WHEN LEARNING MISSES THE TARGET: BRIDGING THE GAP BETWEEN RLVR AND ITS OPTIMUM

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning with Verifiable Rewards (RLVR) has improved the reasoning ability of large language models using outcome-based signals. However, it remains unclear whether policies learned by practical, sample-based RLVR recover the optimum of the underlying KL-regularized objective. To explore this, we derive the closed form solution of the desired optimum, i.e., a reference policy reweighted by a Gibbs distribution, whereas the Gibbs term pushes policy to explore on tokens with higher expected reward. We further find that practical RLVR policies deviate substantially from this structure and concentrate probability on a narrower set of successful solutions. We therefore introduce Reference-Anchored Policy Correction (RAPC), which freezes the reference language model and learns a bounded process-level correction from terminal verifier feedback. The resulting acting policy combines the reference distribution with the learned reward-dependent correction. On reasoning benchmarks, RAPC more closely follows the predicted Gibbs structure and retains greater coverage of successful paths than updating the full language model under comparable evaluation protocols. These results suggest that practical RLVR can better exploit the information encoded in the KL-regularized optimum by explicitly learning its reward-induced correction.

1 INTRODUCTION

By modeling Large Language Models (LLMs) (Ouyang et al., 2022; Shao et al., 2024) as a policy model, Reinforcement Learning (RL) (Sutton et al., 1998; Schulman et al., 2017) has become a standard approach to improve the complex reasoning abilities of LLMs (Lightman et al., 2024; Shao et al., 2024; Guo et al., 2025; Zhang et al., 2026b), especially for those tasks with verifiable rewards, e.g., mathematical reasoning (Lightman et al., 2024; Shao et al., 2024) and code generation (Le et al., 2022; Zhang et al., 2026a). Such Reinforcement Learning with Verifiable Rewards (RLVR) paradigm achieves significant success. The existing learning pattern is arguably believed to be a contextual bandit (Xiong et al., 2023; Shao et al., 2024; Li et al., 2026e;b). In this regime, higher-reward responses tend to dominate sampled trajectories during training, hindering other successful trajectories and reducing policy diversity. Recent work also shows that finite-batch sampling in RLVR can increasingly concentrate the policy on a small subset of high-reward responses, while reducing probabilities of other valid but less frequently sampled responses (Fan et al., 2026).

To investigate this behavior, we first characterize the optimum of the KL-regularized RLVR objective. The probability of any continuation from a given prefix, including a single token, a reasoning step, or a longer segment, has a closed form: its reference conditional transition probability is reweighted by the reward potential of the completions that remain possible after that continuation. For binary verifier rewards, this potential is determined by the probability of eventual success under reference-policy continuations. The same Gibbs structure that appears in entropy-regularized RL therefore holds at every generation granularity (Geist et al., 2019; Haarnoja et al., 2017; 2018). This raises the central question of whether practical sample-based RLVR actually recovers that structure.

To explore this, we empirically examine whether the practical RLVR trained policy model recovers the optimal target distribution. Unfortunately, our results show that the trained policy remains far from the optimal target. Since the target optimum preserves the reference policy's relative distribution among reward-equivalent responses, this deviation indicates that practical RLVR does not retain the same diversity-preserving structure. This is consistent with the bandit-like behavior observed

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.

