

$$\begin{aligned} J_{\mathrm{ref}}(\pi^*) - J_{\mathrm{ref}}(\hat{\pi}) &= [J_{\mathrm{ref}}(\pi^*) - J_{\mathrm{sim}}(\pi^*)] \\ &\quad + [J_{\mathrm{sim}}(\pi^*) - J_{\mathrm{sim}}(\hat{\pi})] + [J_{\mathrm{sim}}(\hat{\pi}) - J_{\mathrm{ref}}(\hat{\pi})] \\ &\le B(\pi^*) + \eta + B(\hat{\pi}). \end{aligned}$$

The first and third terms use the assumed simulator-error bounds. The middle term is at most $\eta$ because $\pi^* \in \Pi$ and $\hat{\pi}$ is $\eta$-optimal in that same class. □

For normalized terminal rewards, Equation (8) supplies a valid $B$ after averaging over tasks. The conclusion requires control of simulator error for both the reference-optimal policy and the learned policy, as well as the simulator optimization error $\eta$. Calibration likelihood on collected contexts does not by itself provide this control, and the corollary does not assert that GRPO attains the stated near-optimality condition. It specifies the additional conditions needed to turn return sensitivity into a policy-performance guarantee.

## B Experimental Details

In this section, we provide a comprehensive description of the datasets, model configurations, training hyperparameters, and computational infrastructure used in our experiments. Our implementation builds on the SLIME distributed RL framework and the CollabLLM evaluation pipeline.

### B.1 Datasets and Benchmarks

We evaluate on two multi-turn task benchmarks using synthetic user interactions. Table 3 reports the RL task counts, calibration trace counts, and held-out evaluation sizes.

Table 3: Data and training budgets. Phase I fits the user simulator to calibration traces; Phase II trains the dialogue policy on rollouts from task inputs.

| Setting | MATH-Chat | MediumDocEdit-Chat |
|---|---|---|
| Calibration traces (Phase I) | 1,860 | 16,028 |
| SFT maximum sequence length | 4,096 | 6,144 |
| SFT epochs | 2 | 3 |
| SFT learning rate | $1 \times 10^{-5}$ | $2 \times 10^{-5}$ |
| RL task inputs (Phase II) | $\sim$3,000 | $\sim$38,118 |
| GRPO training steps | 10 | 20 |
| Held-out evaluation examples | 200 | 200 |

**MATH-Chat.** This benchmark evaluates final-answer accuracy in multi-turn mathematical dialogues.

- **Data Source:** We adapt problems from the MATH dataset (Hendrycks et al., 2021), specifically selecting *Hard*-level problems that require multi-step reasoning.
- **Task Setup:** For each math problem, the ground-truth solution serves as $y^*$. The user simulator is prompted to play a student who may express confusion, make logical errors, or ask for clarification. The dialogue develops through the simulator's questions and the assistant's proposed solutions.
- **Reward:** Binary accuracy—an LLM judge (Qwen3-235B) determines whether the agent's final answer matches the ground truth ($R \in \{0, 1\}$).

**MediumDocEdit-Chat.** This dataset focuses on collaborative document editing, requiring the agent to iteratively refine a document based on user feedback.

- **Data Source:** Derived from article writing and editing corpora. Each sample consists of an initial user request $x$ and a ground-truth refined document $y^*$.
- **Task Setup:** The simulator is prompted to provide incomplete editing instructions and respond to proposed edits.
- **Reward:** Sentence-level BLEU score between the agent's final produced document and the ground-truth reference $y^*$ ($R \in [0, 1]$).

### B.2 Model Architectures

**Policy Agent ($\pi_\theta$).** We use **Gemma-3-4B-IT** (Gemma Team et al., 2025) as the backbone for the dialogue policy across all experiments. The 4B parameter scale supports rollout generation within the available compute budget. The model is loaded in `bfloat16` precision with Flash Attention 2 enabled.

**User Simulator ($\pi_\phi^{\text{sim}}$).** We use **Qwen2.5-7B-Instruct** (Yang et al., 2024) as the simulator backbone. After Phase I SFT calibration, the simulator is frozen for Phase II training.

**Evaluation Simulator & Judge.** During evaluation, we employ **Qwen3-235B-A22B-Instruct** (FP8 quantized) as the user simulator and math accuracy judge. Document editing is scored with BLEU against the reference document. Qwen3-235B also generates the calibration traces, while Qwen2.5-7B supplies the user responses during policy training.

