TEST-TIME POLICY ADAPTATION FOR ENHANCED MULTI-TURN INTERACTIONS WITH LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large Language Models (LLMs) employ multi-turn interaction as a fundamental paradigm for completing complex tasks. However, their performance often degrades in extended interactions, as they are typically trained on static, single-turn data, which hinders their ability to adapt to real-time user feedback. To address this limitation, we first propose a new paradigm: *Test-Time Policy Adaptation for Multi-Turn Interactions* (T^2PAM), which utilizes user feedback from the ongoing interaction as a reward signal to estimate a latent optimal policy aligned with user preferences, then updates a small subset of parameters to steer the model toward this policy, ultimately enabling efficient in-conversation self-correction. We then introduce *Optimum-Referenced One-Step Adaptation* (ROSA), a lightweight algorithm that operationalizes T^2PAM. ROSA guides the model parameters toward a theoretical optimal policy in a single, efficient update step, avoiding costly iterative gradient-based optimization and minimizing computational overhead. We provide a rigorous theoretical analysis guaranteeing that the policy of ROSA converges to the preference of user as the number of interactions increases. Extensive experiments on challenging benchmark demonstrate that ROSA achieves significant improvements in both effectiveness and efficiency.

1 INTRODUCTION

Multi-turn conversation is the predominant interaction paradigm between human and *Large Language Models* (LLMs) (Li et al., 2025b; Yi et al., 2025). This conversational modality is essential for real-world applications (Zhang et al., 2025a), as it enables users to progressively refine initially underspecified intentions into concrete objectives (Herlihy et al., 2024; Zheng et al., 2023), engaging the model in a collaborative problem-solving process (Chen et al., 2023). However, a fundamental mismatch exists between this prevalent use case and existing LLM alignment methodologies (Laban et al., 2025; Van Miltenburg et al., 2025). Prevailing alignment methods, *Supervised Fine-Tuning* (SFT) (Chung et al., 2024; Wei et al., 2025a; Lester et al., 2021) and *Reinforcement Learning from Human Feedback* (RLHF) (Ouyang et al., 2022; Wei et al., 2025c), predominantly rely on single-turn data for both training (Shu et al., 2024) and evaluation (Chang et al., 2023). This paradigm misalignment not only limits the potential of the model in complex interactions (Irvine et al., 2023; Hendrycks et al., 2021c), but also creates a significant gap between its benchmark performance and its practical utility (Shinn et al., 2023; Wu et al., 2024). Consequently, while the combination of SFT for imparting extensive knowledge (Chu et al., 2025; Wei et al., 2025b) and RLHF for aligning with human preferences (Rafailov et al., 2023; Meng et al., 2024) endows models with strong single-turn capabilities (Zeng et al., 2024), these models often exhibit a pronounced degradation in performance during multi-turn interactions (Wang et al., 2024). In fact, previous work has highlighted that such models often perform poorly in multi-turn scenarios, resulting in diminished capabilities and increased instability (Laban et al., 2025). While multi-turn training strategies have been explored (Shi et al., 2025; Qu et al., 2024; Chen et al., 2025a), they are frequently hindered by the costs of collecting high-quality data, training on long context sequences (Li et al., 2025b), and the risk of behavioral collapse caused by training-inference distribution mismatch (Ren & Sutherland, 2025).

To address these challenges, we propose a new paradigm: *Test-Time Policy Adaptation for Multi-Turn Interactions* (T^2PAM), shifting the existing static training paradigm to a flexible test-time adaption one. Specifically, it requires a model trained on single-turn data to perform effective online policy adaptation during multi-turn reasoning. This paradigm utilizes conversational user feedback

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.

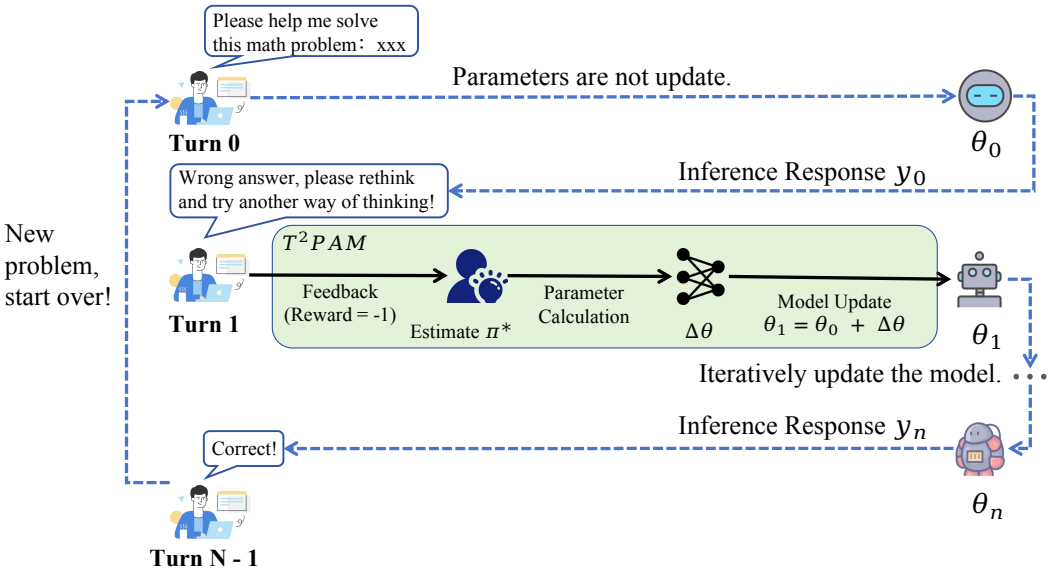

Figure 1: An illustration of the *Test-Time Policy Adaptation for Multi-Turn Interactions* (T^2PAM) paradigm. Different from static inference where the policy of model remains fixed (θ_0, Turn 0), this paradigm treats conversational feedback as an active signal that guides real-time parameter updates (e.g., from θ_0 to θ_1). This iterative process of in-conversation self-correction allows the policy to progressively evolve and align with the preference of user (θ_n) throughout the interaction.

as a reward signal to refine its policy and align with the underlying user intent, as illustrated in Figure 1. Importantly, this adaptation must be computationally lightweight to remain imperceptible without incurring unaffordable inference latency or GPU memory overhead. Under this paradigm, the model dynamically instantiates a user-specific policy for each conversational context, enhancing multi-turn interaction reliability. While target preferences naturally vary across taskssuch as strict correctness for math versus personal styles in othersthe ultimate objective remains universally consistent: generating a response the user fully accepts.

Existing methodologies (Shani et al., 2024) tackle adaptation but lack the lightweight, feedback-driven updates of T^2PAM. Specifically: (1) *Prompt Engineering* (Hu et al., 2024; Chen et al., 2023; Shinn et al., 2023) alters behavior via context rather than direct parameter updates. (2) *Retrieval-Augmented Generation (RAG)* (Gao et al., 2024; Lewis et al., 2020) increases inference overhead by lengthening context and heavily relies on external database quality. (3) *Model Editing (ME)* (Fang et al., 2025; Yao et al., 2023) updates parameters to internalize facts, which differs from adapting to conversational preferences. (4) Existing *test-time methods* (Li et al., 2025a; Zuo et al., 2025; Hu et al., 2025; Liu et al., 2023) target single-turn tasks and incur significant computational latency through extensive sampling. See Appendix A for details.

To bridge this gap, we introduce *Optimum-Referenced One-Step Adaptation* (ROSA), a lightweight online adaptation algorithm that operationalizes our proposed paradigm T^2PAM. The core principle of ROSA is to leverage user feedback to analytically compute an estimate of the optimal policy and then steer the model towards this target in a single, efficient update step. This approach avoids costly iterative optimization, enabling principled in-conversation self-correction with minimal computational overhead. Our main contributions are summarized as follows:

- We demonstrate that current LLMs underperform in multi-turn interactions and propose T^2PAM paradigm to address this issue (Section 2).
- We propose ROSA, a practical algorithm to implement this paradigm, which updates model parameters and align user preferences quickly during multi-turn interactions (Section 3).
- We establish a solid theory for ROSA, ensuring that its gap with user preferences narrows as the number of interaction turns increases (Section 4).
- We conduct extensive experiments on multiple challenging datasets. Our results show that ROSA outperforms baseline methods in both effectiveness and efficiency (Section 5).

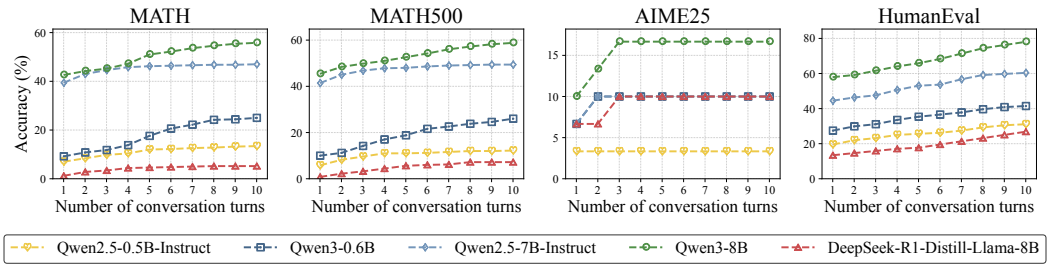

Figure 2: LLM accuracy after 10 rounds of interaction with humans. Although LLM accuracy shows a continuous and gradual improvement, this prompt-based correction process is inefficient.

2 THE T^{2}PAM PARADIGM

The performance of LLMs often degrades in multi-turn interactions, because their alignment on static, single-turn datasets creates a paradigm mismatch that hinders their ability to adapt to user feedback or correct initial errors (Laban et al., 2025). To show this inefficiency, we empirically evaluated several LLMs on reasoning tasks. We first plot the cumulative accuracy over 10 conversational turns where human-like prompts were provided after each incorrect attempt. The results in Figure 2 show that while multi-turn interaction gradually improves accuracy, the process exhibits sharply diminishing returns. To diagnose this, Figure 3 plots the number of *newly* solved problems at each conversational turn on the `MATH` dataset. The data reveal that the vast majority of problems are solved on the first attempt, with very few successful corrections in subsequent turns. This demonstrates that current models treat user interactions as passive context rather than as active signals for policy correction, highlighting a critical gap in their ability to perform efficient test-time adaptation.

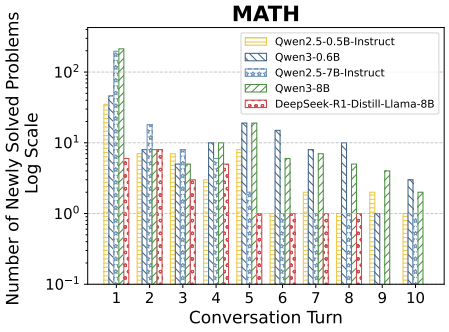

Figure 3: Number of *newly* solved problems per turn on the `MATH` dataset.

To address this gap, we propose a new paradigm: *test-time policy adaptation for multi-turn interactions* (T^{2}PAM). As summarized in Table 1, T^{2}PAM resolves the trade-off in traditional approaches. While prompting lacks real-time adaptability and multi-turn training is costly, yielding a static policy, T^{2}PAM synthesizes the benefits of both. It operates during inference with zero additional offline training cost but, through online parameter modification, achieves high, policy-level adaptability that is more direct than prompting and more flexible than offline training. Notably, this paradigm shifts model alignment from a static, offline training stage to a dynamic, online inference process. More specifically, it requires methods to update the model's policy in real-time by directly leveraging rich feedback signals from live conversations. We formally define T^{2}PAM below:

Paradigm: Test-Time Policy Adaptation for Multi-Turn Interactions (T^{2}PAM)

Let an inference-time multi-turn interaction be a sequence of interactions indexed by turn $k \in \{1, \dots\}$. At the beginning of turn k, the language model is defined by a policy $\pi_{\theta_{k-1}}$ with parameters θ_{k-1}. The paradigm proceeds as follows:

1. **Generation**: The model generates a response $\mathbf{y}_k \sim \pi_{\theta_{k-1}}(\cdot|\mathbf{x})$ given the conversational context $\mathbf{x}$.
2. **Feedback**: The subsequent interaction of user provides feedback, which is mapped to a scalar reward r_k indicating task success (i.e., $r_k = +1$) or failure (i.e., $r_k = -1$).
3. **Adaptation**: If the task succeeds (i.e., $r_k = +1$), the multi-turn interaction is finished. Otherwise, an **effective and efficient** online adaptation function $\mathcal{A}$ updates the model parameters at inference time based on this failure feedback (i.e., $r_k = -1$) such that the model is more likely to succeed in the next turn:

$$\theta_k = \mathcal{A}(\theta_{k-1}, r_k, \mathbf{y}_k; \mathbf{x}) = \theta_{k-1} + \Delta\theta_k \ .$$

Table 1: Conceptual comparison of paradigms for improving multi-turn LLM performance.

Feature	Prompt-based Methods	Multi-turn Data Training	TšPAM (Ours)
Intervention Timing	During inference	During training	During inference
Operating Mode	Reactive	**Proactive** (at training)	**Proactive** (at inference)
Additional Offline Training Cost	**Zero** (uses single-turn model)	High	**Zero** (uses single-turn model)
Inference Cost	Low (long context)	**Near-zero**	Low (preference alignment)
Parameter Modification	No	Yes (offline)	Yes (online)
Real-time Adaptability	Low (context-dependent)	Low (static policy)	**High** (policy-level)

3 Optimum-Referenced One-Step Adaptation

To solve the paradigm we proposed above, we develop the *Optimum-Referenced One-Step Adaptation* (ROSA) approach (Algorithm 1), which enables effective and efficient online adaptation of a language model policy in direct response to real-time user feedback during multi-turn interactions. The core principle is to guide the model parameters towards a theoretical optimum in a single, efficient update step, avoiding iterative gradient-based optimization. This approach first defines the *Reinforcement Learning from Human Feedback* (RLHF) objective (Section 3.1) to maximize reward with KL regularization. It then leverages a closed-form analytical solution to directly identify the optimal policy (Section 3.2), applying exponential re-weighting to observed responses for practical one-step updates. Finally, parameter updates are efficiently computed via linearized optimization using the *Conjugate Gradient* algorithm (Section 3.3).

3.1 The RLHF Objective for Turn-Wise Adaptation

We propose to solve the T^{2}PAM paradigm above using *Reinforcement Learning from Human Feedback* (RLHF) techniques (Ouyang et al., 2022). In this approach, we learn from a reward signal $r(\mathbf{x}, \mathbf{y})$ that reflects human preference given the context $\mathbf{x}$ and the response $\mathbf{y}$. Specifically, we model this feedback as a binary signal where $r(\mathbf{x}, \mathbf{y}) \in \{-1, +1\}$ corresponds to negative and positive feedback, respectively. The objective is to find an updated policy π_θ that maximizes the expected reward while penalizing significant divergence from the policy of the previous turn $\pi_{\theta_{k-1}}$ for stable and controlled updates. The deviation is measured by the *Kullback-Leibler* (KL) divergence. This leads to the following turn-wise optimization objective for turn k:

$$\max_{\pi_\theta} \quad \mathbb{E}_{\mathbf{y}\sim\pi_\theta(\cdot|\mathbf{x})}\left[r(\mathbf{x}, \mathbf{y})\right] - \beta D_{\mathrm{KL}}\left(\pi_\theta(\cdot|\mathbf{x}) \,\|\, \pi_{\theta_{k-1}}(\cdot|\mathbf{x})\right) \tag{1}$$

where $\beta > 0$ is a coefficient that controls the strength of the KL regularization.

3.2 From Theoretical Optimum to a Practical One-Step Update

While the objective presented in Equation equation 1 is conventionally optimized using iterative gradient-based methods (Sra et al., 2011; Kingma & Ba, 2017), such approaches are often characterized by their computational intensity and slow convergence, rendering them impractical for real-time online adaptation scenarios. Our methodology circumvents this inefficiency by leveraging a critical insight: this specific optimization problem admits a well-established closed-form analytical solution (Rafailov et al., 2023). Rather than relying on incremental approximations, we can directly ascertain the optimal policy. This foundational result is formalized in Theorem 1 (proof in Appendix B.1).