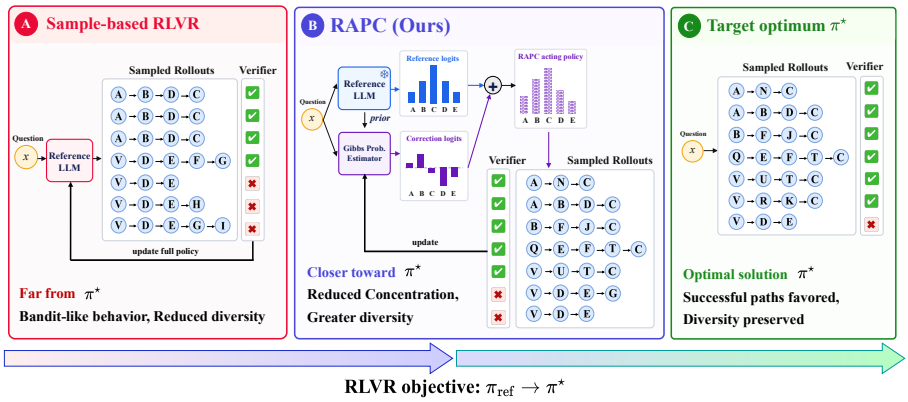


Figure 1: Comparison of three policy models. The left shows standard sample-based training, which leads to bandit-like behavior in practice; the middle shows our RAPC, which inherits the diversity-preserving advantage of the target optimum through its designed structure; the right shows the target optimum, which preserves output diversity owing to its Gibbs structure.

in practice. This mismatch arises because each update is estimated from a finite set of sampled responses through shared model parameters, making the update depend on which responses happen to be sampled. Since the updated policy affects the next round of sampling, these sample-dependent differences can accumulate during training, leading the aforementioned mismatch.

To obviate the mismatch, we propose to learn the Gibbs-type guidance part of the optimal policy model directly. Concretely, we propose *Reference-Anchored Policy Correction* (RAPC), which keeps the language model fixed and learns a token-level correction for candidate completions. The reference distribution and the learned correction jointly define the token-level acting policy used for ordinary auto-regressive sampling (Holtzman et al., 2019). Specifically, since the guidance signal can be viewed as a sequential model, we model it with a separate Transformer estimator that predicts the *Gibbs Guidance Probability (GGP)* p_ϕ. The estimator is trained in parallel using a causal mask (Vaswani et al., 2017; Radford et al., 2019). Notably, our framework is related to Q-learning approaches for LLMs (Wang et al., 2024a), but differs in how the guidance signal is learned: we train it through an online RL pipeline with lower computational cost. Finally, we summarize our contributions as follows:

1. We characterize the KL-regularized RLVR optimum at the prefix level and show that, under binary rewards, its Gibbs correction is determined by reference-policy success probability.
2. We introduce Reference-Anchored Policy Correction (RAPC), which explicitly models the policy model as its optimal formulation. Under this framework, we further propose an efficient learning paradigm to the extra guidance term.
3. Across standard reasoning benchmarks, RAPC improves the average benchmark score by 4.3% relative to GRPO and by 5.8% relative to GRPO with entropy regularization, while increasing output diversity.

2 Related Work

RL for LLM. RL has been widely used to align language models and improve their reasoning capability (Bai et al., 2022; Ouyang et al., 2022; Shao et al., 2024; Guo et al., 2025; Li et al., 2026d). Early reinforcement learning from human feedback (RLHF) methods (Ziegler et al., 2019; Su et al., 2025) typically trained a reward model from human preferences and optimized a policy model with algorithms such as Proximal Policy Optimization (PPO) (Schulman et al., 2017; Ouyang et al., 2022; Li et al., 2026c). For tasks with verifiable answers, RLVR (Lambert et al., 2024) replaces or supplements learned preference rewards with direct signals such as answer correctness or program execution (Shao et al., 2024; Guo et al., 2025; Zhang et al., 2026b). Subsequent studies have also studied the stability (Zheng et al., 2023; 2025), computational efficiency (Li et al., 2023; Ahmadian et al., 2024), and alternative policy-update rules (Liu et al., 2025; Yu et al., 2026) of RL for LLM. However, none of these existing studies have explored a basic question, whether these algorithms can really obtain the target optimal policy model.

Reward-Guided Methods. Similar to our method in this paper, reward-guided methods (Li et al., 2026b; Zhang et al., 2026b) use learned reward signals to steer language-model generation toward high-reward outputs. The methods are mainly categorized into two categories: i). learning a process reward model (Cui et al., 2025; Chen et al., 2025; Zhu et al., 2026; Kaddour, 2026) and use it to guide the training process as in standard RL (Schulman et al., 2017). ii). learning the token-level Q-value model (Wang et al., 2024a), and use it to guide the generation process (Mudgal et al., 2023; Han et al., 2024; Zhou et al., 2026). The common issue for such two techniques under RLVR is that the supervision signal is empirically estimated with extremely high variance (Yuan et al., 2026; Wang et al., 2024b), leading a large bias in guidance signal. In contrast, we propose to plug the guidance signal into the policy model, and training it as standard policy gradient-based methods (Schulman et al., 2015), which significantly reduces bias as empirically observed.