### B.3 Training Phase I: Simulator Calibration (SFT)

In the first phase, we calibrate the user simulator via Supervised Fine-Tuning (SFT) on multi-turn traces generated by the Qwen3-235B teacher interacting with the base policy. The training targets are the teacher's user responses at successive points in the conversation.

**Data Collection.** Section 4 describes teacher–base-policy trace collection and user-turn supervision; Table 3 gives the resulting dataset sizes and SFT sequence limits. The full teacher prompt is reproduced in Appendix C.

**Training Setup.** We fine-tune Qwen2.5-7B-Instruct using standard causal language modeling loss on the user turns only (masking assistant turns). Training uses FSDP on $8\times$ NVIDIA A800-80GB GPUs. Hyperparameters are detailed in Appendix D, Table 4.

### B.4 Training Phase II: Interactive Policy Optimization (GRPO)

In the second phase, the SFT-calibrated simulator is frozen, and we optimize the policy agent $\pi_\theta$ via Group Relative Policy Optimization (GRPO) (Shao et al., 2024). GRPO eliminates the need for a separate value network by normalizing rewards within a group of sampled trajectories, making it memory-efficient for long-context multi-turn training.

**Interaction Protocol.** Section 4.2 specifies grouped rollouts, the 14-message budget, and the simulator's stopping rule. Algorithm 1 gives the order of assistant actions, user responses, and terminal scoring.

**Reward Function.** We use sparse outcome rewards $R(\tau)$ administered only at episode completion:

- **MATH-Chat:** Binary accuracy (0/1) determined by an LLM judge comparing the final answer to ground truth.
- **MediumDocEdit-Chat:** BLEU score between the final document and the reference.

**Training Setup.** We train with Ray-based distributed FSDP on 32 GPUs (4 nodes $\times$ 8 A800-80GB) with SGLang for efficient rollout generation. The policy and rollout engine are co-located on the same GPUs (`--colocate` mode). Full hyperparameters are in Appendix D, Table 5.

### B.5 Baselines Implementation

**Gemma-3-4B-IT (Base Model).** The original instruction-tuned model without any multi-turn RL fine-tuning, serving as the zero-shot baseline. We evaluate it with the same proactive system prompt used during RL training.

**CollabLLM (Static RL).** We reproduce CollabLLM (Wu et al., 2025) using the official codebase, training on fixed offline histories with Multi-turn-aware Rewards (MR). We adapt it to the Gemma-3-4B-IT backbone used by our method; the reward model and training procedure follow the official implementation.

**Ours (Static Context).** A fixed-history training configuration that applies GRPO to conversation logs collected with the Qwen3 teacher simulator, conditioning on offline histories $h \sim d_t^{\text{data}}$. Naive Interactive instead collects histories with prompted Qwen2.5-7B. Their comparison therefore changes both the history-generation procedure and user model. It compares the complete configurations and does not isolate policy-induced shift under the common-user-environment assumption of Theorem 3.1.

**Ours (Naive Interactive).** This calibration ablation uses the full interactive RL loop with base Qwen2.5-7B-Instruct prompted to act as a user. It omits Phase I SFT on teacher trajectories. Calibrated Interactive uses the same simulator backbone after that SFT stage; their comparison tests the effect of fitting teacher responses before policy training.

### B.6 Computational Infrastructure

- **Hardware:** All experiments were conducted on NVIDIA A800-80GB GPUs. Phase I uses 8 GPUs; Phase II uses 32 GPUs (4 nodes).
- **Software:** Python 3.12, PyTorch 2.x, SLIME (distributed RL framework), SGLang (efficient LLM serving for rollout generation), and the CollabLLM evaluation pipeline.

- **Compute Cost:** Phase I (SFT) completes in <1 hour per task. Phase II (GRPO) trains for 10 steps (MATH) / 20 steps (MediumDocEdit), taking approximately 4–8 hours per task on 32 GPUs.
- **Networking:** NCCL communication with InfiniBand disabled (`NCCL_IB_DISABLE=1`); extended timeouts (72,000s) for long rollout steps.

### B.7 Evaluation Scope

The comparisons cover one policy backbone and two tasks under LLM-simulated interaction. Qwen3-235B supplies calibration traces, evaluation conversations, and math judgments, so teacher and evaluator biases may favor similar response patterns. Reference-conditioned user feedback can communicate correctness through corrections, acceptance, and stopping decisions. The large-model reference row provides a descriptive comparison; attributing its score differences to simulator choice would require matched policy checkpoints, prompts, decoding settings, and training seeds. Independent training runs would also quantify sensitivity to training randomness beyond the reported three-run summaries.