Theorem 1 (Closed-Form Optimal Policy)**.** *Let $Z_k(\mathbf{x}) = \sum_{\mathbf{y}'\in\mathcal{Y}} \pi_{\theta_{k-1}}(\mathbf{y}'|\mathbf{x}) \exp\left(\frac{1}{\beta} r(\mathbf{x}, \mathbf{y}')\right)$ be the partition function over the entire response space $\mathcal{Y}$, the policy $\pi^*_{\theta_k}$ that maximizes the turn-wise RLHF objective in Equation equation 1 is given by:*

$$\pi^*_{\theta_k}(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_k(\mathbf{x})} \pi_{\theta_{k-1}}(\mathbf{y}|\mathbf{x}) \exp\left(\frac{1}{\beta} r(\mathbf{x}, \mathbf{y})\right) . \tag{2}$$

Theorem 1 demonstrates that the optimal policy is a re-weighted version of the reference policy, where the probability of a given response is exponentially modulated by its associated reward. In

Algorithm 1 Optimum-Referenced One-Step Adaptation (ROSA)

Input: Initial model parameters θ_0, hyperparameter β.
$k \leftarrow 1$
while true do
  // Step 1: Generate response and receive feedback
  Generate response $\mathbf{y}_k \sim \pi_{\theta_{k-1}}(\cdot|\mathbf{x})$.
  Receive reward r_k based on user feedback.
  if $r_k = +1$ **then**
    Terminate // Stop immediately on success signal
  end if
  // Step 2: Construct the practical online target (Section 3.2)
  Compute target value $\tilde{\pi}^*_{\theta_k}(\mathbf{y}_k|\mathbf{x}) = \frac{1}{Z_k(\mathbf{x})}\pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x}) \exp\left(\frac{1}{\beta}r_k\right)$.
  // Step 3: Compute parameter update via linearized optimization (Section 3.3)
  Define residual $\mathbf{d}_k = \tilde{\pi}^*_{\theta_k} - \pi_{\theta_{k-1}}$.
  Solve $(\mathbf{J}_k^\top \mathbf{J}_k)\Delta\theta_k = \mathbf{J}_k^\top \mathbf{d}_k$ for $\Delta\theta_k$ using *Conjugate Gradient* method.
  // Step 4: Update model parameters
  Update parameters: $\theta_k \leftarrow \theta_{k-1} + \Delta\theta_k$.
end while

practical applications, feedback is typically received for only a *single* generated response, $\mathbf{y}_k$, often corresponding to a negative reward ($r_k = -1$) for an incorrect output. This constraint necessitates the construction of an update target utilizing solely the observed data point $(\mathbf{x}, \mathbf{y}_k, r_k)$. We achieve this by applying the exponential re-weighting derived from the optimal policy in Equation equation 2 exclusively to the observed response, thereby yielding a practical target value (derivation in Appendix B.2):

$$\tilde{\pi}^*_{\theta_k}(\mathbf{y}|\mathbf{x}) = \begin{cases} \frac{1}{Z_k(\mathbf{x})}\pi_{\theta_{k-1}}(\mathbf{y}|\mathbf{x}) \exp\left(\frac{1}{\beta}r_k\right), & \text{if } \mathbf{y} = \mathbf{y}_k, \\ \frac{1}{Z_k(\mathbf{x})}\pi_{\theta_{k-1}}(\mathbf{y}|\mathbf{x}), & \text{if } \mathbf{y} \neq \mathbf{y}_k\ . \end{cases} \tag{3}$$

where $Z_k(\mathbf{x}) = 1 - \left(1 - \exp\left(\frac{1}{\beta}r_k\right)\right)\pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x})$. This formulation provides a direct learning signal for a one-step parameter update. For an incorrect response with reward $r_k = -1$, the target probability is scaled down relative to the current policy, effectively instructing the model to diminish the likelihood of generating that specific erroneous output in the future. This approach transforms an otherwise intractable global optimization problem into a targeted, sample-wise correction, forming the fundamental basis for our efficient adaptation mechanism.

3.3 Efficient Parameter Update via Linearized Optimization

With a practical target policy $\tilde{\pi}^*_{\theta_k}$ established, the subsequent step involves computing the parameter update $\Delta\theta_k$ that adjusts the current policy $\pi_{\theta_{k-1}}$ towards this target. This is accomplished through linearized optimization. This linearization is chosen for its computational ease and efficiency, allowing for rapid online adaptation without the prohibitive costs of higher-order optimization methods, as demonstrated in our efficiency analysis in Section D. Initially, the policy function is approximated using a first-order Taylor expansion around the current parameters θ_{k-1}:

$$\pi_{\theta_{k-1}+\Delta\theta_k}(\mathbf{y}_k|\mathbf{x}) \approx \pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x}) + \nabla_\theta \pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x})^\top \Delta\theta_k\ . \tag{4}$$

Our objective is to determine $\Delta\theta_k$ such that the updated policy $\pi_{\theta_{k-1}+\Delta\theta_k}$ closely matches our target $\tilde{\pi}^*_{\theta_k}$. For the single data point $(\mathbf{x}, \mathbf{y}_k)$, this yields a linear system of equations:

$$\mathbf{J}_k \Delta\theta_k \approx \tilde{\pi}^*_{\theta_k}(\mathbf{y}_k|\mathbf{x}) - \pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x})\ . \tag{5}$$

where $\mathbf{J}_k = \nabla_\theta \pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x})^\top$ represents the Jacobian of the policy output with respect to the model parameters. To obtain a stable, least-squares solution for $\Delta\theta_k$, we solve the following equations:

$$(\mathbf{J}_k^\top \mathbf{J}_k)\Delta\theta_k = \mathbf{J}_k^\top \left(\tilde{\pi}^*_{\theta_k}(\mathbf{y}_k|\mathbf{x}) - \pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x})\right)\ . \tag{6}$$

Explicitly forming the Hessian-approximating matrix $\mathbf{J}_k^\top \mathbf{J}_k$ is computationally prohibitive for models with a large number of parameters. As a consequence, we employ the *Conjugate Gradient*

(CG) algorithm (Atkinson, 1988), an iterative solver that efficiently determines the solution to Equation equation 6 without materializing this matrix. This is critical for memory efficiency, as it avoids storing the full Hessian-like matrix, making our approach incur less GPU memory overhead, as shown in Appendix D. The CG algorithm only requires the computation of the matrix-vector product $(\mathbf{J}_k^\top \mathbf{J}_k)\mathbf{p}$ for an arbitrary vector $\mathbf{p}$. This computation is performed in a matrix-free manner by efficiently chaining two operations using automatic differentiation: a Jacobian-vector product (JVP) to compute $\mathbf{J}_k\mathbf{p}$, followed by a vector-Jacobian product (VJP) to compute $\mathbf{J}_k^\top(\mathbf{J}_k\mathbf{p})$. Positive-feedback updates and learning from successful trajectories are evaluated in Appendix E.11.

Once the optimal $\Delta\theta_k$ is computed via CG method, the model parameters are updated in one step:

$$\theta_k \leftarrow \theta_{k-1} + \Delta\theta_k \,. \tag{7}$$

This entire procedure, encompassing feedback reception and parameter update computation, constitutes one complete cycle of ROSA, as comprehensively detailed in Algorithm 1.

4 Theoretical Results

Having established the mechanics of ROSA, we now provide its theoretical underpinnings. This section demonstrates that our ROSA is not merely an effective heuristic but a principled algorithm with formal guarantees. Our analysis unfolds in three stages: we first prove that each corrective step is guaranteed to be productive (Section 4.1), then bound the accumulated gains over corrective updates (Section 4.2), and finally, provide a unified bound that accounts for the practical approximation errors inherent in our efficient update step (Section 4.3).

Of note, a central aspect of our theoretical analysis revolves around the *Kullback-Leibler* (KL) divergence, specifically $D_{\mathrm{KL}}(\pi^*_{\text{user}} \| \tilde{\pi}^*_{\theta_k})$. This metric quantifies the dissimilarity between the underlying user optimal policy π^*_{user} (representing the true preferences from the user and the ideal way to solve the task) and our adapted policy $\tilde{\pi}^*_{\theta_k}$. Minimizing this divergence is crucial because it directly implies that the generated responses from a model are becoming increasingly aligned with what the user desires and expects. When the model policy closely mirrors the user optimal policy, it is inherently more likely to produce correct and satisfactory outputs, thereby increasing the probability of task success and reducing the number of interaction turns required to achieve user intent.

For this analysis, define the ideal user policy as any distribution supported entirely on the acceptable-response set $\mathcal{A}(\mathbf{x})$. Thus, $\pi^*_{\text{user}}(\mathbf{y} \mid \mathbf{x}) = 0$ for unacceptable responses. The bounds below concern reliable negative feedback identifying such responses, at a fixed $\mathbf{x}$, with finite initial KL divergence; they do not require a unique distribution over acceptable answers.

4.1 Monotonic Error Reduction

Our first key result establishes that the adaptation mechanism in ROSA is *provably productive*. Each corrective feedback, the resulting update is guaranteed to reduce the KL divergence between the user policy and our estimated target policy, as formally shown in Theorem 2 (proof in Appendix B.3).

Theorem 2 (Monotonic Error Reduction)**.** *Let π^*_{user} be the underlying user policy and $\tilde{\pi}^*_{\theta_k}$ be the practical target policy in Equation equation 3 after receiving negative feedback $r_k = -1$ on an incorrect response $\mathbf{y}_k$ at turn k. Suppose $\pi_{\theta_k} = \tilde{\pi}^*_{\theta_k}$ by applying the exact policy update in ROSA. Letting $C = 1 - \exp(-1/\beta) \in (0, 1)$, the change in KL divergence from the previous turn is bounded as follows:*

$$D_{\mathrm{KL}}(\pi^*_{\text{user}} \| \tilde{\pi}^*_{\theta_k}) - D_{\mathrm{KL}}(\pi^*_{\text{user}} \| \tilde{\pi}^*_{\theta_{k-1}}) \leq -C \cdot \pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x}) < 0 \,. \tag{8}$$

Remark. This theorem provides a powerful guarantee for the reliability of ROSA. The upper bound of this reduction consists of two highly informative terms, both of which offer significant insights into the adaptation dynamics. First, the term C acts as a scaling factor regulated by β, where a smaller β yields a larger C that leads to a faster reduction in error. While this implies rapid initial learning, it introduces a critical trade-off with approximation errors that we formally quantify in Section 4.3. Second, the $\pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x})$ term reveals that the error reduction is strictly proportional to the model's prior confidence in its own mistake. This elegant property indicates that the most

impactful learning signals come from correcting high-confidence errors instead of random, low-probability hallucinations. In summary, this result confirms that every piece of corrective negative feedback is guaranteed to be productive, which provides strong theoretical justification for the one-step adaptation design in ROSA by avoiding the potential instability of iterative optimization.

4.2 Cumulative Convergence Guarantee

While Theorem 2 guarantees improvement at each step, our second theorem extends this result to the entire multi-turn interaction, providing a bound on the cumulative error over reliably rejected responses in our Theorem 3 (proof in Appendix B.4).

Theorem 3 (Cumulative Error Bound). *Suppose $\pi_{\theta_k} = \tilde{\pi}^*_{\theta_k}$ by applying exact policy update in ROSA, after K reliably rejected-response updates, the KL divergence between the underlying user policy π^*_{user} and the practical target policy $\tilde{\pi}^*_{\theta_K}$ in Equation equation 3 is bounded as follows:*

$$D_{\mathrm{KL}}(\pi^*_{\mathrm{user}} \| \tilde{\pi}^*_{\theta_K}) \leq D_{\mathrm{KL}}(\pi^*_{\mathrm{user}} \| \pi_{\theta_0}) - C \sum_{k=1}^{K} \pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x}) \,. \tag{9}$$

Remark. The benefits of exact corrective updates accumulate over time through the subtracted summation term. The bound describes progress toward the acceptable-response set; convergence to a particular distribution within that set requires additional conditions. Appendix B.4 gives the corresponding conditional sample-efficiency analysis under diverse failure modes.

4.3 Unified Error Bound for the Adapted Policy

The previous theorems guarantee our *target* policy improves. However, the *final* policy, π_{θ_k}, is subject to the approximation error from the first-order Taylor expansion used for our efficient update. The following unified theorem combines the guaranteed improvement from feedback with the accumulated error to provide a comprehensive bound on the true performance of ROSA (proof in Appendix B.5).

Theorem 4 (Unified Convergence Bound). *Assume $\log \pi_\theta$ is Lipschitz-smooth with constant L along each update, and let ϵ_k bound the weighted linearized target residual defined in Appendix B.5. After K reliably rejected-response updates in ROSA, the divergence of the final adapted policy π_{θ_K} from the underlying user policy π^*_{user} is bounded by:*

$$D_{\mathrm{KL}}(\pi^*_{\mathrm{user}} \| \pi_{\theta_K}) \leq \underbrace{D_{\mathrm{KL}}(\pi^*_{\mathrm{user}} \| \pi_{\theta_0})}_{\text{Initial Error}} \underbrace{- C \sum_{k=1}^{K} \pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x})}_{\text{Improvement}} + \underbrace{\frac{L}{2} \sum_{k=1}^{K} \|\Delta\theta_k\|_2^2 + \sum_{k=1}^{K} \epsilon_k}_{\text{Approx. Error}} \,. \tag{10}$$

Remark. This unified bound rigorously quantifies the inherent trade-off in online policy adaptation. Each turn reduces the KL divergence from the underlying user optimal policy by a reward-driven term $C\pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x})$, while incurring an approximation error $\frac{L}{2}\|\Delta\theta_k\|_2^2$ due to linearization, together with a linearized target-fitting error ϵ_k. Convergence requires the net progress per turn to remain positive. This balance is affected by two factors. Firstly, the approximation error is controlled because the update magnitude $\Delta\theta_k$ is constrained by $\pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x})$ according to Equation equation 3, which ensures the guaranteed improvement can effectively outweigh the approximation cost. Additionally, our algorithm effectively minimizes the fitting error ϵ_k by employing the Conjugate Gradient (CG) solver. Secondly, the regularization coefficient β modulates this trade-off by directly determining the constant C and indirectly affecting the update norm $\|\Delta\theta_k\|_2$. A smaller β yields a larger C to accelerate learning, but risks amplifying the approximation error, whereas a larger β stabilizes updates at the cost of slower progress. This interplay explains the two-phase behavior observed in practice, which features rapid initial corrections followed by stable, fine-grained refinements as detailed in Appendix F.2. The theorem therefore serves as both a robust theoretical guarantee and a practical design guide for balancing adaptation speed and stability.

Table 2: Accuracy (%) of ROSA across diverse domains. **Baseline** is standard multi-turn interaction. (+A+B) denotes update location (LM: LM Head, HS: Hidden States) and reward type (R: rule-based, M: model-based). Red values show absolute improvements. Further details on parameter updates and reward models are provided in Appendix C.5 and C.4, respectively.

		Mathematical Reasoning		General Reasoning		Multilingual Reasoning		Code Gen.
Model	**Method**	**MATH**	**MATH-500**	**MMLU-R**	**SuperGPQA**	**MT-AIME24**	**MT-MATH100**	**HumanEval**
Qwen2.5-0.5B -Instruct	Baseline	13.40	12.20	7.27	1.90	3.48	15.40	31.09
	ROSA (+LM + R)	**30.40** (+17.00)	28.00 (+15.80)	9.07 (+1.80)	5.63 (+3.73)	3.67 (+0.19)	22.80 (+7.40)	37.19 (+6.10)
	ROSA (+HS + R)	25.40 (+12.00)	25.00 (+12.80)	11.00 (+3.73)	5.00 (+3.10)	4.90 (+1.42)	20.90 (+5.50)	37.27 (+6.18)
	ROSA (+LM + M)	27.00 (+13.60)	**28.40** (+16.20)	**13.72** (+6.45)	**6.57** (+4.67)	**6.13** (+2.65)	**25.20** (+9.80)	**39.37** (+8.28)
Qwen3-0.6B	Baseline	25.00	26.00	18.60	4.20	4.80	31.30	41.46
	ROSA (+LM + R)	50.20 (+25.20)	51.60 (+25.60)	33.40 (+14.80)	9.13 (+4.93)	7.58 (+2.78)	56.60 (+25.30)	45.73 (+4.27)
	ROSA (+HS + R)	50.80 (+25.80)	50.60 (+24.60)	36.00 (+17.40)	9.70 (+5.50)	7.90 (+3.10)	51.90 (+20.60)	47.27 (+5.81)
	ROSA (+LM + M)	**52.20** (+27.20)	**54.60** (+28.60)	**40.68** (+22.08)	**15.73** (+11.53)	**9.43** (+4.63)	**59.40** (+28.10)	**49.37** (+7.91)
Qwen2.5-7B -Instruct	Baseline	47.00	49.40	45.36	19.31	19.24	60.34	57.92
	ROSA (+LM + R)	63.40 (+16.40)	62.40 (+13.00)	62.17 (+16.81)	37.26 (+17.95)	27.14 (+7.90)	73.16 (+12.82)	63.41 (+5.49)
	ROSA (+HS + R)	64.40 (+17.40)	63.40 (+14.00)	67.31 (+21.95)	36.27 (+16.96)	26.75 (+7.51)	72.27 (+11.93)	64.24 (+6.32)
	ROSA (+LM + M)	**65.20** (+18.20)	**65.60** (+16.20)	**68.47** (+23.11)	**40.67** (+21.36)	**30.21** (+10.97)	**75.13** (+14.79)	**67.36** (+9.44)
Qwen3-8B	Baseline	55.80	58.80	51.35	27.61	30.37	74.74	78.04
	ROSA (+LM + R)	65.80 (+10.00)	**72.80** (+14.00)	67.27 (+15.92)	36.11 (+8.50)	40.16 (+9.79)	85.16 (+10.42)	81.71 (+3.67)
	ROSA (+HS + R)	65.80 (+10.00)	66.20 (+7.40)	68.37 (+17.02)	37.73 (+10.12)	42.27 (+11.90)	86.93 (+12.19)	82.37 (+4.33)
	ROSA (+LM + M)	**67.40** (+11.60)	68.40 (+9.60)	**70.36** (+19.01)	**40.34** (+12.73)	**43.93** (+13.56)	**88.37** (+13.63)	**83.65** (+5.61)
DeepSeek-R1 -Distill-Llama-8B	Baseline	5.20	7.20	30.46	10.37	4.73	17.35	25.00
	ROSA (+LM + R)	7.80 (+2.60)	18.40 (+11.20)	41.14 (+10.68)	20.49 (+10.12)	6.13 (+1.40)	21.17 (+3.82)	39.03 (+14.03)
	ROSA (+HS + R)	8.40 (+3.20)	18.20 (+11.00)	42.18 (+11.72)	21.34 (+10.97)	7.27 (+2.54)	23.85 (+6.50)	38.37 (+13.37)
	ROSA (+LM + M)	**8.60** (+3.40)	**20.80** (+13.60)	**45.79** (+15.33)	**24.97** (+14.60)	**8.19** (+3.46)	**24.67** (+7.32)	**39.26** (+14.26)

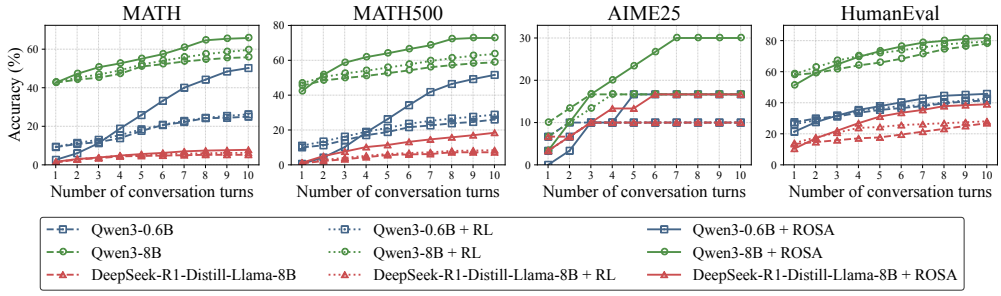

Figure 4: Comparison of *baselines*, *RL* (Appendix F.1), and *ROSA* across datasets. Unlike the slow improvement in Figure 2, ROSA yields steeper learning curves and higher absolute accuracy, demonstrating its superior efficiency in online error correction during multi-turn interactions.