3 Target Structure of KL-Regularized RLVR

In this section, we first explore the closed-form solution of optimal policy under KL-regularized RLVR. The result indicates that the transition probability of target optimum has a Gibbs-type reward guidance signal, which guarantees the diversity of output.

3.1 Sequence-Level Gibbs Optimum

In this paper, we mainly explore RLVR under LLM as policy model which optimizes a language model over complete responses using terminal verifier rewards. Let $x \sim \mathcal{D}$ denote a prompt, $y = \{y_1, \cdots, y_T\}$ is a complete response (y_t is a single token), and $R(x, y) \in \{0, 1\}$ as its verifiable reward (our analysis below can be extended to scalar reward). We define π_{ref} as a pretrained reference model and the standard KL-regularized RLVR objective is (Ouyang et al., 2022; Shao et al., 2024):

$$\max_{\pi} \mathcal{J}(\pi) = \mathbb{E}_{x\sim\mathcal{D}} \left[\mathbb{E}_{y\sim\pi(\cdot|x)}[R(x, y)] - \beta D_{\text{KL}} \left(\pi(\cdot \mid x) \| \pi_{\text{ref}}(\cdot \mid x) \right) \right]. \tag{1}$$

The objective maximizes expected reward while the KL regularizer keeps the policy close to the pretrained reference model π_{ref} to adapt its basic ability.

For a fixed prompt, the solution to this KL-regularized objective is the Gibbs reweighting of the reference distribution (Rafailov et al., 2023; Geist et al., 2019):

$$\pi^\star(y \mid x) \propto \pi_{\text{ref}}(y \mid x) \exp\left(\frac{R(x, y)}{\beta} \right), \tag{2}$$

The key point is that the optimum reweights the reference distribution rather than replacing it with an arbitrary response distribution. For binary rewards, all responses within the same reward level receive the same multiplicative factor. The total probability assigned to correct and incorrect responses can therefore change, while the reference conditional distribution within each reward level is preserved. The (2) gives the sequence-level Gibbs solution describes how probability is redistributed over complete responses. However, the generation of LLM follows an auto-regressive paradigm, which generates the response in a token-by-token manner. Therefore, we consider a more general formulation of the target optimum. We define the general prefix s_t as $s_t = \{x, y_1, \cdots, y_t\}$, which concat prefix x with partial response $y_{\leq t}$. Moreover, let $y_{t:t+k} = \{y_{t+1}, \cdots y_{t+k}\}$, we have the following theorem to characterize the conditional transition probability of $y_{t:t+k}$ under prefix s_t. For a response prefix s, define the continuation potential:

Theorem 3.1 (Prefix-extension characterization of the target optimum)**.** *For any prefix s_t with positive probability under π_{ref}, the transition probability of optimal distribution* (2) *satisfies*

$$\pi^\star(y_{t:t+k} \mid s_t) \propto \pi_{\text{ref}}(y_{t:t+k} \mid s_t) h^\star(s_{t+k}), \quad h^\star(s_t) := \mathbb{E}_{y\sim\pi_{\text{ref}}(\cdot|s_t)} \left[\exp\left(\frac{R(x, y)}{\beta} \right) \right], \tag{3}$$

where $h^\star(s_t)$ is a Gibbs-type energy function.

The theorem is similarly obtained in (Li et al., 2026b), while we prove it under an optimal control manner in Appendix D.2. The general transition probability is an extension to (2), when $k = 1$, it degenerates into the token-level decoding, and for $k > 1$, it represents a reasoning step, or a longer segment. The intuition to this formulation is that the Gibbs guidance signals increase the weights on those intermediate steps (token or reasoning step) with higher expected reward.

Since $R \in \{0, 1\}$, the $h^\star(s_t)$ reduces to a function of the reference-policy success probability. Define$p_{\mathrm{ref}}(s_t) := \Pr_{y\sim\pi_{\mathrm{ref}}(\cdot|s_t)}[R(x, y) = 1]$, Then simple algebra shows that

$$h^\star(s_t) = 1 + p_{\mathrm{ref}}(s_t)\left(e^{1/\beta} - 1\right). \tag{4}$$

Thus, under RLVR, the Gibbs correction assigned to an extension is determined by its probability of eventual success under reference-policy continuations, which is more intuitive than (3).