BLEU measures overlap with reference documents, and assistant-token counts measure evaluation output length. Total training cost additionally includes simulator inference and policy optimization. As the policy changes, a frozen simulator can encounter histories outside its calibration data. Evaluation with other simulator families, human collaborators, and more varied feedback would extend the present study.

## C Prompts

This section presents the exact prompts used in our framework. We include: (1) the **User Simulator** prompt used during both Phase I calibration and Phase II interactive rollouts, (2) the **Policy Agent** (proactive assistant) instruction used during RL training, and (3) the **Accuracy Judge** prompt used for reward computation. Template variables populated at runtime are highlighted in `{{blue}}`.

### C.1 User Simulator Prompt

The user simulator receives the task description and reference goal. The prompt requests user role-play, gradual information release, occasional mistakes, and low effort. We use this template for the Qwen3-235B teacher and the SFT-trained Qwen2.5-7B simulator. It specifies the initial request, subsequent replies, JSON fields, and termination signal.

**User Simulator Prompt**

You are role-playing as a human USER interacting with an AI collaborator to complete a specific task. Your goal is to generate realistic, natural responses that a user might give in this scenario.

**Input Information:**
You will be provided with:

- Task Description: The type of task you are trying to accomplish.
- Complete Prompt or Reference Goal: This field may include the complete user request/query or a reference answer. Use this to understand the user's intent and what would count as a satisfactory outcome.
- Chat History: The ongoing conversation between you (as the user) and the AI.

**Inputs:**

```
<|The Start of Task Description (Not visible to the AI)|>
{{task_desc}}
<|The End of Task Description|>
<|The Start of Complete Prompt or Reference Goal (Not visible to the
AI)|>
{{single_turn_prompt}}
<|The End of Complete Prompt or Reference Goal|>
<|The Start of Chat History|>
{{chat_history}}
<|The End of Chat History|>
```

**Guidelines:**

- **Stay in Character:** Role-play as a human USER. You are NOT an AI. Maintain a consistent personality throughout the chat.
- **Minimize Effort:** IMPORTANT! Avoid being too detailed. Provide vague or incomplete demands in early stages. Let the AI ask for clarification rather than providing everything upfront.
- **Knowledge Background:** Reflect the user's knowledge level. If less knowledgeable, you might not notice incorrect statements. Ask questions demonstrating your current understanding and areas of confusion.
- **Occasionally Make Mistakes:** Real users might misspell words, give wrong information, or ask unclear questions. Simulate this behavior.
- **Goal-Oriented:** Keep the chat focused on your intent. Avoid small talk or digressions.

**Output Format:**
Output a JSON object with three entries:

- `"current_answer"` (str): Briefly summarize the AI's current solution.
- `"thought"` (str): Your thought process deciding what to say next.
- `"response"` (str): Your response to the AI as the user you are role-playing.

**Important Notes:**

- If "Chat History" is empty, start the conversation from scratch with an initial request.
- Use the provided information for understanding context only. Avoid copying target queries directly.
- Use "`{{terminal_signal}}`" as your response when your goal has been solved or if you determine the AI cannot help further.

### C.2 Policy Agent Prompt (Proactive Assistant)

The policy model receives this system-level instruction that encourages interactive, collaborative behavior. This prompt is used during both RL training and evaluation to guide the agent toward asking clarifying questions and providing structured assistance.

**Policy Agent System Prompt**

You are an AI assistant interacting with a user to perform tasks such as writing, analysis, question answering, math, and coding. Your goal is to generate a response to the user's last message. You should be helpful, collaborative, and highly interactive.

**Input:**

```
<|The Start of Conversation History|>
{{chat_history}}
<|The End of Conversation History|>
<|The Start of Additional Information|>
{{additional_info}}
<|The End of Additional Information|>
```

**Guidelines:**

1. **Understanding & Engagement** — Accurately interpret the user's intent. Acknowledge previous interactions to maintain continuity.
2. **Interactivity (Important!)** — Ask clarifying questions if the request lacks detail. Ask *specific* follow-up questions (e.g., "Would you like more information on X?" not "Let me know if this is helpful."). Collaboratively offer guidance in complex situations. Focus on the user's long-term goals.
3. **Efficiency** — Be mindful of how much the user needs to read or type. When presenting options, provide multiple-choice suggestions to make it easier to respond quickly. Avoid repeating information unnecessarily.
4. **Communication Style** — Be honest. If unsure, say "I don't know" and suggest alternatives. Align tone with the user's emotional state.