5 Empirical Results

We validate the effectiveness and efficiency of ROSA in dynamic multi-turn settings, demonstrating state-of-the-art performance across diverse tasks (Section 5.1) and robust online error correction (Section 5.2). Setup details (datasets, baselines, evaluation protocol, feedback robustness) and ablations (optimization strategy, β) are detailed in Appendices C, C.6, C.7, and F.

Evaluation protocol. The primary verifiable-task evaluation allows up to 10 turns, stops at a verified correct answer, and resets adaptation between problems. Baseline and ROSA use the same textual-feedback protocol; ROSA additionally uses binary or model-based scalar rewards. Accuracy measures problems solved within the budget, and correction uplift measures recovery from first-turn failures. Full protocols and reward calibration are in Appendices C.6 and C.4.

5.1 Effectiveness and Generalizability Across Task Domains

To validate that ROSA effectively operationalizes T^{2}PAM, we evaluate it across diverse settings. T^{2}PAM demands dynamic, instance-specific adaptation via real-time feedback, and our results confirm ROSA fulfills this across three key dimensions:

Broad Applicability and Effectiveness. ROSA consistently outperforms standard multi-turn *baselines* across mathematical, general, code, and multilingual domains (Table 2; Appendix C.1, E.8). Correcting errors online, it overcomes traditional stagnation, matching or exceeding expensive multi-turn training (Appendix E.9) and advanced test-time compute baselines without massive latency (Appendix E.2), while scaling robustly to massive foundation models (Appendix E.3).

Table 3: Comparison of Correction Uplift (%).

Model	Method	MATH	AIME25	MATH-500	HumanEval
Qwen2.5-0.5B-Instruct	Baseline	6.88	0.00	6.79	14.39
	ROSA	**25.48** (+18.60)	**6.67** (+6.67)	**24.05** (+17.26)	**26.09** (+11.70)
Qwen3-0.6B	Baseline	17.40	3.57	17.78	19.33
	ROSA	**48.87** (+31.47)	**16.67** (+13.10)	**51.31** (+33.53)	**31.01** (+11.68)
Qwen2.5-7B-Instruct	Baseline	12.54	3.57	13.65	28.57
	ROSA	**41.53** (+28.99)	**20.69** (+17.12)	**36.91** (+23.26)	**40.00** (+11.43)
Qwen3-8B	Baseline	23.00	7.41	24.54	47.83
	ROSA	**40.42** (+17.42)	**27.59** (+20.18)	**52.94** (+28.40)	**62.50** (+14.67)
DeepSeek-R1-Distill -Llama-8B	Baseline	4.05	3.57	6.45	15.49
	ROSA	**6.30** (+2.25)	**13.79** (+10.22)	**17.41** (+10.96)	**31.97** (+16.48)

Architectural and Signal Flexibility. Fulfilling T^{2}PAM's lightweight constraints, ROSA achieves significant improvements updating either the *LM Head* or *Hidden States* (Appendix C.5). It also adapts dynamically to varying feedback granularities: while dense *model-based* rewards or Process Reward Models (PRM, Appendix E.5; calibration in Appendix C.4) yield peak performance, even sparse *rule-based* rewards deliver substantial gains, corroborating Theorem 2.

Universal Preference Alignment. Crucially, T^{2}PAM aims to satisfy diverse user preferences. Our task-agnostic evaluations (Appendix E) show that ROSA translates distinct feedback into precise policy alignments. It maximizes objective reasoning accuracy given correctness feedback, and seamlessly aligns with stylistic or subjective human preferences: on ALOE, the mean Alignment Level improves from 3.78 to 4.40 (Appendix E.12, Figure 7); IFEval strict accuracy surges from 58.31% to 70.26%; and MT-Bench subjective alignment rises from 60.36 to 75.57 (Appendix E.1, Table 12). Even in delayed-reward scenarios like UI Agents (Appendix E.1), ROSA efficiently steers the model toward any user-defined acceptable response.

5.2 Effectiveness in Online Error Correction

A core claim of our work is that ROSA fundamentally transforms LLMs from passive context-processors into active, continuous learners during multi-turn interactions. While ROSA consistently matches or surpasses computationally heavy test-time scaling methods and expensive multi-turn training paradigms (Appendices E.2 and E.9), our primary focus is dissecting the dynamics of its in-conversation self-correction. Traditional prompt-based models suffer from sharply diminishing returns because they rarely recover in subsequent turns after an initial failure, which leads to a severe performance plateau (Figure 6, Appendix E.10). ROSA breaks this stagnation by translating failure feedback into targeted policy updates. As evidenced by the *Correction Uplift* metric (Table 3), our method sustains a significantly higher volume of newly solved problems deep into the conversation. This continuous correction generates significantly steeper learning curves compared to the slow, plateauing trajectories of zero-shot baselines or direct RL optimization (Figure 4).

Crucially, this empirical self-correction capability aligns perfectly with our theoretical guarantees in Theorem 3, demonstrating that the model progressively aligns with user preferences by effectively learning from consecutive failures. Because continuous online updates naturally raise concerns regarding operational feasibility and model degradation, we rigorously verify the robustness of our approach. Specifically, we demonstrate that ROSA remains highly stable over extended interaction horizons of up to 20 turns (Appendix E.4) and strictly preserves general knowledge without catastrophic forgetting (Appendix E.6). Furthermore, ROSA achieves these dynamic capabilities with remarkable practical viability. It requires negligible GPU memory overhead and rapidly overtakes static baselines in time-to-accuracy efficiency (Appendix D), with further insights provided through a detailed case study (Appendix G) and an analysis of multi-sample trade-offs (Appendix E.7).

6 Conclusions and Limitations

We address LLM performance degradation in multi-turn dialogues by introducing the T^{2}PAM paradigm and its practical algorithm, ROSA. ROSA enables efficient in-conversation self-correction via real-time parameter updates. While theoretically and empirically validated, ROSA is currently less effective on tasks heavily bottlenecked by the pre-trained knowledge of model.

AI USE STATEMENT

We have not used generative AI tools for any of the tasks whose disclosure is required—proposing or refining hypotheses, developing conceptual frameworks, designing methodology or experiments, implementing methods, processing or cleaning data, analysing or interpreting results, formulating or proving mathematical claims, or translation—all of which were carried out entirely by the authors. We did use generative AI tools for two tasks whose disclosure is recommended: creating and refining the figures in this paper, and copy-editing the manuscript to improve grammar and readability. All AI-assisted figures were verified against the raw experimental outputs by the authors, and all AI-suggested text edits were reviewed by an author and restricted to wording, with no change to the technical content, claims, or conclusions. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

AIME. AIME problems and solutions, 2025. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

Kendall E. Atkinson. *An Introduction to Numerical Analysis*. John Wiley and Sons, 2nd edition, 1988. ISBN 978-0-471-50023-0. Section 8.9.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, Wei Ye, Yue Zhang, Yi Chang, Philip S. Yu, Qiang Yang, and Xing Xie. A survey on evaluation of large language models, 2023. URL https://arxiv.org/abs/2307.03109.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Maximillian Chen, Ruoxi Sun, Tomas Pfister, and Sercan O Arik. Learning to clarify: Multi-turn conversations with action-based contrastive self-training. In *The Thirteenth International Conference on Learning Representations*, 2025a. URL https://openreview.net/forum?id=SIE6VFps9x.

Yongchao Chen, Yilun Hao, Yueying Liu, Yang Zhang, and Chuchu Fan. Codesteer: Symbolic-augmented language models via code/text guidance. In *Forty-second International Conference on Machine Learning*, 2025b. URL https://openreview.net/forum?id=ezna4V4zHs.

Zhipeng Chen, Kun Zhou, Beichen Zhang, Zheng Gong, Xin Zhao, and Ji-Rong Wen. ChatCoT: Tool-augmented chain-of-thought reasoning on chat-based large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 14777–14790, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.985. URL https://aclanthology.org/2023.findings-emnlp.985/.

Tianzhe Chu, Yuexiang Zhai, Jihan Yang, Shengbang Tong, Saining Xie, Dale Schuurmans, Quoc V Le, Sergey Levine, and Yi Ma. SFT memorizes, RL generalizes: A comparative study of foundation model post-training. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=dYur3yabMj.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tai, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. Scaling instruction-finetuned language models. *J. Mach. Learn. Res.*, 25(1), January 2024. ISSN 1532-4435.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Zixin Ding, Junyuan Hong, Zhan Shi, Jiachen T. Wang, Zinan Lin, Li Yin, Meng Liu, Zhangyang Wang, and Yuxin Chen. Scaling textual gradients via sampling-based momentum, 2025. URL https://arxiv.org/abs/2506.00400.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. Alphaedit: Null-space constrained model editing for language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HvSytvg3Jh.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024. URL https://arxiv.org/abs/2312.10997.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021a.

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. *NeurIPS*, 2021b.

Dan Hendrycks, Mantas Mazeika, Andy Zou, Sahil Patel, Christine Zhu, Jesus Navarro, Dawn Song, Bo Li, and Jacob Steinhardt. What would jiminy cricket do? towards agents that behave morally.

In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021c. URL https://openreview.net/forum?id=G1muTb5zuO7.

Christine Herlihy, Jennifer Neville, Tobias Schnabel, and Adith Swaminathan. On overcoming miscalibrated conversational priors in llm-based chatbots. In *Proceedings of the Fortieth Conference on Uncertainty in Artificial Intelligence*, UAI '24. JMLR.org, 2024.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Wenyang Hu, Yao Shu, Zongmin Yu, Zhaoxuan Wu, Xiangqiang Lin, Zhongxiang Dai, See-Kiong Ng, and Bryan Kian Hsiang Low. Localized zeroth-order prompt optimization. In *The Thirty-Eighth Conference on Neural Information Processing Systems* (**NeurIPS Spotlight**), 2024.

Yang Hu, Xingyu Zhang, Xueji Fang, Zhiyang Chen, Xiao Wang, Huatian Zhang, and Guojun Qi. Slot: Sample-specific language model optimization at test-time, 2025. URL https://arxiv.org/abs/2505.12392.

Robert Irvine, Douglas Boubert, Vyas Raina, Adian Liusie, Ziyi Zhu, Vineet Mudupalli, Aliaksei Korshuk, Zongyi Liu, Fritz Cremer, Valentin Assassi, Christie-Carol Beauchamp, Xiaoding Lu, Thomas Rialan, and William Beauchamp. Rewarding chatbots for real-world engagement with millions of users, 2023. URL https://arxiv.org/abs/2303.06135.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan A, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. DSPy: Compiling declarative language model calls into state-of-the-art pipelines. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=sY5N0zY5Od.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017. URL https://arxiv.org/abs/1412.6980.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, John D Co-Reyes, Avi Singh, Kate Baumli, Shariq Iqbal, Colton Bishop, Rebecca Roelofs, Lei M Zhang, Kay McKinney, Disha Shrivastava, Cosmin Paduraru, George Tucker, Doina Precup, Feryal Behbahani, and Aleksandra Faust. Training language models to self-correct via reinforcement learning, 2024. URL https://arxiv.org/abs/2409.12917.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. Llms get lost in multi-turn conversation, 2025. URL https://arxiv.org/abs/2505.06120.

Brian Lester, Rami Al-Rfou, and Noah Constant. The power of scale for parameter-efficient prompt tuning. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih (eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 3045–3059, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.243. URL https://aclanthology.org/2021.emnlp-main.243/.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Yafu Li, Xuyang Hu, Xiaoye Qu, Linjie Li, and Yu Cheng. Test-time preference optimization: On-the-fly alignment via iterative textual feedback. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=ArifAHrEVD.

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, 2025b. URL https://arxiv.org/abs/2504.04717.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

Bo Liu, Leon Guertler, Simon Yu, Zichen Liu, Penghui Qi, Daniel Balcells, Mickel Liu, Cheston Tan, Weiyan Shi, Min Lin, Wee Sun Lee, and Natasha Jaques. Spiral: Self-play on zero-sum games incentivizes reasoning via multi-agent multi-turn reinforcement learning, 2025. URL https://arxiv.org/abs/2506.24119.

Jinxin Liu, Hongyin Zhang, Zifeng Zhuang, Yachen Kang, Donglin Wang, and Bin Wang. Design from policies: Conservative test-time adaptation for offline policy optimization. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=jZYf1GxH1V.

Yu Meng, Mengzhou Xia, and Danqi Chen. SimPO: Simple preference optimization with a reference-free reward. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=3Tzcot1LKb.

Jiao Ou, Jiayu Wu, Che Liu, Fuzheng Zhang, Di Zhang, and Kun Gai. Inductive-deductive strategy reuse for multi-turn instructional dialogues. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17402–17431, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.964. URL https://aclanthology.org/2024.emnlp-main.964/.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc. ISBN 9781713871088.

Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. Recursive introspection: Teaching language model agents how to self-improve. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=DRC9pZwBwR.

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.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=HPuSIXJaa9.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=Ti67584b98.

Yi Ren and Danica J. Sutherland. Learning dynamics of LLM finetuning. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tPNHOoZFl9.

Lior Shani, Aviv Rosenberg, Asaf Cassel, Oran Lang, Daniele Calandriello, Avital Zipori, Hila Noga, Orgad Keller, Bilal Piot, Idan Szpektor, Avinatan Hassidim, Yossi Matias, and Remi Munos. Multi-turn reinforcement learning with preference human feedback. In *The Thirty-eighth Annual*

Conference on Neural Information Processing Systems, 2024. URL https://openreview.net/forum?id=rVSc3HIZS4.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys 25, pp. 12791297. ACM, March 2025. doi: 10.1145/3689031.3696075. URL http://dx.doi.org/10.1145/3689031.3696075.

Taiwei Shi, Zhuoer Wang, Longqi Yang, Ying-Chun Lin, Zexue He, Mengting Wan, Pei Zhou, Sujay Jauhar, Sihao Chen, Shan Xia, Hongfei Zhang, Jieyu Zhao, Xiaofeng Xu, Xia Song, and Jennifer Neville. Wildfeedback: Aligning llms with in-situ user interactions and feedback, 2025. URL https://arxiv.org/abs/2408.15549.

Wentao Shi, Mengqi Yuan, Junkang Wu, Qifan Wang, and Fuli Feng. Direct multi-turn preference optimization for language agents. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2312–2324, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.138. URL https://aclanthology.org/2024.emnlp-main.138/.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R Narasimhan, and Shunyu Yao. Reflexion: language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=vAElhFcKW6.

Yao Shu, Wenyang Hu, See-Kiong Ng, Bryan Kian Hsiang Low, and Fei Yu. Ferret: Federated full-parameter tuning at scale for large language models, 2024. URL https://openreview.net/forum?id=9H1uctBWgF.

Guijin Son, Jiwoo Hong, Hyunwoo Ko, and James Thorne. Linguistic generalizability of test-time scaling in mathematical reasoning. *arXiv preprint arXiv:2502.17407*, 2025.