4 Diagnosing Structural Deviation in Sample-Based RLVR

In the above section, we prove the optimal distribution of RLVR. The result in (3) shows that the policy model should adopt the response diversity from reference model (Kruszewski et al., 2026). However, for policy model initialized from reference model, the empirical observations in (Fan et al., 2026; Chen et al., 2026) imply that the practically learned policy model has bandit-similar behavior, and will gradually lose the output diversity during training (Li et al., 2026a). This motivates us to explore whether the optimal formulation (3) holds in practice. To do so, we conduct an empirical exploration, our results show that the trained policy model is in fact stay away from the optimal formulation (3). We further attribute this phenomenon as to the accumulation of sampling error.

4.1 Diagnostics on Mismatch to Target Optimum

The optimal transition probability (3) is applied to steps $y_{t:t+k}$, it represents token for $k = 1$ and steps for $k > 1$. In this subsection, we examine the mismatch between trained policy model and the optimal target under these two settings.

Sequence-Level Diagnostic. Firstly, we consider the a representative case of sequence level mismatch, i.e., the complete responses. For a prompt x, consider two responses y and y' with the same reward, i.e., $R(x, y) = R(x, y')$. According to (2), they should receive the same reference-relative probability adjustment:

$$\log \frac{\pi^\star(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} = \log \frac{\pi^\star(y' \mid x)}{\pi_{\mathrm{ref}}(y' \mid x)}. \tag{5}$$

We measure the deviation of the learned policy from this condition as

$$D_{\mathrm{seq}}(x; y, y') := \left|\log \frac{\pi_\theta(y \mid x)}{\pi_{\mathrm{ref}}(y \mid x)} - \log \frac{\pi_\theta(y' \mid x)}{\pi_{\mathrm{ref}}(y' \mid x)}\right|. \tag{6}$$

Since under the target optimum (3), it should satisfying $D_{\mathrm{seq}}(x; y, y') = 0$. Thus, if this the target optimum is violated, $D_{\mathrm{seq}}(x; y, y')$ should be away from zero. To check this, we sample responses for each prompt, form pairs with the same reward, and report the distribution of D_{seq} across these pairs.

Token-Level Diagnostic. Next, we consider the token level mismatch by examining whether the policy follows the target structure at each generation step. Given a prefix s_t, let y_{t+1} and y'_{t+1} be two candidate next tokens, which lead to s_{t+1} and s'_{t+1}, respectively. When $k = 1$, (3) implies:

$$\log \frac{\pi^\star(y_{t+1} \mid s_t)}{\pi_{\mathrm{ref}}(y_{t+1} \mid s_t)} - \log \frac{\pi^\star(y'_{t+1} \mid s_t)}{\pi_{\mathrm{ref}}(y'_{t+1} \mid s_t)} = \log h^\star(s_{t+1}) - \log h^\star(s'_{t+1}). \tag{7}$$

We therefore define the token-level deviation as

$$D_{\mathrm{tok}}(s_t; y_{t+1}, y'_{t+1}) := \left|\left(\log \frac{\pi_\theta(y_{t+1} \mid s_t)}{\pi_{\mathrm{ref}}(y_{t+1} \mid s_t)} - \log \frac{\pi_\theta(y'_{t+1} \mid s_t)}{\pi_{\mathrm{ref}}(y'_{t+1} \mid s_t)}\right) - \log \frac{h^\star(s_{t+1})}{h^\star(s'_{t+1})}\right|. \tag{8}$$

Then by verifying whether $D_{\mathrm{tok}}(s_t; y_{t+1}, y'_{t+1}) = 0$, we can check whether the token level policy model satisfying the target optimum. Concretely, for each sampled prefix, we form candidate token pairs and estimate the corresponding $h^\star$ values using reference-policy rollouts. We then report the distribution of D_{tok} across these pairs.Construction details for both diagnostics are provided in Appendix B