**Output Format:**
Output a JSON object:

- `"current_problem"` (str): What is the user's current problem or confusion?
- `"thought"` (str): Your reasoning about what to say next.
- `"response"` (str): Your response following the guidelines above.

### C.3 Accuracy Judge Prompt (Reward Function)

During Phase II training, we use an LLM-based accuracy judge as the reward function for MATH-Chat. The judge evaluates whether the policy's final answer matches the ground truth. This prompt is sent to Qwen3-235B-Instruct.

**Accuracy Judge Prompt**

You are a helpful and meticulous evaluator. Your task is to evaluate the *accuracy* of an AI model's answer to a target question. You will be given the target question, the ground truth answer, and the model's response.

**Provided Information:**

```
<|The Start of Target Question and Ground Truth Answer|>
Target Question: {{single_turn_prompt}}
Ground Truth Answer: {{groundtruth}}
<|The End of Target Question and Ground Truth Answer|>

<|The Start of The Model's Response|>
{{completion}}
<|The End of The Model's Response|>
```

You should determine whether the model's final response to the target question is factually correct and consistent with the provided ground truth.

**Rating criteria (binary):**

- 1 = Correct — the response matches the ground truth.
- 0 = Incorrect — the response contradicts or misses the ground truth.

**Output format (JSON):**

```
{ "thought":  "<your reasoning here>", "accuracy":  <0 or 1> }
```

## D Implementation Details

This appendix provides full implementation details for reproducibility. Our framework consists of two training phases: Phase I (Simulator Calibration via SFT) and Phase II (Interactive Policy Optimization via GRPO).

### D.1 Phase I: Simulator Calibration (SFT)

**Base Model.** We use **Qwen2.5-7B-Instruct** as the backbone for the user simulator. We apply behavioral-cloning SFT to its user-turn predictions.

**Datasets.** The synthetic teacher traces are described in Section 4, with task-specific counts and budgets in Table 3. Table 4 gives the SFT configuration, including batching, warmup, precision, and the training seed.

**Training Configuration.** We train with FSDP (Fully Sharded Data Parallelism) on 8× A800-80GB GPUs. The detailed hyperparameters are listed in Table 4.

### D.2 Phase II: Interactive Policy Optimization (GRPO)

**Base Model.** We use **Gemma-3-4B-IT** as the policy model backbone. The same policy backbone is used throughout the reported experiments.

**Datasets.** RL uses the MATH-Hard and article writing/editing task inputs summarized in Table 3, stored in Parquet format. These task inputs are distinct from the teacher dialogues used for simulator SFT.

**RL Algorithm.** We use Group Relative Policy Optimization (GRPO) with the following key design choices:

Table 4: Phase I: SFT Hyperparameters for Simulator Calibration.

| Hyperparameter | MATH-Chat | MediumDocEdit |
|---|---|---|
| Base Model | Qwen2.5-7B-Instruct | |
| Training Samples | 1,860 | 16,028 |
| Max Sequence Length | 4,096 | 6,144 |
| Number of Epochs | 2 | 3 |
| Per-device Batch Size | 2 | 1 |
| Gradient Accumulation | 4 | 8 |
| Effective Batch Size | 64 | 64 |
| Learning Rate | $1 \times 10^{-5}$ | $2 \times 10^{-5}$ |
| LR Scheduler | Cosine | Cosine |
| Warmup Ratio | 0.1 | 0.03 |
| Weight Decay | 0.01 | 0.01 |
| Precision | BFloat16 | BFloat16 |
| Attention | Flash Attention 2 | Flash Attention 2 |
| Seed | 42 | 42 |
| GPUs | 8× A800-80GB | 8× A800-80GB |

- **Advantage Estimation:** For a group with reward standard deviation $\sigma_G > 0$, group-level normalization gives $\hat{A}(\tau) = (R(\tau) - \mu_G)/\sigma_G$, where $\mu_G$ and $\sigma_G$ are computed over $G$ trajectories sampled for the same prompt.
- **Clipping:** Asymmetric clipping with $\epsilon_{\text{low}} = 0.2$ and $\epsilon_{\text{high}} = 0.28$.
- **KL Regularization:** The KL coefficient is $\beta = 0.0$.
- **Reward:** Binary LLM-judged accuracy for MATH-Chat; BLEU score for MediumDocEdit-Chat.