Suvrit Sra, Sebastian Nowozin, and Stephen J Wright. *Optimization for machine learning*, pp. 351368. Mit Press, 2011.

Yuchong Sun, Che Liu, Kun Zhou, Jinwen Huang, Ruihua Song, Xin Zhao, Fuzheng Zhang, Di Zhang, and Kun Gai. Parrot: Enhancing multi-turn instruction following for large language models. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9729–9750, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.525. URL https://aclanthology.org/2024.acl-long.525/.

M-A-P Team, Xinrun Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, Kang Zhu, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, Kaixing Deng, Shuyue Guo, Shian Jia, Sichao Jiang, Yiyan Liao, Rui Li, Qinrui Li, Sirun Li, Yizhi Li, Yunwen Li, Dehua Ma, Yuansheng Ni, Haoran Que, Qiyao Wang, Zhoufutu Wen, Siwei Wu, Tianshun Xing, Ming Xu, Zhenzhu Yang, Zekun Moore Wang, Junting Zhou, Yuelin Bai, Xingyuan Bu, Chenglin Cai, Liang Chen, Yifan Chen, Chengtuo Cheng, Tianhao Cheng, Keyi Ding, Siming Huang, Yun Huang, Yaoru Li, Yizhe Li, Zhaoqun Li, Tianhao Liang, Chengdong Lin, Hongquan Lin, Yinghao Ma, Zhongyuan Peng, Zifan Peng, Qige Qi, Shi Qiu, Xingwei Qu, Yizhou Tan, Zili Wang, Chenqing Wang, Hao Wang, Yiya Wang, Yubo Wang, Jiajun Xu, Kexin Yang, Ruibin Yuan, Yuanhao Yue, Tianyang Zhan, Chun Zhang, Jingyang Zhang, Xiyue Zhang, Xingjian Zhang, Yue Zhang, Yongchi Zhao, Xiangyu Zheng, Chenghua Zhong, Yang Gao, Zhoujun Li, Dayiheng Liu, Qian Liu, Tianyu Liu, Shiwen Ni, Junran Peng, Yujia Qin, Wenbo Su, Guoyin Wang, Shi Wang, Jian Yang, Min Yang, Meng Cao, Xiang Yue, Zhaoxiang Zhang, Wangchunshu Zhou, Jiaheng Liu, Qunshu Lin, Wenhao Huang, and Ge Zhang. Supergpqa: Scaling llm evaluation across 285 graduate disciplines, 2025. URL https://arxiv.org/abs/2502.14739.

Emiel Van Miltenburg, Anouck Braggaar, Emmelyn Croes, Florian Kunneman, Christine Liebrecht, and Gabriella Martijn. Measure only what is measurable: towards conversation requirements for

evaluating task-oriented dialogue systems. In Ofir Arviv, Miruna Clinciu, Kaustubh Dhole, Rotem Dror, Sebastian Gehrmann, Eliya Habba, Itay Itzhak, Simon Mille, Yotam Perlitz, Enrico Santus, João Sedoc, Michal Shmueli Scheuer, Gabriel Stanovsky, and Oyvind Tafjord (eds.), *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM²)*, pp. 231–238, Vienna, Austria and virtual meeting, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-261-9. URL https://aclanthology.org/2025.gem-1.18/.

Xingyao Wang, Zihan Wang, Jiateng Liu, Yangyi Chen, Lifan Yuan, Hao Peng, and Heng Ji. MINT: Evaluating LLMs in multi-turn interaction with tools and language feedback. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=jp3gWrMuIZ.

Chenxing Wei, Yao Shu, Ying Tiffany He, and Fei Yu. Flexora: Flexible low-rank adaptation for large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14643–14682, Vienna, Austria, July 2025a. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.713. URL https://aclanthology.org/2025.acl-long.713/.

Chenxing Wei, Yao Shu, Mingwen Ou, Ying Tiffany He, and Fei Richard Yu. Paft: Prompt-agnostic fine-tuning, 2025b. URL https://arxiv.org/abs/2502.12859.

Chenxing Wei, Jiarui Yu, Ying Tiffany He, Hande Dong, Yao Shu, and Fei Yu. Redit: Reward dithering for improved LLM policy optimization. In *2nd Workshop on Models of Human Feedback for AI Alignment*, 2025c. URL https://openreview.net/forum?id=nDzBpkbxk7.

Shujin Wu, Yi R. Fung, Cheng Qian, Jeonghwan Kim, Dilek Hakkani-Tur, and Heng Ji. Aligning LLMs with individual preferences via interaction. In Owen Rambow, Leo Wanner, Marianna Apidianaki, Hend Al-Khalifa, Barbara Di Eugenio, and Steven Schockaert (eds.), *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 7648–7662, Abu Dhabi, UAE, January 2025. Association for Computational Linguistics. URL https://aclanthology.org/2025.coling-main.511/.

Yiran Wu, Feiran Jia, Shaokun Zhang, Hangyu Li, Erkang Zhu, Yue Wang, Yin Tat Lee, Richard Peng, Qingyun Wu, and Chi Wang. Mathchat: Converse to tackle challenging math problems with LLM agents. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*, 2024. URL https://openreview.net/forum?id=S7vIB7OGQe.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025a. URL https://arxiv.org/abs/2505.09388.

Chenyu Yang, Shiqian Su, Shi Liu, Xuan Dong, Yue Yu, Weijie Su, Xuehui Wang, Zhaoyang Liu, Jinguo Zhu, Hao Li, Wenhai Wang, Yu Qiao, Xizhou Zhu, and Jifeng Dai. Zerogui: Automating online gui learning at zero human cost, 2025b. URL https://arxiv.org/abs/2505.23762.

Yunzhi Yao, Peng Wang, Bozhong Tian, Siyuan Cheng, Zhoubo Li, Shumin Deng, Huajun Chen, and Ningyu Zhang. Editing large language models: Problems, methods, and opportunities. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 10222–10240, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.632. URL https://aclanthology.org/2023.emnlp-main.632/.

Zihao Yi, Jiarui Ouyang, Zhe Xu, Yuwen Liu, Tianhao Liao, Haohao Luo, and Ying Shen. A survey on recent advances in llm-based multi-turn dialogue systems, 2025. URL https://arxiv.org/abs/2402.18013.

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Pan Lu, Zhi Huang, Carlos Guestrin, and James Zou. Optimizing generative ai by backpropagating language model feedback. *Nature*, 639(8055):609–616, 2025.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. Evaluating large language models at evaluating instruction following. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=tr0KidwPLc.

Chen Zhang, Xinyi Dai, Yaxiong Wu, Qu Yang, Yasheng Wang, Ruiming Tang, and Yong Liu. A survey on multi-turn interaction capabilities of large language models, 2025a. URL https://arxiv.org/abs/2501.09959.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, Irwin King, Xue Liu, and Chen Ma. A survey on test-time scaling in large language models: What, how, where, and how well?, 2025b. URL https://arxiv.org/abs/2503.24235.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=uccHPGDlao.

Qinhao Zhou, Zihan Zhang, Xiang Xiang, Ke Wang, Yuchuan Wu, and Yongbin Li. Enhancing the general agent capabilities of low-paramter LLMs through tuning and multi-branch reasoning. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2922–2931, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-naacl.184. URL https://aclanthology.org/2024.findings-naacl.184/.

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, 2025. URL https://arxiv.org/abs/2503.15478.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. Ttrl: Test-time reinforcement learning, 2025. URL https://arxiv.org/abs/2504.16084.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. TTRL: Test-time reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=VuVhgEiu20.

A Related Work

Research on improving the multi-turn capabilities of LLMs has largely proceeded along three main fronts: in-context learning, fine-tuning with multi-turn data, and reinforcement learning.

In-Context Learning and Prompting Strategies. A prominent line of work enhances multi-turn performance without modifying model parameters by leveraging the context window to guide the model's reasoning (Ou et al., 2024; Sun et al., 2024). For instance, ChatCoT (Chen et al., 2023) models the chain-of-thought process as a multi-turn interaction to improve reasoning. Similarly, Reflexion (Shinn et al., 2023) refines model behavior by converting environmental feedback into textual summaries, which are appended to the context for subsequent turns. MathChat (Wu et al., 2024) extends this by introducing a user agent that can execute tools and inject the resulting feedback into the conversation. These approaches use textual context as the adaptation channel. Our method is complementary: it retains conversational feedback while using its scalar evaluation to construct an explicit policy target and update a small parameter subset.

Fine-Tuning with Multi-Turn Data. Another approach involves fine-tuning the model on datasets specifically designed to capture multi-turn dynamics (Wang et al., 2024; Zhou et al., 2024). For instance, WildChat (Shi et al., 2025) leverages live user feedback to automatically construct a preference dataset for subsequent fine-tuning. Addressing challenges within this domain, Codesteer (Chen et al., 2025b) identifies a "gradient cancellation" issue, where gradients from early turns can interfere with those from later, more informative ones, and mitigates this by up-weighting the loss from the final turns of the interaction. However, a key limitation of such offline SFT approaches is their potential insufficiency in cultivating robust self-correcting behavior (Li et al., 2025b; Yi et al., 2025). This challenge often stems from a distribution mismatch between the errors present in the training data and those produced by the model at inference time, as well as the risk of "behavioral collapse," where the model overfits to a narrow set of correction patterns.

Reinforcement Learning Approaches. Several methods employ reinforcement learning (RL) to teach models to self-improve over multiple rounds (Zhou et al., 2025; Liu et al., 2025). For instance, RISE (Qu et al., 2024) utilizes multi-round offline RL with reward supervision, applying a majority vote over candidate outputs at inference time. SCoRe (Kumar et al., 2024) adopts a two-stage process, first teaching the model to self-correct and then maximizing this capability via RL. Other works have explored multi-round group preference optimization by decomposing conversations into single-turn problems (Shi et al., 2024; Yang et al., 2025b). While these RL-based strategies can cultivate sophisticated, self-correcting behaviors, they often face significant challenges, including high computational costs and training instability, particularly when applied to long, multi-turn dialogue contexts.

Test-Time Adaptation and Positioning. Test-time optimization (Zhang et al., 2025b; Zuo et al., 2025; Chen et al., 2025a) also changes model behavior during inference. Our distinction is the combination of an interaction-specific scalar feedback signal, an analytically constructed KL-regularized policy target, and a one-step linearized parameter update. Prompting methods adapt through context; offline multi-turn learning transfers behavior learned before deployment; sampling-based methods spend compute exploring candidate outputs. These mechanisms are complementary, and none is excluded by the definition of user preference. ROSA focuses on lightweight adaptation within an ongoing interaction, with zero additional offline training cost; its online generation, reward evaluation, and update costs are reported separately in Appendix D.

B Proofs

B.1 Proof of Theorem 1

Proof. Fix $\mathbf{x}$ and write $p = \pi_{\theta_{k-1}}(\cdot \mid \mathbf{x})$. For $\beta > 0$, define

$$J(\pi) = \mathbb{E}_\pi[r(\mathbf{x}, \mathbf{y})] - \beta D_{\mathrm{KL}}(\pi \| p), \qquad Z_k(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{y}) \exp(r(\mathbf{x}, \mathbf{y})/\beta). \tag{11}$$

Assume $0 < Z_k(\mathbf{x}) < \infty$, and define $\pi^*_{\theta_k}(\mathbf{y} \mid \mathbf{x}) = p(\mathbf{y}) \exp(r(\mathbf{x}, \mathbf{y})/\beta)/Z_k(\mathbf{x})$. Then

$$-J(\pi) = \beta \mathbb{E}_\pi \left[\log \frac{\pi(\mathbf{y} \mid \mathbf{x})}{p(\mathbf{y}) \exp(r(\mathbf{x}, \mathbf{y})/\beta)} \right] \tag{12}$$

$$= \beta D_{\mathrm{KL}}(\pi \| \pi^*_{\theta_k}) - \beta \log Z_k(\mathbf{x}). \tag{13}$$

Thus $\arg\max_\pi J(\pi) = \arg\min_\pi D_{\mathrm{KL}}(\pi \| \pi^*_{\theta_k})$. The objective equality retains both β and the normalization constant; dropping them is valid only when identifying the optimizer. Over distributions supported on p, the minimum KL divergence is zero, attained by $\pi^*_{\theta_k}$. This distribution-space optimum need not be exactly realizable by a restricted parameter update; Theorem 4 accounts for approximation. □

B.2 Derivation of Equation 3

Definition 1 (Single-Sample Feedback Constraint)**.** *In practical applications, feedback is typically received for only a single generated response, $\mathbf{y}_k$. We define the turn-local surrogate reward $\tilde{r}_k$, rather than assume that unobserved responses have zero true user utility:*

$$\tilde{r}_k(\mathbf{x}, \mathbf{y}) = r_k \cdot \mathbb{I}(\mathbf{y} = \mathbf{y}_k) = \begin{cases} r_k, & \text{if } \mathbf{y} = \mathbf{y}_k \\ 0, & \text{if } \mathbf{y} \neq \mathbf{y}_k \end{cases} \tag{14}$$

Derivation of the Practical Target from the Theoretical Optimum. Our goal is to derive the practical, single-sample update target (Equation 3) and its corresponding partition function from the general theoretical optimal policy (Equation 2) under the Single-Sample Feedback Constraint (Definition 1).

1. Derivation of the Practical Target Policy $\tilde{\pi}^*_{\theta_k}$. We substitute the surrogate reward from Definition 1 into the general policy formula from Equation 2. This naturally yields a piecewise expression:

- For the observed response where $\mathbf{y} = \mathbf{y}_k$, the reward is r_k, yielding:

$$\tilde{\pi}^*_{\theta_k}(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_k(\mathbf{x})} \pi_{\theta_{k-1}}(\mathbf{y}|\mathbf{x}) \exp\left(\tfrac{1}{\beta} r_k\right) \tag{15}$$

- For all other responses where $\mathbf{y} \neq \mathbf{y}_k$, the reward is 0, yielding:

$$\tilde{\pi}^*_{\theta_k}(\mathbf{y}|\mathbf{x}) = \frac{1}{Z_k(\mathbf{x})} \pi_{\theta_{k-1}}(\mathbf{y}|\mathbf{x}) \exp(0) = \frac{1}{Z_k(\mathbf{x})} \pi_{\theta_{k-1}}(\mathbf{y}|\mathbf{x}) \tag{16}$$

Combining these two results gives the piecewise form in Equation 3.

2. Derivation of the Practical Partition Function $Z_k(\mathbf{x})$. Next, we apply the same constrained reward to the general partition function definition by splitting the sum over the entire response space $\mathcal{Y}$:

$$\begin{aligned} Z_k(\mathbf{x}) &= \sum_{\mathbf{y}' \in \mathcal{Y}} \pi_{\theta_{k-1}}(\mathbf{y}'|\mathbf{x}) \exp\left(\frac{1}{\beta} r_k \cdot \mathbb{I}(\mathbf{y}' = \mathbf{y}_k)\right) \\ &= \pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x}) \exp\left(\frac{1}{\beta} r_k\right) + \sum_{\mathbf{y}' \neq \mathbf{y}_k} \pi_{\theta_{k-1}}(\mathbf{y}'|\mathbf{x}) \exp(0) \\ &= \pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x}) \exp\left(\frac{1}{\beta} r_k\right) + \left(1 - \pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x})\right) \\ &= 1 - \left(1 - \exp\left(\frac{1}{\beta} r_k\right)\right) \pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x}) \end{aligned}$$

This confirms the expression for the practical partition function used in Equation 3. □

B.3 Proof of Theorem 2

Proof. Fix $\mathbf{x}$, let $P = \pi^*_{\text{user}}$ and $p_k = \pi_{\theta_k}$, and set $q_k = \tilde{\pi}^*_{\theta_k}$. By definition, $P(\mathcal{A}) = 1$. Reliable rejection gives $\mathbf{y}_k \notin \mathcal{A}$ and hence $P(\mathbf{y}_k) = 0$; this feedback-reliability condition is distinct from the definition of the ideal policy. The target uses the turn-local surrogate, with $r_k = -1$:

$$q_k(\mathbf{y}) = \frac{p_{k-1}(\mathbf{y}) \exp[-\mathbb{I}(\mathbf{y} = \mathbf{y}_k)/\beta]}{Z_k}, \qquad Z_k = 1 - C p_{k-1}(\mathbf{y}_k), \quad C = 1 - e^{-1/\beta}. \tag{17}$$

Consequently,

$$D_{\mathrm{KL}}(P\|q_k) - D_{\mathrm{KL}}(P\|p_{k-1}) = \mathbb{E}_P \log \frac{p_{k-1}}{q_k} = \log Z_k + \frac{P(\mathbf{y}_k)}{\beta} \tag{18}$$

$$= \log(1 - C p_{k-1}(\mathbf{y}_k)) \le -C p_{k-1}(\mathbf{y}_k) < 0. \tag{19}$$

With exact updates, $p_{k-1} = q_{k-1}$ (and $q_0 = p_0$), giving the stated result. This is a guarantee for reliable rejection of unacceptable outputs; a noisy negative score on an acceptable output does not satisfy its premise. □

B.4 Proof of Theorem 3

Proof of Theorem 3. We want to bound the final estimation error after K turns, $D_{\mathrm{KL}}(\pi^*_{\text{user}}\|\tilde{\pi}^*_{\theta_K})$. We can express this final error as the initial error at turn 0 plus the sum of all one-step changes in error from turn 1 to K:

$$D_{\mathrm{KL}}(\pi^*_{\text{user}}\|\tilde{\pi}^*_{\theta_K}) = D_{\mathrm{KL}}(\pi^*_{\text{user}}\|\pi_{\theta_0}) + \sum_{k=1}^{K} \Big(D_{\mathrm{KL}}(\pi^*_{\text{user}}\|\tilde{\pi}^*_{\theta_k}) - D_{\mathrm{KL}}(\pi^*_{\text{user}}\|\tilde{\pi}^*_{\theta_{k-1}}) \Big) \tag{20}$$

where we define $\tilde{\pi}^*_{\theta_0} = \pi_{\theta_0}$ as the initial policy.