Empirical Results. Figure 2 shows deviations from the target optimum at both levels. At the sequence level, the target is zero for every same-reward pair. Nevertheless, the median D_{seq} is 9.41

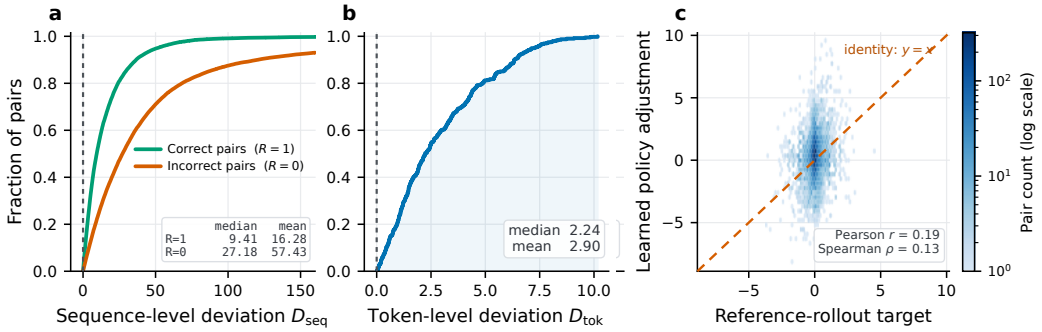


Figure 2: **Structural deviations from the target optimum under sample-based RLVR.** (a) Sequence-level deviations for correct and incorrect same-reward response pairs. (b) Token-level deviations over candidate-token pairs sharing a prefix. Dashed vertical lines mark zero deviation. (c) The token level mismatch revealed by regression over (7), the misalignment between $y = x$ reveals the mismatching.

for correct-response pairs (green) and 27.18 for incorrect-response pairs (orange). Thus, the learned policy does not preserve the equality of the reference-relative log-probability ratios required by the target optimum. At the token level, the target optimum requires $D_{\text{tok}} = 0$ for every candidate-token pair. However, the median D_{tok} is 2.24, indicating substantial deviations from the target. The sequence-level deviations are numerically larger because token-level discrepancies accumulate over the generated sequence. We further examine (7) by directly comparing its two sides. If the learned policy followed the target optimum, all points would lie on the identity line $y = x$. Instead, the two quantities have only weak Pearson and rank correlations ($r = 0.189$ and $\rho = 0.133$, respectively).

4.2 Why Sample-Based RLVR Does Not Preserve the Gibbs Structure

The diagnostics above show that practical policy model does not recover the optimal target (3) as expected. Next, we make a theoretical explanation to this phenomenon. Roughly speaking, the violation is attributed to the sampling error of policy gradient and its accumulation across training.

As can be seen in (1) the practical policy gradient is computed in expectation over response y. We prove in Appendix D.4, under proper regularity conditions, training with the ground-truth gradient can converge to the target optimum. However, due to the sampling error existed in empirical gradient, the accumulated error will lead the policy model stay away from the target optimum, making it an unstable point. The following theorem informally characterizes this conclusion the complete arguments refer to Appendix D.4.

Theorem 4.1 (Informal)**.** *Under the proper regularity assumptions in Appendix D.4, the parameter updated by ground-truth policy gradient will converge to target optimal, while the one updated by empirical policy gradient will stay away from optimal one with probability one.*

5 Reference-Anchored Policy Correction

In Section 4.2, the finite-sample analysis shows that the target Gibbs structure can be preserved only when the policy-gradient update is computed over the full policy distribution, while the required computational cost is unacceptable in practice. Thereafter, we propose Reference-Anchored Policy Correction (RAPC) in this section, explicitly modeling the p_{ref} required in target distribution (3). By combining it with reference policy model π_{ref}, we can directly obtain the policy model follows the target formulation. Next, we illustrate the way of modeling $p_{\text{ref}}(s_t)$ and its learning paradigm.