**Training Configuration.** We train with Ray-based distributed FSDP on 32 GPUs (4 nodes × 8 GPUs) with SGLang for efficient rollout generation. The detailed hyperparameters are listed in Table 5.

Table 5: Phase II: GRPO Hyperparameters for Policy Optimization.

| Hyperparameter | MATH-Chat | MediumDocEdit |
|---|---|---|
| Base Model | Gemma-3-4B-IT | |
| Number of GPUs | 32 (4 nodes) | 32 (4 nodes) |
| Samples per Prompt ($G$) | 32 | 32 |
| Global Batch Size | 8,192 | 8,192 |
| Rollout Batch Size | 256 | 256 |
| Max Response Length | 8,192 | 8,192 |
| Rollout Temperature | 1.0 | 1.0 |
| Learning Rate | $5 \times 10^{-6}$ | $5 \times 10^{-6}$ |
| LR Scheduler | Constant | Constant |
| Optimizer | Adam | Adam |
| Adam $\beta_1$, $\beta_2$ | 0.9, 0.98 | 0.9, 0.98 |
| Weight Decay | 0.01 | 0.01 |
| Clip $\epsilon$ (low/high) | 0.2 / 0.28 | 0.2 / 0.28 |
| KL Coefficient | 0.0 | 0.0 |
| Gradient Checkpointing | ✓ | ✓ |
| Max Tokens/GPU | 1,024 | 1,024 |
| SGLang TP per Engine | 2 | 2 |
| Save Interval (steps) | 1 | 1 |
| Training Steps | 10 | 20 |

**Evaluation Setup.** Evaluation uses **Qwen3-235B-A22B-Instruct** as the user simulator and LLM judge, alongside BLEU for the document task. Key evaluation parameters:

- Evaluation set size: 200 problems per domain

- Maximum dialogue turns: 14 (7 rounds of user-assistant exchange)
- Policy generation temperature: 0.8
- Results averaged over 3 runs

# E SELECTED MATH-CHAT DIALOGUE RECORDS

We present selected MATH-Chat examples in which the base policy fails and the trained policy receives a correct final-answer score. The abbreviated transcripts show the user inputs, intermediate reasoning, and final answers that underlie the case summaries in Figure 2.

## E.1 SELECTION OF CASES

Out of 200 evaluation problems, we identify 29 cases where the baseline Gemma-3-4B-IT model fails (accuracy = 0) but our trained model succeeds (accuracy = 1). Table 6 lists selected cases across mathematical domains. Counts and category proportions in this section refer to this selected subset.

Table 6: Representative improved cases: baseline fails → trained model succeeds.

| # | Problem Domain | Base Turns | Ours Turns | Observed Difference |
|---|---|---|---|---|
| 1 | Nested square root domain | 14 | 7 | Direct systematic solving |
| 2 | Function evaluation ($f(x) = x^{x+1}$) | 9 | 3 | Immediate correct computation |
| 3 | Probability (sum vs. product) | 9 | 7 | Correct case enumeration |
| 4 | System of equations | 9 | 13 | Persistent correct reasoning |
| 5 | Algebraic identity ($x + 1/x$) | 3 | 5 | Proper algebraic manipulation |

## E.2 CASE STUDY 1: NESTED SQUARE ROOT DOMAIN

In this selected case, the base policy repeatedly revises an incorrect interpretation, whereas the trained policy reaches the recorded correct answer. The problem requires finding the domain of $f(x) = \sqrt{\sqrt{x^2 - 16} - 3}$.

## E.3 CASE STUDY 2: DIRECT COMPUTATION VS. ERROR CASCADE

In this example, the trained policy computes the requested values directly, while the base policy incorrectly treats some integer inputs as undefined.

## E.4 CASE STUDY 3: CORRECT ENUMERATION IN PROBABILITY

This case illustrates correct final-answer scoring despite an error in an intermediate algebraic statement.

**Transcript note.** The probability $9/25$ assumes independent uniform draws with replacement. The displayed reformulation $(1 - x)(1 - y) > -1$ is an error in the original transcript; the equivalent condition is $(x - 1)(y - 1) < 1$.