From Theorem 2, we have an upper bound for each one-step change in error, where $C = 1 - \exp(-1/\beta)$:

$$D_{\mathrm{KL}}(\pi^*_{\text{user}}\|\tilde{\pi}^*_{\theta_k}) - D_{\mathrm{KL}}(\pi^*_{\text{user}}\|\tilde{\pi}^*_{\theta_{k-1}}) \le -C \cdot \pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x}) \tag{21}$$

We can apply this inequality to the summation term. By summing the upper bounds for each step from $k = 1$ to K, we get an upper bound for the total change:

$$\sum_{k=1}^{K} \Big(D_{\mathrm{KL}}(\pi^*_{\text{user}}\|\tilde{\pi}^*_{\theta_k}) - D_{\mathrm{KL}}(\pi^*_{\text{user}}\|\tilde{\pi}^*_{\theta_{k-1}}) \Big) \le \sum_{k=1}^{K} \big(-C \cdot \pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x}) \big) \tag{22}$$

Substituting this bounded sum back into our expression for the final error, we arrive at the desired result:

$$D_{\mathrm{KL}}(\pi^*_{\text{user}}\|\tilde{\pi}^*_{\theta_K}) \le D_{\mathrm{KL}}(\pi^*_{\text{user}}\|\pi_{\theta_0}) - C \sum_{k=1}^{K} \pi_{\theta_{k-1}}(\mathbf{y}_k|\mathbf{x}) \tag{23}$$

This completes the proof. □

Sample efficiency under diverse failure modes. Let $D_0 = D_{\mathrm{KL}}(P\|p_0) < \infty$. If a sequence of K reliably rejected-response updates satisfies $p_{k-1}(\mathbf{y}_k) \ge p_{\min} > 0$, the exact-update bound gives

$$D_K \le D_0 - CKp_{\min}. \tag{24}$$

Conditionally on such a sequence, $K \ge \lceil (D_0 - \eta)/(Cp_{\min}) \rceil$ suffices for $D_K \le \eta$, for $0 < \eta < D_0$. This is a finite-run conditional bound, not an unconditional sampling-complexity guarantee: the lower bound on rejected-response probability need not persist, and interactions can stop after success. High-confidence recurrent errors yield larger one-step gains; diffuse, low-probability failures may yield smaller gains. With approximate updates, replace $Cp_{\min}$ by a positive lower bound γ on $Cp_{k-1}(\mathbf{y}_k) - \epsilon_k - L\|\Delta\theta_k\|_2^2/2$, giving $D_K \le D_0 - K\gamma$ over the corresponding sequence. Empirical analyses of longer horizons and multiple feedback samples are provided in Appendices E.4 and E.7.

Acceptability versus a unique ideal distribution. Let $a = p_{k-1}(\mathcal{A}) > 0$ and $b = 1 - a$. An exact rejection update preserves relative probabilities within $\mathcal{A}$ and yields $q_k(\mathcal{A}) = a/Z_k$, so

$$b - q_k(\mathcal{A}^c) = \frac{aCp_{k-1}(\mathbf{y}_k)}{1 - Cp_{k-1}(\mathbf{y}_k)} > 0. \tag{25}$$

This is the relevant improvement in unacceptable-answer probability. Rejection alone does not identify an arbitrary preferred distribution among acceptable answers. In particular, if all unacceptable mass vanishes, the remaining distribution is the initial policy conditioned on $\mathcal{A}$, unless additional feedback distinguishes acceptable answers.

B.5 Proof of Theorem 4

Assumption 1 (Lipschitz-Smooth Log-Policy)**.** *For every* $\mathbf{y}$ *on the support of* $P = \pi^*_{\text{user}}$*, the log-policy gradient is* L*-Lipschitz along the segment from* θ_{k-1} *to* θ_k*. All expectations below are finite. With* $\Delta\theta_k = \theta_k - \theta_{k-1}$*,*

$$\log p_k(\mathbf{y}) = \log p_{k-1}(\mathbf{y}) + \nabla \log p_{k-1}(\mathbf{y})^\top \Delta\theta_k + R_k(\mathbf{y}), \quad |R_k(\mathbf{y})| \le \frac{L}{2}\|\Delta\theta_k\|_2^2. \tag{26}$$

Linearized target-fitting error. Write $p = p_{k-1}$, $q = q_k = \tilde{\pi}^*_{\theta_k}$, and define

$$e_k(\mathbf{y}) = p(\mathbf{y}) + \nabla p(\mathbf{y})^\top \Delta\theta_k - q(\mathbf{y}), \qquad \epsilon_k = \mathbb{E}_{\mathbf{y}\sim P}\frac{|e_k(\mathbf{y})|}{p(\mathbf{y})} < \infty. \tag{27}$$

Thus ϵ_k measures mismatch between the linearized policy prediction and the desired target, whereas $L\|\Delta\theta_k\|_2^2/2$ controls the log-policy Taylor remainder. Neither error is discarded.

Proof. Decompose each change in divergence as

$$D_{\mathrm{KL}}(P\|p_k) - D_{\mathrm{KL}}(P\|p) = \underbrace{D_{\mathrm{KL}}(P\|q) - D_{\mathrm{KL}}(P\|p)}_{A_k} \tag{28}$$

$$+ \underbrace{D_{\mathrm{KL}}(P\|p_k) - D_{\mathrm{KL}}(P\|q)}_{B_k}. \tag{29}$$

Bounding Term A (Improvement). Reliable rejection and the definition of P give

$$A_k = \log Z_k \le -Cp(\mathbf{y}_k), \qquad Z_k = 1 - Cp(\mathbf{y}_k). \tag{30}$$

Bounding Term B (Approximation Error). On the support of P, $q(\mathbf{y}) = p(\mathbf{y})/Z_k$, and Equation equation 27 implies

$$\nabla \log p(\mathbf{y})^\top \Delta\theta_k = \frac{1}{Z_k} - 1 + \frac{e_k(\mathbf{y})}{p(\mathbf{y})}. \tag{31}$$

Consequently, applying the log-policy expansion rather than a triangle inequality for KL divergence yields

$$\begin{aligned} B_k &= \mathbb{E}_P \log \frac{q}{p_k} \\ &= -\log Z_k - \left(\frac{1}{Z_k} - 1\right) - \mathbb{E}_P \frac{e_k}{p} - \mathbb{E}_P R_k \\ &\le -\log Z_k - \left(\frac{1}{Z_k} - 1\right) + \epsilon_k + \frac{L}{2}\|\Delta\theta_k\|_2^2 \\ &\le \epsilon_k + \frac{L}{2}\|\Delta\theta_k\|_2^2. \end{aligned} \tag{32}$$

The last inequality follows from $\log u \le u - 1$ with $u = 1/Z_k$. Summing $A_k + B_k$ gives

$$D_{\mathrm{KL}}(P\|p_K) \le D_{\mathrm{KL}}(P\|p_0) - C\sum_{k=1}^{K} p_{k-1}(\mathbf{y}_k) + \sum_{k=1}^{K}\epsilon_k + \frac{L}{2}\sum_{k=1}^{K}\|\Delta\theta_k\|_2^2. \tag{33}$$

□

Interpretation of approximation control. Equation equation 3 controls the requested change in probability through the rejected answer's mass. Translating a small target change into a small parameter update additionally depends on the conditioning of the linearized solve; the probability alone is not a parameter-norm bound. CG reduces the numerical residual of that solve. A small normal-equation residual alone does not bound the full P-weighted residual in Equation equation 27: target realizability and appropriate conditioning are also needed. Under these conditions, accurate CG solves support small ϵ_k. The positive-net-progress requirement is $Cp_{k-1}(\mathbf{y}_k) > \epsilon_k + L\|\Delta\theta_k\|_2^2/2$. The empirical results support practical effectiveness but do not establish this inequality at every update.

C Experimental setting

We conduct a comprehensive evaluation of ROSA across a diverse set of tasks and models to validate its generalizability, effectiveness, and efficiency.

C.1 Datasets.

To demonstrate the broad applicability of ROSA, we select challenging benchmarks spanning four distinct problem-solving domains. A summary of these datasets is provided in Table 4, followed by detailed descriptions.

Table 4: Overview of the datasets used for evaluation. "N/A" indicates that the dataset is primarily for evaluation and does not have a standard, predefined training set.

Domain	Dataset Name	Training Size	Test Size
Mathematical Reasoning	`MATH`	7,500	5,000
	`AIME25`	N/A	30
	`MATH-500`	N/A	500
General Reasoning	`GPQA-diamond`	N/A	198
	`MMLU-Redux`	N/A	3,000
	`SuperGPQA`	26,500	N/A
Code Generation	`HumanEval`	N/A	164
Multilingual Reasoning	`MCLM`	N/A	156

Mathematical Reasoning. This domain focuses on complex, multi-step mathematical problem-solving. We use three standard benchmarks. `MATH` (Hendrycks et al., 2021b) is a dataset of 12,500 challenging competition mathematics problems from high school level, covering topics like algebra, geometry, and calculus. `AIME25` (AIME, 2025) is a curated set of 25 highly difficult problems from the American Invitational Mathematics Examination (AIME), designed to test advanced reasoning capabilities. `MATH-500` (Lightman et al., 2023) is a well-known evaluation subset of the `MATH` test set, consisting of 500 problems often used for efficient model assessment.

General Reasoning. To evaluate reasoning on a broad range of topics, we use three expert-level question-answering datasets. `GPQA-diamond` (Rein et al., 2024) is a challenging set of graduate-level, Google-proof questions written by domain experts, where the "diamond" subset represents the highest-quality questions. `MMLU-Redux` (Hendrycks et al., 2021a) is a revised and cleaned version of the Massive Multitask Language Understanding benchmark, which covers 57 diverse subjects from elementary mathematics to US history and law. `SuperGPQA` (Team et al., 2025) significantly expands upon GPQA, containing nearly 5,000 expert-validated questions across 285 graduate-level disciplines.

Code Generation. We test the ability of models to generate functionally correct code from natural language descriptions using `HumanEval` (Chen et al., 2021). This dataset consists of 164 hand-written programming problems with function signatures, docstrings, and unit tests to verify the correctness of the generated code.

Multilingual Reasoning. To assess reasoning capabilities across different languages, we use the MCLM (Son et al., 2025) benchmark. This dataset was created by translating challenging English reasoning benchmarks into multiple languages. Our evaluation focuses on its subsets, including multilingual versions of IMO, AIME, and MATH problems (M-IMO, MT-AIME24, and MT-MATH100).

Dataset Usage in Experiments. Our primary evaluation of effectiveness of ROSA is conducted on official, held-out test sets to simulate real-world performance. For experiments where a dedicated test set is not available, or for ablation studies, we utilize the corresponding training or development sets for analysis. This ensures a comprehensive assessment of ROSA capabilities across different conditions while maintaining a clear distinction between final evaluation and component analysis. Specifically, we only sample part of the data from the SuperGPQA training set for testing, and the rest of the data sets are tested on the test set.

C.2 MODELS

Our model selection separates breadth from mechanism analysis: cross-model experiments test generality, while most controlled MATH diagnostics use Qwen3-0.6B. ALOE and successful-trajectory experiments use Qwen3-8B and are reported separately; results across these settings are not pooled. Our evaluation includes a variety of recent open-source LLMs to ensure our findings are not model-specific. These models are selected to cover a range of sizes and specializations, as summarized in Table 5 and detailed below. To mitigate potential data contamination issues with the Qwen2.5 series on certain benchmarks, we also conduct validation experiments on the more recent Qwen3 and DeepSeek-R1 models. All models used are instruction-tuned variants designed for chat and instruction-following tasks.

Table 5: Overview of the language models used in our experiments, categorized by scale and specialization.

Category	Model Name	Parameters	Variant
Small-Scale Models	Qwen2.5-0.5B-Instruct	0.5B	Instruct
	Qwen3-0.6B	0.6B	Base
Large-Scale Models	Qwen2.5-7B-Instruct	7B	Instruct
	Qwen3-8B	8B	Base
Reasoning-Focused	DeepSeek-R1-Distill-Llama-8B	8B	Reasoning-Tuned
	DeepSeek-R1-Distill-Qwen-7B	7B	Reasoning-Tuned

Small-Scale Models. To assess the performance of ROSA on more compact models, we selected two from the Qwen family, known for their strong general-purpose capabilities. Qwen2.5-0.5B-Instruct (Qwen et al., 2025) is a 0.5 billion parameter model from the Qwen2.5 series, optimized for instruction following. Qwen3-0.6B (Yang et al., 2025a) is a 0.6 billion parameter model from the newer Qwen3 generation, featuring architectural improvements.

Large-Scale Models. We evaluate on larger, more capable base models to test the scalability of our approach. These include Qwen2.5-7B-Instruct (Qwen et al., 2025), a widely-used 7 billion parameter instruction-tuned model, and Qwen3-8B (Yang et al., 2025a), its 8 billion parameter successor from the Qwen3 series.

Reasoning-Focused Models. To specifically test performance on complex reasoning, we use models from the DeepSeek-R1 series, which are explicitly optimized for reasoning capabilities through reinforcement learning (DeepSeek-AI et al., 2025). The models we use are distilled versions of a larger, proprietary model. DeepSeek-R1-Distill-Llama-8B is an 8 billion parameter model that uses a Llama-based architecture. DeepSeek-R1-Distill-Qwen-7B is a 7 billion parameter variant that is instead based on the Qwen architecture, allowing for a more controlled comparison with the general-purpose Qwen models.

C.3 Evaluation Metrics

We assess ROSA based on two primary aspects: performance and efficiency.

Performance Metrics. To measure problem-solving success, we define two key metrics. *Accuracy* is the final proportion of unique problems solved correctly within a total of K conversational turns. Let $\mathcal{P}$ be the set of all problems, and let $S_i \in \{0, 1\}$ be an indicator variable where $S_i = 1$ if problem i is solved at any turn up to K. The accuracy is given by:

$$\text{Accuracy} = \frac{\sum_{i \in \mathcal{P}} S_i}{|\mathcal{P}|} \tag{34}$$

Correction Uplift measures the ability of model to recover from initial failures. It is the percentage of problems that were answered incorrectly in the first turn but were successfully corrected in a subsequent turn. Let $\mathcal{P}_{\text{fail}} \subset \mathcal{P}$ be the subset of problems that the model failed to solve in the first turn. The Correction Uplift is:

$$\text{Correction Uplift} = \frac{\sum_{i \in \mathcal{P}_{\text{fail}}} S_i}{|\mathcal{P}_{\text{fail}}|} \times 100\% \tag{35}$$

Efficiency Metrics. To evaluate the computational overhead of our method, we measure two metrics. *Latency* is the average wall-clock time required for a single generation and update cycle. *Peak GPU Memory* is the maximum GPU memory consumed during this cycle. These metrics are crucial for assessing the practical feasibility of deploying ROSA in real-world interactive systems.

C.4 Reward Models

To simulate different real-world feedback scenarios, we employ two types of reward models.

Rule-Based Reward Model. This model simulates scenarios with definitive, high-level judgments by providing a sparse feedback signal of $\{-1, +1\}$. It programmatically extracts the final answer from a model's response, typically from a `\boxed{}` environment, and compares it to the ground-truth solution. A reward of $+1.0$ is assigned for a correct answer, and -1.0 otherwise. This mimics situations where feedback is based solely on the final outcome. The core logic implementation is shown in the following table.

Core logic for the rule-based reward model.

```
class MathVerifyRewardModel:
    def __init__(self, ground_truth_answer: str):
        self.ground_truth_answer = ground_truth_answer

    def get_reward(self, response_text: str) -> float:

        return 1.0 if compute_score(response_text,
        self.ground_truth_answer) == 1.0 else -1.0

def compute_score(solution_str, ground_truth) -> float:
    retval = 0.0
    try:
        string_in_last_boxed =
        last_boxed_only_string(solution_str)
        if string_in_last_boxed is not None:
            answer = remove_boxed(string_in_last_boxed)
            if is_equiv(answer, ground_truth):
                retval = 1.0
    except Exception:
        pass
    return retval
```

Model-Based Reward Model. This model simulates more nuanced, fine-grained human feedback by providing a dense, continuous reward score in the range $[-1.0, +1.0]$. We use the large language model, `Qwen/Qwen3-30B-A3B-Instruct-2507`, as the reward judge. The model is deployed using the VLLM inference engine for efficient scoring. It evaluates the generated response based on correctness, reasoning, and style by comparing it against the problem statement and the ideal solution. The prompt used to elicit the score is shown in the following table.