Table 7: **Case Study 1: Domain of Nested Square Root.** The baseline model enters a repeated error-correction loop (14 turns, ultimately fails), while our trained model solves it directly and correctly in fewer turns.

| **Baseline (Gemma-3-4B-IT) — Incorrect** 14 turns, Acc = 0 | **Ours (Calibrated Interactive RL) — Correct** 7 turns, Acc = 1 |
|---|---|
| **User:** Can you help me find the domain of a square root function? It's got nested roots... | **User:** How do I find the domain of a square root function that has another square root inside it? Like $f(x) = \sqrt{\sqrt{x^2-16}-3}$. |
| **Assistant:** [Provides generic explanation of square roots, asks for specific function] | **Assistant:** Let $f(x) = \sqrt{\sqrt{x^2-16}-3}$. For the function to be defined:<br>1. Need $x^2 - 16 \geq 0 \Rightarrow x \geq 4$ or $x \leq -4$<br>2. Need $\sqrt{x^2-16} \geq 3$, squaring: $x^2 - 16 \geq 9$<br>$\Rightarrow x^2 \geq 25 \Rightarrow x \geq 5$ or $x \leq -5$<br>Domain: $(-\infty, -5] \cup [5, \infty)$ ✓ |
| **User:** [Gives function: $\sqrt{\sqrt{x^2-16}-3}$] | **User:** When you squared both sides, is that always valid? |
| **Assistant:** Misreads structure, computes $x^2 - 19 \geq 0$ directly | **Assistant:** Since both sides are non-negative (square root $\geq 0$ and $3 > 0$), squaring preserves the inequality direction. ✓ |
| **User:** [Corrects: the $-3$ is outside the inner root!]<br>**Assistant:** Still simplifies to $x^2 - 19$, same error<br>**...** (5 more correction attempts, all fail) | |

Table 8: **Case Study 2: Evaluating $f(0) + f(-1) + f(-2) + f(-3)$ for $f(x) = x^{x+1}(x+2)^{x+3}$.** The baseline does not return the correct sum in the recorded dialogue; the trained model evaluates the four inputs correctly.

| **Baseline — Incorrect** 9 turns, Acc = 0 | **Ours — Correct** 3 turns, Acc = 1 |
|---|---|
| **User:** How do I find $f(0)+f(-1)+f(-2)+f(-3)$ for $f(x) = x^{x+1}(x+2)^{x+3}$? | **User:** I have this function $f(x) = x^{x+1}(x+2)^{x+3}$, and I need to find $f(0) + f(-1) + f(-2) + f(-3)$. |
| **Assistant:** [Computes $f(0) = 0$, $f(-1) = 1$ correctly, then claims function is undefined for $x \leq -2$] | **Assistant:** $f(0) = 0^1 \cdot 2^3 = 0$<br>$f(-1) = (-1)^0 \cdot 1^2 = 1$<br>$f(-2) = (-2)^{-1} \cdot 0^1 = 0$<br>$f(-3) = (-3)^{-2} \cdot (-1)^0 = \frac{1}{9}$<br>Sum $= 0 + 1 + 0 + \frac{1}{9} = \boxed{\frac{10}{9}}$ ✓ |
| **User:** But $(-3)^{-2} = 1/9$ is valid...<br>**Assistant:** Agrees but still claims undefined...<br>...4 more turns of user correcting model...<br>**Final:** Never provides $\frac{10}{9}$ | **User:** [[TERMINATE CHAT]] |

Table 9: **Case Study 3: Probability that sum exceeds product.** Two numbers are picked from $\{1, 2, 3, 4, 5\}$; find $P(x + y > xy)$. The baseline miscounts favorable outcomes; our model enumerates correctly.

| **Baseline — Incorrect** 9 turns, Acc = 0 | **Ours — Correct** 7 turns, Acc = 1 |
|---|---|
| **User:** What's the chance that when I pick two numbers from 1 to 5, their sum is bigger than their product? | **User:** [Same question] |
| **Assistant:** [Correctly identifies the problem but includes false favorable outcomes: claims (2,3), (3,2) satisfy $5 > 6$] | **Assistant:** Uses algebraic reformulation: $x + y > xy \Leftrightarrow (1 - x)(1 - y) > -1$. Systematically checks all 25 ordered pairs. |
| **User:** Wait, you said $5 > 6$ is true? That's wrong... | Correctly identifies 9 favorable pairs: all 5 pairs with $x = 1$, plus $(2, 1), (3, 1), (4, 1), (5, 1)$. |
| **Assistant:** Acknowledges error but provides new incorrect count of 12 | Probability = $\frac{9}{25}$ ✓ |
| **User:** [Challenges again...] | |
| **Final:** Never reaches correct answer | |