The prompt template for the model-based reward system.

```
A student AI was asked the following problem: {problem}.
The student AI gave the following answer: {generated_text}.
The ideal correct solution and answer is {solution}.
Please grade strictly but fairly.
Compare the student's answer to the ideal answer.
Evaluate the student's answer based on correctness, reasoning, and style.
Note: Based on your evaluation, please provide a floating point score
from -1.0 (completely wrong) to 1.0 (perfect).
The score should be placed at the end of your answer in the format: SCORE: [score].
```

Reward rubric and calibration. We additionally evaluate a discrete reasoning-aware rubric using GPT-5.6. Given the question, generated response, reference reasoning, and reference answer, the judge assigns one of five scores using Table 6. The final-answer verifier separately determines task success: a positive partial-credit score can trigger an update without terminating an incorrect attempt. This supplementary rubric study is distinct from the original Qwen3-30B-A3B outcome judge and the GPT-5.2 PRM evaluation. For example, consider a triangle with side lengths 7, 10, and x^2,

Table 6: Five-level reward rubric.

Score	Criterion
$+1.0$	Correct reasoning and correct final answer.
$+0.5$	Essentially correct method with a local arithmetic or presentation error.
$+0.1$	Partially correct core idea, but a critical step is missing or flawed.
-0.5	Limited valid progress; the main reasoning is incorrect.
-1.0	Incorrect or irrelevant reasoning and answer.

where x is a positive integer. The valid values are $x \in \{2, 3, 4\}$ because $3 < x^2 < 17$. A correct derivation and answer receives $+1$; correctly deriving these values but transcribing the final set as $\{2, 3, 5\}$ receives $+0.5$; using only $x^2 < 17$ and returning $\{1, 2, 3, 4\}$ receives $+0.1$; reversing the triangle inequality to $17 \leq x^2$ receives -0.5; simply adding $7 + 10$ and answering $x = 17$ receives -1.

For calibration, five annotators independently rated 50 responses to 50 MATH problems, with 10 responses sampled from each GPT-5.6 score level. They received the same rubric, problem, generated response, reference reasoning, and reference answer. Pairwise exact agreement was 93.6%, all-five agreement was 86.0%, and Fleiss' κ was 0.920. GPT-5.6 agreed with individual annotators on 96.4% of ratings and with the majority on all 50 responses; mean Cohen's κ was 0.955. These figures describe this small, score-stratified mathematical-reasoning calibration set, rather than population-wide reliability or calibration of the separate ALOE judge. Downstream rubric results are reported in Table 16.

C.5 Parameter Update Mechanisms

To implement the policy update $\Delta\theta$ computed in Section 3.3, we introduce two distinct, lightweight update mechanisms. These methods are designed to be computationally efficient, allowing for real-time policy adaptation during the inference phase without significant overhead.

1. LM Head Update via LoRA. The first mode targets the final layer of the model, the language modeling (LM) head. The LM head is typically a linear layer (an MLP matrix) that projects the final hidden state representation of the model into the vocabulary space to produce logits. We augment

this layer by adding a Low-Rank Adaptation (LoRA) (Hu et al., 2022) matrix. Specifically, a low-rank decomposition, represented by two matrices $\mathbf{A} \in \mathbb{R}^{d\times r}$ and $\mathbf{B} \in \mathbb{R}^{r\times V}$ (where d is the hidden size, V is the vocabulary size, and $r \ll d, V$ is the rank), is added to the original LM head weight matrix. During our online update process, only the parameters of these small LoRA matrices $\mathbf{A}$ and $\mathbf{B}$ are modified. The parameter update $\Delta\theta$ calculated by the CG solver is applied directly to the flattened weights of $\mathbf{A}$ and $\mathbf{B}$. This approach confines the policy optimization to a very small subset of the total model parameters, preserving the model's foundational knowledge while enabling rapid and efficient adaptation of its final output probabilities. The specific LoRA configuration is shown in Table 7.

Table 7: LoRA Hyperparameter Configuration.

Hyperparameter	Value
`target_modules`	lm_head
`Rank`	1
`lora_alpha`	8
`lora_dropout`	0.1

2. Hidden State Modification. The second mode operates not on the model's weights, but directly on its activations (Hu et al., 2025). Instead of modifying a layer, we intercept the final hidden state $\mathbf{H} \in \mathbb{R}^{1\times d}$ just before it is passed to the LM head. We then compute an update vector $\Delta\mathbf{H} \in \mathbb{R}^{1\times d}$ (which in this context represents our $\Delta\theta$) and add it directly to the hidden state to produce a modified activation:

$$\mathbf{H}_{\text{new}} = \mathbf{H} + \Delta\mathbf{H} \tag{36}$$

This new hidden state, $\mathbf{H}_{\text{new}}$, is then passed to the original, unmodified LM head to generate the final logits. This method is implemented using model hooking techniques, which allow us to register a forward hook on the LM head layer. The hook intercepts the input ($\mathbf{H}$), applies the additive modification, and returns the transformed tensor as the new input for the layer's forward pass. This approach completely avoids any updates to the persistent model weights and instead performs a transient, state-dependent policy correction on the activation flow.

C.6 Multi-Turn Evaluation Protocol

To ensure reproducibility and a strictly fair comparison between our proposed ROSA framework and all baseline methods, we standardize the multi-turn interaction construction process across all datasets. The primary verifiable-task evaluation uses at most 10 turns, stops upon verified success, and resets adaptation before the next problem. Its interaction construction follows these steps:

- **Initial Turn:** The interaction begins by providing the exact same initial task prompt to all evaluated methods.
- **Simulated User Feedback:** If the model generates an incorrect answer, we programmatically provide a standardized, simulated user feedback message. For instance, the default template used in our primary evaluation is: *"Wrong answer, please rethink and try another way of thinking!"*.
- **Context Appending:** For each subsequent turn, the model's previous response and the corresponding user feedback are appended to the conversational history. This growing context is then used as the input for the next generation step.

Fairness and Consistency: It is critical to emphasize that both the baseline models and ROSA receive the exact same initial prompts and simulated feedback messages throughout the evaluation. The only variation in the provided context stems naturally from the different intermediate responses generated by the models themselves across turns. This controls the textual-feedback construction. The scalar reward used for adaptation is specified separately in Appendix C.4.

C.7 Robustness to Feedback Templates

A common challenge in standard prompt-based multi-turn interactions is the high sensitivity of Large Language Models to the specific phrasing of the feedback. To verify that the performance gains of ROSA are fundamentally rooted in its dynamic parameter adaptation rather than artifacts of a specific prompt design, we evaluate its robustness across varying feedback formulations.

We conducted this evaluation on the MATH dataset using the Qwen3-0.6B model. In addition to the default feedback template used in our primary experiments, we tested two alternative templates designed to simulate different user interaction styles:

- **Template A (Default):** The standard conversational feedback used across our main evaluation (e.g., *"Wrong answer, please rethink and try another way of thinking!"*).
- **Template B (Direct):** A concise and direct notification of failure (e.g., *"Incorrect"*).
- **Template C (Guiding):** A slightly more encouraging and guiding prompt formulation (e.g., *"That doesn't seem quite right. Could you review your previous steps and see if there's an alternative approach?"*).

Table 8: Robustness to different feedback templates on the MATH dataset using Qwen3-0.6B. The baseline accuracy without ROSA is 25.00%.

Method	Template A (Default)	Template B (Direct)	Template C (Guiding)
ROSA (+LM+R)	50.20%	48.40%	52.60%
ROSA (+HS+R)	50.80%	50.20%	51.60%
ROSA (+LM+M)	52.20%	53.60%	54.40%

As demonstrated in Table 8, ROSA consistently and significantly outperforms the standard baseline across all evaluated templates. The variation in accuracy among the different templates is relatively small, particularly for the model-based reward variant (+LM+M), which remains highly stable across different phrasing. This empirical evidence confirms that the self-correction capability of ROSA is highly robust to the syntactic variations of user feedback, validating our claim that the effectiveness of the method stems from its test-time policy optimization mechanism rather than superficial prompt engineering.

D Computational Efficiency Analysis

D.1 Time-to-Accuracy Efficiency

A potential concern with test-time adaptation is the computational overhead introduced by the online parameter updates. Indeed, ROSA requires calculating gradients and updating the model at intermediate steps, which theoretically increases the latency per turn. To rigorously evaluate this trade-off, we present the Time-to-Accuracy curves in Figure 5. As illustrated, in the initial phase of interaction, ROSA (solid lines) may exhibit a lower accuracy compared to the baseline (dashed lines) for the same elapsed time. This initial lag is the direct consequence of the additional computational cost required for gradient calculation and backpropagation, which extends the duration of each turn.

However, as the interaction progresses, the benefits of ROSA become evident. By dynamically modifying model parameters to align with user preferences, ROSA corrects errors more effectively, resulting in a significantly steeper learning curve. Consequently, ROSA rapidly overtakes the baseline, achieving a substantially higher accuracy within the same total wall-clock time. This crossover demonstrates that despite the per-turn latency overhead, the superior error-correction rate of ROSA makes it more time-efficient in the long run for solving complex problems.

D.2 Memory and Latency Overhead

In this section, we analyze the absolute computational overhead of ROSA in terms of latency and spatial complexity. ROSA introduces an explicit parameter update step, which incurs additional time

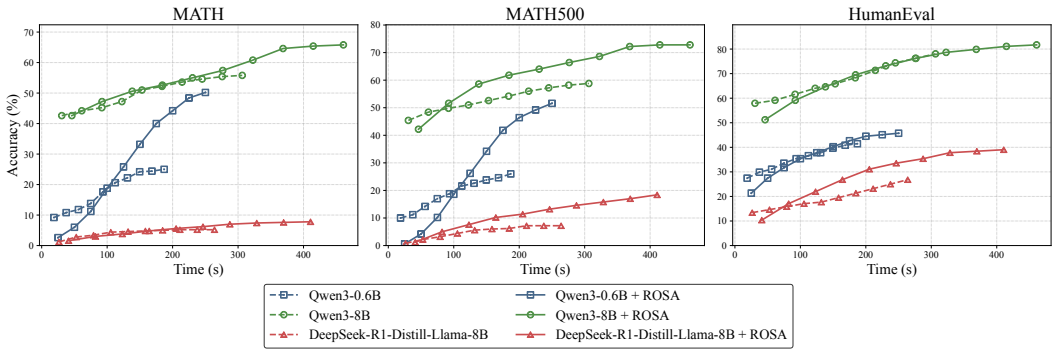

Figure 5: Time-to-Accuracy comparison across different models and datasets. The x-axis represents cumulative wall-clock time. Our method ROSA consistently exhibits steeper learning curves, highlighting its significant advantage in time efficiency.

and memory costs. As shown in Table 9, the average update time makes the total processing time per turn roughly double that of the baseline's inference-only time. However, this update process is designed to be executed asynchronously. In a real-world application, the update can be performed in the background while the user is interpreting the response of the model and formulating their next prompt. This parallel processing makes the update latency largely imperceptible to the user, enabling a seamless and responsive interactive experience.

Regarding spatial complexity, ROSA incurs negligible memory overhead. As shown in Table 9, the update peak shows only a modest increase over the inference peak for models up to 8B parameters. Crucially, to validate the scalability of our approach on larger foundation models, we profiled the overhead on the Qwen3-32B model on the MATH dataset, as detailed in Table 10. ROSA requires a maximum memory increase of only $+0.7$ GB on a 32B model. Because our approach utilizes a matrix-free Conjugate Gradient solver, it avoids materializing full Hessian-like matrices. This confirms that our method enhances reasoning capabilities without imposing prohibitive hardware constraints, and the relative latency overhead remains highly practical for real-time deployment even at large scales.

Table 9: Efficiency analysis of ROSA on models up to 8B parameters. We report the averaged inference latency and peak GPU memory per turn. The "Update" columns show the additional overhead introduced by ROSA.

Model	**Method**	**Averaged Latency (s)**		**Peak GPU Memory (GB)**	
		Inference	**Update**	**Inference**	**Update**
Qwen2.5-0.5B-Instruct	Baseline	8.42	0	11.85	0
	ROSA	7.97	7.58	11.68	18.60
Qwen3-0.6B	Baseline	18.71	0	15.61	0
	ROSA	18.85	13.16	16.24	23.65
Qwen2.5-7B-Instruct	Baseline	19.40	0	42.01	0
	ROSA	19.72	23.28	43.41	48.95
Qwen3-8B	Baseline	30.64	0	51.90	0
	ROSA	30.70	28.39	51.90	58.63

Table 10: Computational overhead on the Qwen3-32B model, evaluated on the MATH dataset.

Model / Method	**Peak GPU Memory (GB)**	**Avg. Time to Solve**
Qwen3-32B (Baseline)	94.2	1.0x (Reference)
+ ROSA (+LM+R)	94.9 (+0.7)	1.9x

E Extended Empirical Evaluations

E.1 Generalization to Open-Ended and Delayed-Reward Tasks

To comprehensively validate that the effectiveness of ROSA is not limited to tasks with clean, immediately verifiable reward signals (such as mathematical reasoning or code generation), we extend our evaluation to open-ended, subjective, and delayed-reward scenarios.

Delayed-Reward Scenarios (UI Agent Tasks): Real-world autonomous agents frequently operate in environments where feedback is exceptionally sparse and delayed. To simulate this, we evaluate ROSA on the OSWorld and AndroidWorld benchmarks using the UI-TARS-7B model (both SFT and DPO variants). In this setup, the models navigate operating systems with no intermediate feedback. A single, delayed binary reward is provided only after the agent completes a long trajectory (averaging around 15 turns). If the task fails, ROSA applies a single parameter update based on the entire failed trajectory.

As shown in Table 11, ROSA successfully handles long-horizon delayed rewards, yielding consistent improvements in success rates. This proves that our algorithm can efficiently backpropagate sparse failure signals over extended interactions to correct underlying navigation policies.

Table 11: Success Rate (%) on delayed-reward UI Agent tasks (OSWorld and AndroidWorld). The models complete long trajectories ($\sim$15 turns) before receiving a single delayed reward.

Model	Setup	OSWorld (Online)	AndroidWorld (Online)
UI-TARS-7B-SFT	Baseline	13.23	27.61
	+ ROSA (+LM+R)	16.71 (+3.48)	30.26 (+2.65)
	+ ROSA (+LM+M)	17.83 (+4.60)	30.93 (+3.32)
UI-TARS-7B-DPO	Baseline	14.80	28.90
	+ ROSA (+LM+R)	17.32 (+2.52)	30.37 (+1.47)
	+ ROSA (+LM+M)	18.04 (+3.24)	31.68 (+2.78)

Subjective & Open-Ended Scenarios: Furthermore, we evaluated ROSA on tasks lacking objective, deterministic ground truths. We utilized MT-Bench to assess subjective conversational alignment and IFEval to measure strict formatting instruction following. For MT-Bench, we employed GPT-5.2 as a continuous judge to provide fine-grained, continuous reward scores ($r_k \in [-1, 1]$). For IFEval, we utilized a programmatic structural constraint checker.

The results in Table 12 demonstrate substantial performance leaps. Notably, on MT-Bench, ROSA improved the subjective score metric by +15.21, and on IFEval, the strict accuracy increased by +11.95. These results decisively confirm that ROSA is highly task-agnostic; it can dynamically align the policy in-situ with subjective human preferences and complex structural constraints just as effectively as it solves deterministic math problems.

Table 12: Generalization to open-ended and subjective tasks using Qwen3-0.6B.

Task Domain	Benchmark	Metric	Baseline	ROSA
Subjective	MT-Bench	GPT-5.2 Score (Scaled)	60.36	75.57 (+15.21)
Instruction	IFEval	Strict Acc. (%)	58.31	70.26 (+11.95)

E.2 Comparison with Test-Time Compute Baselines

To definitively prove that the performance gains of ROSA stem from algorithmic innovation rather than simply scaling test-time compute, we comprehensively benchmarked our method against several families of test-time scaling and prompt optimization techniques. We evaluated these baselines on the MATH dataset using the Qwen3-0.6B model, enforcing a strict maximum limit of 10 interactions or samples per problem for all methods to ensure a fair comparison.

The baselines are categorized into four distinct paradigms:

- **Automatic Prompt Engineering (APE):** Methods like DSPy, TextGrad, and TSGD-M that optimize the prompt on a training subset (e.g., 200 samples from the MATH training set) using textual gradients or momentum, and apply the best prompt statically during inference.
- **Search & Sampling:** Methods that generate multiple candidate responses and select the best one, including Best-of-N ($N = 10$) and Reranking using a Reward Model (RM).
- **Self-Critique / Reflexion:** Context-based self-correction where the model verbally reflects on its past errors and attempts to self-improve based on textual feedback.
- **Iterative Adaptation:** Test-time optimization methods that update model weights iteratively using standard gradient descent at inference time (e.g., TTRL, SLOT).

Table 13: Comprehensive comparison against test-time compute baselines on the MATH dataset using Qwen3-0.6B. ROSA achieves the highest accuracy and correction uplift with a highly competitive time overhead.

Paradigm	Specific Method	Final Acc. (%)	Correction Uplift (%)	Avg. Time to Solve
Standard	Zero-Shot Baseline	26.00	17.40	1.0x (Reference)
Automatic Prompt Eng.	APE (DSPy Khattab et al. (2024))	29.20	16.83	∼1.0x
	APE (TextGrad Yuksekgonul et al. (2025))	29.40	16.76	∼1.0x
	APE (TSGD-M Ding et al. (2025))	30.20	17.16	∼1.0x
Search & Sampling	Best-of-N ($N = 10$)	36.40	20.36	∼10.0x
	Reranking ($N = 10$ + RM)	37.80	21.57	∼10.0x + RM
Self-Critique	Reflexion Shinn et al. (2023)	39.20	22.73	∼2.1x
Iterative Adaptation	TTRL Zuo et al. (2026) (Iterative)	45.40	31.67	∼21.3x
	SLOT Hu et al. (2025) (Iterative)	40.37	29.13	∼8.0x
T^2PAM **(Ours)**	**ROSA (+LM+R)**	**50.20**	**48.87**	**∼1.8x**

As detailed in Table 13, ROSA significantly outperforms all compared baselines in both final accuracy and the ability to correct initial mistakes. We highlight the following insights from this comparison:

Vs. Search & Sampling: While sampling methods predictably scale performance, they incur massive, linearly scaling computational costs (∼10x latency) and merely select from a static policy distribution without fixing the underlying reasoning flaw. ROSA dynamically alters the policy in-situ, effectively achieving the correct answer in just a few targeted turns with a fraction of the computational cost (∼1.8x).

Vs. Self-Critique / Reflexion: Context-based self-correction bottlenecks heavily on the model's intrinsic evaluation capabilities. Smaller models often lack strong intrinsic self-critique, causing methods like Reflexion to plateau quickly. ROSA bypasses this "verbal reflection" bottleneck entirely via direct, mathematical policy adaptation.

Vs. Iterative Adaptation: Test-time methods that rely on iterative gradient descent (like TTRL and SLOT) cause prohibitive inference latency due to multiple backpropagation passes per turn. Furthermore, iterative gradient descent on single samples risks catastrophic degradation. ROSA solves this by deriving a closed-form optimal policy for a highly efficient, single-step linearized update via a Conjugate Gradient (CG) solver, ensuring both speed and stability.

Vs. Automatic Prompt Engineering (APE): While APE improves starting accuracy by establishing a strong initial prompt (a Turn-0 baseline), it fundamentally yields a global, static policy. It cannot dynamically adapt to instance-specific errors mid-conversation. In contrast, ROSA translates live feedback into an in-situ parameter update, dynamically correcting the reasoning trajectory exactly where static prompts fail.

E.3 Scalability to Large Foundation Models

To validate that the effectiveness of ROSA extends beyond small- and medium-scale models ($\leq$ 8B parameters), we evaluated its scalability by applying it to the much larger Qwen3-32B foundation model. As models increase in scale, their zero-shot baseline performance naturally becomes exceptionally high, making it increasingly difficult to extract further performance gains through test-time adaptation.

We benchmarked Qwen3-32B on four challenging datasets: MATH, MMLU-R, MT-AIME24, and HumanEval. The results are summarized in Table 14.

Table 14: Performance scalability of ROSA on the Qwen3-32B model across diverse benchmarks.

Model / Method	MATH	MMLU-R	MT-AIME24	HumanEval
Qwen3-32B (Baseline)	76.40	73.63	67.93	85.37
+ ROSA (+LM+R)	87.80 (+11.40)	85.12 (+11.49)	76.19 (+8.26)	87.80 (+2.43)
+ ROSA (+HS+R)	87.40 (+11.00)	86.05 (+12.42)	77.27 (+9.34)	87.80 (+2.43)

As shown in Table 14, despite the already formidable baseline performance of the 32B model, ROSA successfully extracts massive performance gains across all evaluated domains. For instance, on the MATH dataset, ROSA improves the accuracy by an absolute +11.40%, pushing the final accuracy to 87.80%. Similar substantial improvements are observed in MMLU-R (+12.42%) and MT-AIME24 (+9.34%). These results decisively confirm that ROSA is highly scalable and remains a powerful mechanism for test-time policy alignment even for large-scale foundation models.

Furthermore, as previously detailed in our efficiency analysis (Section D.2), ROSA achieves these significant performance leaps with a negligible peak GPU memory overhead of only $+0.7$ GB on the 32B model. This confirms that ROSA scales gracefully in terms of both reasoning capability and computational resource requirements.

E.4 Stability in Extended Interaction Horizons

In our primary evaluation, we set the maximum number of interaction turns to 10. This horizon was driven by two empirical observations. First, from a user experience perspective, human patience for iteratively correcting a model averages around 10 turns. Second, standard prompt-based baselines suffer from "context fatigue" and hit a severe performance plateau after 5 to 10 turns due to the expanding context window. Therefore, a 10-turn limit serves as both a realistic boundary for human interaction and a perfect stress test to demonstrate how ROSA breaks the baseline's stagnation bottleneck.

However, to rigorously address potential concerns regarding policy drift and algorithmic stability over significantly longer horizons, we conducted an extended evaluation on the MATH dataset for up to 20 continuous conversational turns using the Qwen3-0.6B model.

The results of this extended stress test are detailed in Table 15.

Table 15: Extended horizon evaluation on the MATH dataset over 20 conversational turns (Accuracy %).

Method	Turn 5	Turn 10	Turn 15	Turn 20
Baseline	19.40	25.00	27.20	27.20
ROSA	25.80 (+6.40)	50.20 (+25.20)	57.40 (+30.20)	64.80 (+37.60)

The empirical trajectory clearly highlights the limitations of standard in-context learning: the baseline completely stagnates after turn 15, yielding no further improvements as the massive context bloat overwhelms the model's reasoning capacity. In stark contrast, ROSA effectively internalizes the continuous negative feedback directly into its parameters, completely bypassing context fatigue. Instead of stagnating or suffering from policy drift, ROSA sustains a steady and robust performance climb, reaching an impressive 64.80% accuracy by turn 20.

The extended-horizon results are consistent with the positive-net-progress regime of Theorem 4. They do not establish that its conditions hold at every empirical update. The conditional failure-mode analysis in Appendix B.4 explains why the probability mass of rejected outputs and the approximation costs affect the number of useful corrective updates.

E.5 Integration with Process Reward Models (PRM)

While our primary evaluation utilizes binary Outcome Reward Models (ORMs) to provide feedback based on the final answer, ROSA is mathematically not limited to binary or sparse rewards. As analyzed in our methodology, the derived optimal policy and the Conjugate Gradient update naturally accommodate any continuous scalar reward $r \in \mathbb{R}$. To demonstrate that ROSA can extend to complex tasks requiring dense supervision, we evaluated its integration with a Process Reward Model (PRM).

In this experiment, we utilized GPT-5.2 as a PRM to evaluate step-by-step Chain-of-Thought (CoT) reasoning on the MATH dataset using the Qwen3-0.6B model. The adaptation mechanism was refined for dense feedback: instead of waiting for the final answer, if a specific reasoning step s_t received negative feedback from the PRM, ROSA applied the parameter update $\Delta\theta$ strictly to the logits of that flawed step prefix.

Table 16 also includes the supplementary five-level rubric comparison (Appendix C.4). The rubric achieves 51.80% accuracy and 6.7 average turns, close to the continuous outcome reward result. The PRM uses a different judge and step-level supervision, so its comparison is not a controlled ablation of the rubric prompt alone.

Table 16: Performance comparison of ROSA using Outcome Reward Models (ORM) versus Process Reward Models (PRM) on the MATH dataset using Qwen3-0.6B.

Method	**Reward Granularity**	**Final Acc. (%)**	**Avg. Turns to Solve**
Baseline	None	26.00	8.9
ROSA (Rule ORM)	Final Outcome Reward	50.20	6.9
ROSA (LLM ORM)	Continuous Outcome Reward	52.20	6.7
ROSA (Rubric)	Five-Level Outcome Reward	51.80	6.7
ROSA (PRM)	Step-level Process Reward	60.40 (+10.20)	5.8

As demonstrated in Table 16, while the standard ORM-based ROSA already provides a massive improvement over the baseline, integrating step-level supervision elevates the performance even further. ROSA-PRM accelerates the adaptation process by isolating and correcting specific logical errors mid-generation, boosting the final accuracy to 60.40% while reducing the average number of turns required to solve the problem from 6.9 down to 5.8. This smoothly demonstrates that ROSA can efficiently harness fine-grained process rewards to achieve superior alignment in fewer rounds.

E.6 General Knowledge Retention

A valid concern regarding test-time parameter adaptation is the potential violation of stationarity assumptions, which could lead to model instability, calibration degradation, or catastrophic forgetting of general knowledge. We clarify that ROSA is intrinsically designed to prevent such degradation through three structural safeguards:

- **Ephemeral Updates:** Parameter adaptations in ROSA are strictly instance-specific. The model weights are immediately reset to their pre-trained base state after the interaction for each specific test sample terminates, ensuring that the global stationarity of the model is fully preserved across different tasks.
- **Sparse Parameter Updates:** ROSA confines its updates strictly to the LM head via a rank-1 LoRA or modifies the transient hidden states. This carefully targeted mechanism alters merely $\sim$0.002% of the model's parameters, leaving 99.998% of the foundational weights completely frozen and intact.
- **Theoretical Constraint:** As defined in our methodology, the closed-form optimal policy is strictly regularized by a KL divergence penalty (β). This explicitly anchors the updated policy to the pre-trained prior, mathematically preventing unconstrained semantic drift.

To empirically validate these safeguards and confirm that repeated online updates do not degrade the model, we conducted a rigorous stress test. We forced the Qwen3-0.6B model through 10 consecutive ROSA updates on unresolved problems from the MATH dataset. We then evaluated this heavily

adapted model state on MMLU (to measure general knowledge retention) and Wikitext (to measure language modeling calibration via perplexity).

Table 17: General knowledge retention stress test on Qwen3-0.6B. The model was evaluated after undergoing 10 consecutive ROSA updates on unresolved math problems.

Model State	MMLU (Zero-shot Acc. %)	Wikitext Perplexity	Average KL Divergence
Base Model	51.03	18.2037	0.0
ROSA Adapted Model (10 updates)	51.36	18.2045	0.001

As demonstrated in Table 17, general knowledge and calibration remain practically perfectly intact even after extensive test-time adaptation. The MMLU accuracy remains highly stable, the perplexity on Wikitext is virtually unchanged, and the average KL divergence from the base model is incredibly minimal (0.001). This definitively confirms that ROSA safely enhances reasoning capabilities without sacrificing the foundational knowledge or stability of the LLM.

E.7 Exploration with Multi-Sample Feedback

A potential question regarding the ROSA mechanism is whether updating based solely on a single rejected response could lead to inefficient exploration, potentially increasing the probability of other random, incorrect responses.

Theoretical Justification: Penalizing a single incorrect response does not lead to unconstrained or inefficient exploration. Recent findings (e.g., Zhu et al., *"The Surprising Effectiveness of Negative Reinforcement in LLM Reasoning,"* NeurIPS 2025) demonstrate that suppressing an incorrect generation efficiently redistributes probability mass toward other plausible, logically coherent candidates. This redistribution is strongly guided by the LLM's robust pre-trained prior, rather than scattering towards random or nonsensical outputs.

Multi-Sample Feedback (ROSA-Multi): To empirically explore whether broader feedback improves adaptation, we implemented a multi-sample variant, ROSA-Multi. In this setup, the model generates $M = 4$ candidate responses per turn. If all candidates are incorrect, we compute the parameter update $\Delta\theta$ for each individual failure and apply the averaged update to the model. This approach acts as a Monte Carlo estimation of the policy optimization, effectively reducing variance.

The results of this experiment on the MATH dataset using Qwen3-0.6B are shown in Table 18.

Table 18: Comparison of single-sample ($M = 1$) versus multi-sample ($M = 4$) feedback on the MATH dataset using Qwen3-0.6B.

Method	Feedback Samples / Turn	Final Acc. (%)	Avg. Turns to Solve	Avg. Time to Solve
Baseline	–	26.00	8.9	1.0x
ROSA-Single (Ours)	$M = 1$	50.20	6.7	∼1.8x
ROSA-Multi	$M = 4$	52.60 (+2.40)	6.5	∼6.9x

Analysis of the Trade-off: As Table 18 demonstrates, ROSA-Multi promotes slightly faster search and finds the correct answer in marginally fewer turns (6.5 vs. 6.7), leading to a higher final accuracy (52.60%). However, this marginal gain comes at the cost of significantly higher latency (∼6.9x compared to the baseline). Because multi-turn interactions inherently require real-time responsiveness, standard ROSA ($M = 1$) is deliberately designed to prioritize low-latency, real-world conversational applicability while still capturing the vast majority of the adaptation benefits.

E.8 Additional Empirical Results

This section presents supplementary empirical results to further validate our findings. First, Table 20 reports the performance of all models on three benchmarks`AIME25`, `GPQA-diamond`, and `M-IMO`which were omitted from the main text due to space constraints. Second, to provide a more complete picture of model performance, Table 19 details the *Accuracy* and *Correction Uplift* for the `DeepSeek-R1-Distill-Qwen-7B` model on both mathematical reasoning and code generation datasets. Across these additional results, a clear and consistent trend emerges: reinforcing

the conclusions from our main analysis, our proposed method, ROSA, significantly enhances both overall task performance and the capacity of model for self-correction.

Table 19: Performance of the **DeepSeek-R1-Distill-Qwen-8B** model on mathematical reasoning and code generation datasets. The values in red indicate the absolute improvement of ROSA over the baseline.

	MATH		AIME25		MATH-500		HumanEval	
Method	**Final Acc. ↑**	**Correction Uplift ↑**	**Final Acc. ↑**	**Correction Uplift ↑**	**Final Acc. ↑**	**Correction Uplift ↑**	**Final Acc. ↑**	**Correction Uplift ↑**
Baseline	7.60	3.14	10.00	3.57	7.40	6.09	45.12	17.05
ROSA	**9.80** (+2.20)	**5.65** (+2.51)	**16.67** (+6.67)	**16.67** (+13.10)	**22.20** (+14.80)	**18.62** (+12.53)	**51.22** (+6.10)	**33.75** (+16.70)

Table 20: Supplementary performance results on additional benchmarks, reporting accuracy (%). The values in red indicate the absolute improvement of ROSA variants over the baseline.

		Mathematical Reasoning	**General Reasoning**	**Multilingual Reasoning**
Model	**Method**	**AIME25**	**GPQA-diamond**	**M-IMO**
Qwen2.5-0.5B -Instruct	Baseline	3.33	3.54	1.99
	ROSA (+LM + R)	**6.67** (+3.34)	7.07 (+3.53)	2.09 (+0.10)
	ROSA (+HS + R)	**6.67** (+3.34)	8.53 (+4.99)	3.20 (+1.21)
	ROSA (+LM + M)	**6.67** (+3.34)	10.27 (+6.73)	4.71 (+2.72)
Qwen3-0.6B	Baseline	10.00	12.20	5.20
	ROSA (+LM + R)	**16.67** (+6.67)	9.09 (+-3.11)	5.30 (+0.10)
	ROSA (+HS + R)	10.00 (+0.00)	10.54 (+-1.66)	5.30 (+0.10)
	ROSA (+LM + M)	10.00 (+0.00)	13.16 (+0.96)	6.60 (+1.40)
Qwen2.5-7B -Instruct	Baseline	10.00	26.14	10.53
	ROSA (+LM + R)	**23.33** (+13.33)	42.24 (+16.10)	17.57 (+7.04)
	ROSA (+HS + R)	20.00 (+10.00)	43.16 (+17.02)	18.36 (+7.83)
	ROSA (+LM + M)	20.00 (+10.00)	45.83 (+19.69)	21.21 (+10.68)
Qwen3-8B	Baseline	16.67	41.16	20.37
	ROSA (+LM + R)	30.00 (+13.33)	69.11 (+27.95)	33.17 (+12.80)
	ROSA (+HS + R)	33.33 (+16.66)	70.27 (+29.11)	37.62 (+17.25)
	ROSA (+LM + M)	**36.67** (+20.00)	75.18 (+34.02)	39.16 (+18.79)
DeepSeek-R1 -Distill-Llama-8B	Baseline	3.33	19.03	4.36
	ROSA (+LM + R)	**16.67** (+13.34)	21.14 (+2.11)	6.32 (+1.96)
	ROSA (+HS + R)	**16.67** (+13.34)	22.23 (+3.20)	5.17 (+0.81)
	ROSA (+LM + M)	**16.67** (+13.34)	25.36 (+6.33)	6.36 (+2.00)

E.9 Comparison with Multi-Turn Training Methods

While our main analysis focuses on test-time adaptation, it is instructive to compare ROSA with traditional training-based methods for multi-turn dialogue. In this section, we benchmark the performance of ROSA against two such paradigms on the `MATH` dataset: Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL).

For the SFT baseline, we first generated a multi-turn dialogue dataset using `DeepSeek-R1` on the `MATH` training set, and then fine-tuned the base model on this newly created data. For the RL baseline (Sheng et al., 2025), we employed a Group Preference Optimization (GRPO) scheme tailored for multi-turn dialogue, similar to the approach described in our related work (Appendix A).

The results, presented in Table 21, report both *Accuracy* and *Correction Uplift*. The key finding is that ROSA, a purely test-time method, achieves performance that is comparable or even superior to these training-based approaches. This highlights a significant advantage of our method: it obviates the need for expensive data collection and resource-intensive model training, offering a more efficient and flexible solution for enhancing multi-turn capabilities.

E.10 Detailed Per-Turn Correction Analysis

In this section, we provide a fine-grained analysis of the error-correction dynamics by examining the distribution of newly solved problems across individual interaction turns. Figure 6 illustrates the number of problems that were initially incorrect but were successfully resolved in each specific round ($k = 1$ to $k = 10$).

Table 21: Comparison of ROSA with training-based methods on the **MATH** dataset for the **Qwen3-8B** model. Our test-time method achieves performance comparable to full Reinforcement Learning (RL) training and surpasses Supervised Fine-Tuning (SFT), with zero additional offline training cost.

Method	**Final Acc.** ↑	**Correction Uplift** ↑
Baseline	55.80	23.00
SFT Training	63.80	39.24
RL Training	**66.20**	**40.45**
ROSA	<u>65.80</u>	<u>40.42</u>

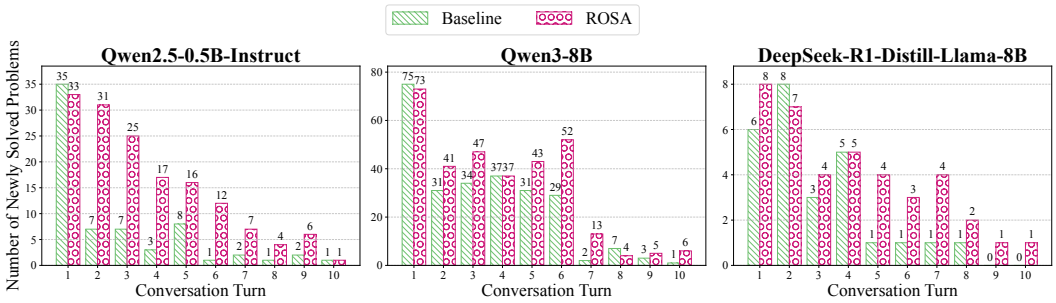

Figure 6: Detailed distribution of newly solved problems per round on the MATH dataset across various models.

The Stagnation of Baselines: The empirical results reveal a critical limitation of standard in-context learning. For the baseline models (green bars), the vast majority of successful corrections occur in the very first turn ($k = 1$). After this initial attempt, the frequency of new corrections drops precipitously, often hitting zero or near-zero by the third or fourth turn. This indicates that if a prompt-based model fails to correct itself immediately, it is highly unlikely to do so later, as the accumulating conversational context fails to provide new structural information for policy shifting.

The Long-Tail Correction of ROSA: In contrast, ROSA (purple bars) demonstrates a robust and sustained "correction tail." Not only does it solve significantly more problems in the first turn, but it continues to resolve a substantial number of difficult problems in later turns ($k = 5$ to $k = 10$). This sustained performance highlights the fundamental advantage of test-time parameter adaptation: by mathematically shifting the policy distribution based on negative rewards, ROSA actively explores new reasoning paths that the base model's static distribution could not reach. This continuous learning behavior confirms that ROSA does not just rely on context, but effectively internalizes the user's corrective feedback to escape local reasoning failures.

E.11 Learning from Successful Trajectories

Although the primary correctness protocol focuses on correcting failed attempts and resets between tasks, the practical target supports both positive and negative scalar feedback. We evaluate learning from successful trajectories on BigCodeBench-Hard with Qwen3-8B and at most 10 turns per task. Table 22 distinguishes instance-specific updates from persistent updates retained across tasks. Holding the persistent-update setting fixed, incorporating successful trajectories raises success from

Table 22: Learning from successful trajectories on BigCodeBench-Hard.

Adaptation scope	Feedback used	Success (%)
No parameter updates	None	3
Instance-specific, reset between tasks	Negative only	14
Persistent across tasks	Negative only	16
Persistent across tasks	Positive and negative	19

16% to 19%. The 14% and 16% comparison separately examines retaining updates across tasks.

This supplementary persistent setting differs from the default reset protocol and does not by itself establish long-term knowledge retention.

E.12 Adapting to Individual Preferences on ALOE

We use ALOE, introduced in Wu et al. (2025), to complement the verifiable correctness evaluations with individual-preference feedback. The evaluation uses 100 human-verified personas, with ten interaction turns per persona. The persona is hidden from the responding model, which must adapt through interaction. Qwen3-8B is evaluated against the standard multi-turn baseline under the same judging protocol. GPT-5.6 assigns Alignment Level (AL) scores $s \in \{1, 2, 3, 4, 5\}$; ROSA maps these to online rewards $r = (s - 3)/2$ and incorporates positive and negative feedback. Interactions continue for all ten turns, and adaptation is reset between personas.

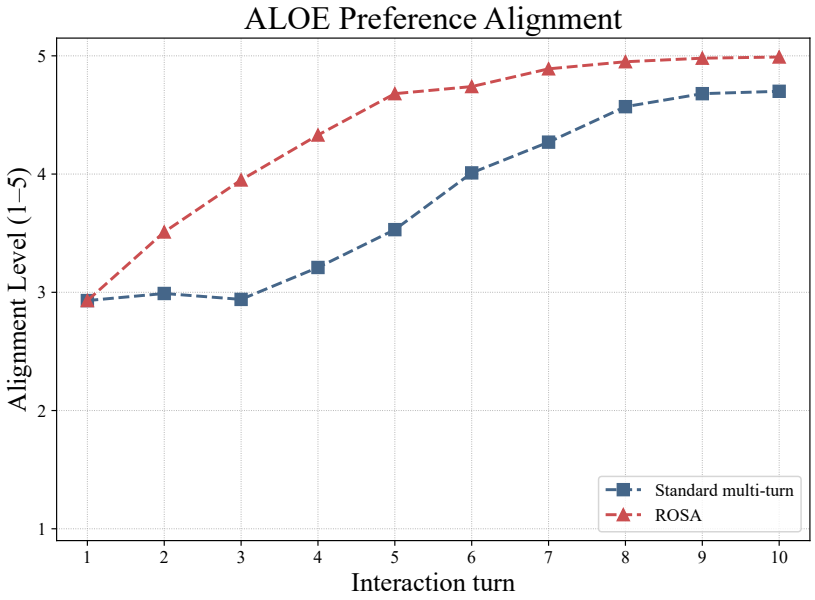

Figure 7: Preference alignment on ALOE using Qwen3-8B. Mean Alignment Level (1–5) across 100 personas is shown at each interaction turn. ROSA improves alignment more rapidly than standard multi-turn interaction, reaching 4.99 versus 4.70 at turn 10. Both methods use the same judging protocol; details are in Appendix E.12.

Table 23: ALOE preference-satisfaction scores (1–5) at each turn. The mean is over ten turns.

Method	1	2	3	4	5	6	7	8	9	10	Mean
Baseline	2.93	2.99	2.94	3.21	3.53	4.01	4.27	4.57	4.68	4.70	3.78
ROSA	2.93	3.51	3.95	4.33	4.68	4.74	4.89	4.95	4.98	4.99	4.40

Both methods start at 2.93. ROSA reaches 4.33 by turn 4 and 4.99 by turn 10, compared with 3.21 and 4.70 for the baseline, improving the reported mean by 0.62. These results complement correctness-based feedback: changing the evaluation signal enables adaptation to an additional preference objective without changing the core update rule. The scores are model-judged; the human rubric calibration in Appendix C.4 concerns MATH, not this persona evaluation.

Table 24: Comparison of Accuracy (%) on mathematical reasoning datasets with RL and ROSA.

Model	Method	MATH	MATH-500	AIME25	HumanEval
Qwen3-0.6B	RL	26.20	28.80	10.00	42.68
	ROSA	**50.20** (+24.00)	**51.60** (+22.80)	**16.67** (+6.67)	**45.73** (+3.05)
Qwen3-8B	RL	59.60	63.60	16.67	79.27
	ROSA	**65.80** (+6.20)	**72.80** (+9.20)	**30.00** (+13.33)	**81.71** (+2.44)
DeepSeek-R1-Distill -Llama-8B	RL	6.20	8.40	10.00	28.05
	ROSA	**7.80** (+1.60)	**18.40** (+10.00)	**16.67** (+6.67)	**39.02** (+10.97)

Table 25: Comparison of Correction Uplift (%) on mathematical reasoning datasets with RL and ROSA.

Model	Method	MATH	MATH-500	AIME25	HumanEval
Qwen3-0.6B	RL	18.54	20.00	6.90	22.31
	ROSA	**48.87** (+30.33)	**51.31** (+31.31)	**16.67** (+9.77)	**31.01** (+8.70)
Qwen3-8B	RL	29.37	31.58	10.71	50.00
	ROSA	**40.42** (+11.05)	**52.94** (+21.36)	**27.59** (+16.88)	**62.50** (+12.50)
DeepSeek-R1-Distill -Llama-8B	RL	4.87	7.29	6.90	16.31
	ROSA	**6.30** (+1.43)	**17.41** (+10.12)	**13.79** (+6.89)	**31.97** (+15.66)

F Ablation Studies

F.1 The Importance of the Optimization Strategy

To isolate the contribution of our proposed optimization method, we conduct an ablation study comparing the full ROSA framework against a more direct reinforcement learning approach. This baseline, which we term *RL*, directly optimizes the standard RLHF objective function in Equation equation 1). To simulate the online, multi-turn interaction setting in a comparable manner to ROSA, we estimate the gradient of $J(\pi_\theta)$ using only a single response $\mathbf{y}$ sampled from the policy π_θ for each prompt $\mathbf{x}$, and then update the model's parameters using this gradient. This approach contrasts with our full ROSA framework, which first computes a stable target policy π^* and then solves for the parameter update $\Delta\theta$.

The results of this comparison are presented in Figure 4. The analysis leads to two clear observations. First, the direct RL optimization (dotted lines) yields only marginal improvements over the baseline models (solid lines) across all three datasets. The proximity of the solid and dotted lines indicates that a naive policy gradient update with a single sample provides a noisy and inefficient learning signal, resulting in minimal performance gains. Second, in stark contrast, ROSA (dashed lines) consistently and significantly outperforms both the baseline and the RL-enhanced version. The steeper slopes of the dashed lines demonstrate that ROSA not only achieves a higher absolute accuracy but also accelerates the error correction process over the conversation turns. For example, on the MATH dataset, the Qwen3-8B model enhanced with ROSA shows a much more rapid accuracy improvement compared to its *RL* counterpart.

The quantitative results of this comparison, presented in Table 24 and Table 25, demonstrate a clear and consistent advantage for ROSA. Table 24 reveals that ROSA achieves substantially higher final accuracy across all models and datasets. For instance, on the `MATH` dataset with the Qwen3-0.6B model, ROSA surpasses the RL baseline by a remarkable **+24.00%**. Furthermore, Table 25 highlights its superior self-correction capability. In the most significant case, ROSA boosts the Correction Uplift score by **+31.31%** on `MATH-500` for the same model. The data consistently show that a direct RL update provides only marginal benefits, while our principled optimization strategy yields significant gains in both overall success and the ability to recover from errors.

This ablation study confirms that the superior performance of ROSA is not merely due to the introduction of an online reward signal. Rather, it is the principled optimization strategy—deriving a stable online target π^* and then efficiently solving for the optimal parameter update $\Delta\theta$—that is crucial for achieving effective and efficient test-time adaptation.

F.2 Ablation Study on the Influence of Hyperparameter β

Experimental Setup and the Role of β. To investigate the sensitivity of our proposed method to its hyperparameters, we conduct an ablation study on the regularization coefficient β. We vary its value across a wide range of $[0.25, 1.75]$ to observe its impact on model performance. As defined in the standard RLHF objective, β controls the trade-off between maximizing the reward and maintaining proximity to the reference policy. In the ROSA framework, its role is to modulate the intensity of the policy update based on the reward signal $r(\mathbf{x}, \mathbf{y})$, as formulated in our practical update target in Equation equation 3.A smaller β amplifies the reward signal, leading to more aggressive updates, while a larger β dampens it, resulting in more conservative updates.

Analysis and Conclusions. The results of our study are presented in Figure 8, which illustrates the cumulative accuracy over 10 conversational turns for each tested β value. A key observation is that while the initial learning trajectories vary—with smaller β values often yielding a steeper initial performance gain—all configurations converge to a similar final accuracy. This convergence can be attributed to the iterative nature of the multi-turn interaction. Although β adjusts the magnitude of each corrective step, the consistent directional feedback provided by the reward signal ensures that the model is always guided towards an improved policy. Consequently, over a sufficient number of turns, even a series of conservative updates can accumulate to achieve the correct solution.

From this analysis, we draw two key conclusions. First, for tasks with definitive solutions, such as mathematical reasoning, different search strategies—ranging from aggressive to conservative—are all highly likely to converge to the correct solution given adequate opportunities for self-correction. Second, this study underscores the **robustness of the ROSA framework**. The model's final performance demonstrates low sensitivity to the choice of β across a wide operational range, indicating that ROSA can achieve stable and effective results without extensive hyperparameter tuning.

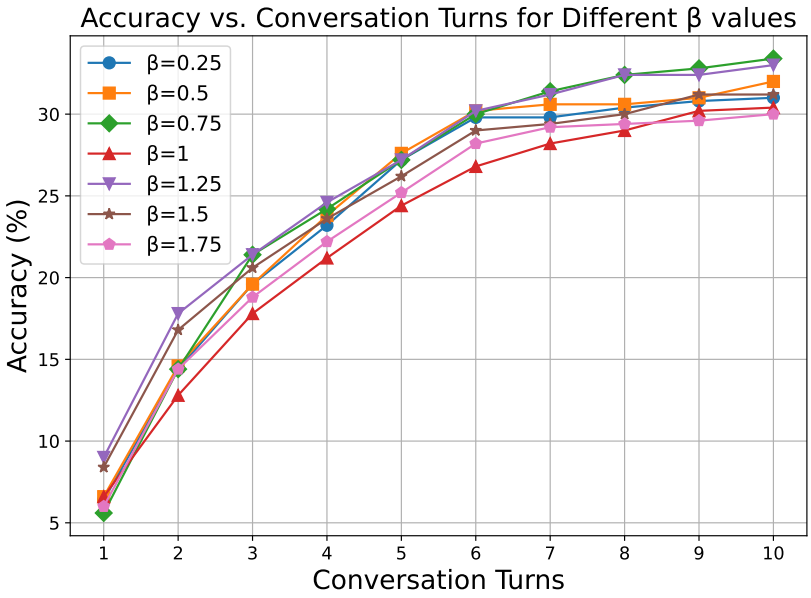

Figure 8: **Ablation study of the hyperparameter β on the MATH dataset.** The figure illustrates the cumulative accuracy over 10 conversational turns for different values of β, ranging from 0.25 to 1.75.

G CASE STUDY

Qwen3-0.8B, MATH dataset

```
Question: Convert the point (0,3) in rectangular coordinates to polar coordinates.
Enter your answer in the...

True answer: \left( 3, \frac{\pi}{2} \right)

# --- Turn 1 ---
    Baseline answer: 'Okay, so I need to convert the point (0, 3) from rectangular
    coordinates to polar coordinates. Hmm, let me remember how to do this....
    So putting it all together, the polar coordinates should be (3, pi/2).
    Let me just'

    ROSA answer: 'Okay, so I need to convert the point (0, 3) from rectangular
    coordinates to polar coordinates. Hmm, let me remember how to do this....
    So putting it all together, the polar coordinates should be (3, pi/2).
    Let me just'


# --- Turn 2 ---

    Baseline answer: 'Okay, let me try to figure this out again. The original
    point is (0, 3). In rectangular coordinates....Therefore, after checking,
    I think the correct answer is (3, pi/2). However, the user initially
    thought it was wrong, but after rethinking, it's correct.'

    ROSA answer: 'Okay, so I got confused earlier. The original point is (0, 3)
    . If I use polar coordinates,....Alternatively, if someone thought that
    theta is measured as the angle between the x-axis and the'


# --- Turn 3 ---
    Baseline answer: 'Okay, so the user initially thought that converting
    (0,3) to polar coordinates woul....Is there any possibility that theta
    could be negative? No, because theta is always between 0 and 2pi,
    and since (0,3) is in the first quadrant'

    ROSA answer: 'Okay, so the user initially thought the answer was
    (3, pi/2) but the second time they said "Wrong answer....
    **Final Answer**
    The polar coordinates of the point (0, 3)
    are \boxed{(3, \frac{\pi}{2})}.'
```