5.1 Reference-Anchored Policy and Outcome-Supervised Training

In contrast to characterizing the policy model as an individual LLM, for token y_t, we modeling the policy model as in (3) and combine it with (11) to get the parameterized model

$$\pi_\phi(y_{t+1} \mid s_t) = \frac{1}{Z_\phi(s_t)} \pi_{\text{ref}}(y_{t+1} \mid s_t) \big[1 + p_\phi(s_t, y_{t+1})(e^{1/\beta} - 1)\big], \tag{9}$$

where the normalization constant $Z_\phi(s_t)$ is defined as

$$Z_\phi(s_t) := 1 + (e^{1/\beta} - 1) \sum_{y_{t+1}} \pi_{\text{ref}}(y_{t+1} \mid s_t) p_\phi(s_t, y_{t+1}). \tag{10}$$

As can be seen, the newly designed π_ϕ follows the framework of optimal target distribution (3), while we parameterize the expected correctness instead of the whole policy model. Since the π_ϕ is a probability, we modeling it as

$$p_\phi(s_t, y_{t+1}) = \sigma(\alpha u_\phi(s_t, y_{t+1})), \tag{11}$$

where $\sigma(\cdot)$ is the sigmoid function(Cybenko, 1989), $\alpha > 0$ is a constant, and $u_\phi(s_t, y_{t+1})$ is the scalar score output by the estimator. The resulting probability p_ϕ is the Gibbs Guidance Probability (GGP). We next describe how the estimator is trained.

Intuitively, the GGP $p_\phi(s_t, y_{t+1})$ estimates the expected correctness of continuing from prefix s_t with token y_{t+1} under π_{ref}. In this sense, it plays a role similar to the score provided by a process reward model (PRM) (Yuan et al., 2024). A natural learning paradigm of $\pi_{\text{ref}}(s_t)$ is collecting trajectories started from s_t, and empirically estimate the supervision signal (Lu et al., 2024). However, we find that such learning paradigm induces large variance and the supervision signal is restricted to one token for each backpropagation leading to a low learning efficiency (see Appendix C for details). To alleviate this, we propose to recognize π_ϕ as a unified policy model, and train it as in standard RL technique for LLM e.g., GRPO (Shao et al., 2024). Concretely, we consider the following two losses.

For each prompt x, let $\{y_i\}_{i=1}^G$ denote a group of G sampled responses, with terminal rewards $r_i = R(x, y_i)$. We use the standard GRPO group-relative advantage $\widehat{A}_i$ computed from $\{r_i\}_{i=1}^G$.

$$\begin{aligned}\widehat{\mathcal{L}}_{\text{pol}}(\phi) = -\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \Bigg[\min\Big(\rho_{i,t}(\phi)\widehat{A}_i,\ \text{clip}\left(\rho_{i,t}(\phi), 1-\epsilon, 1+\epsilon\right)\widehat{A}_i\Big) \\ - \beta D_{\text{KL}}\left(\pi_\phi(\cdot \mid s_{i,t}) \,\|\, \pi_{\text{ref}}(\cdot \mid s_{i,t})\right) \Bigg].\end{aligned} \tag{12}$$

Here the $\widehat{\mathcal{L}}_{\text{pol}}$ is GRPO loss generalized from (1) with $\widehat{A}_i$ denoting the group-relative advantage of the i-th response in a group of G rollouts $\{y_i\}_{i=1}^G$ generated by $\pi_{\phi_{\text{old}}}$. The importance ratio is $\rho_{i,t}(\phi) := \pi_\phi(y_{i,t} \mid s_{i,t}) / \pi_{\phi_{\text{old}}}(y_{i,t} \mid s_{i,t})$, where $y_{i,t}$ is the t-th token of response y_i, $s_{i,t} = (x, y_{i,<t})$ and $\epsilon > 0$ is the clipping threshold.

At the terminal transition of each sampled response, the correctness label is directly observed as $r_i = R(x, y_i)$. We further use this terminal supervision to anchor the predicted correctness probability p_ϕ with a binary cross-entropy loss.

$$\widehat{\mathcal{L}}_{\text{anchor}}(\phi) = -\frac{1}{G} \sum_{i=1}^{G} \left[r_i \log p_\phi^{\text{term}}(y_i) + (1 - r_i) \log\left(1 - p_\phi(y_i)\right) \right]. \tag{13}$$

Here $\widehat{\mathcal{L}}_{\text{anchor}}(\phi)$ is the terminal boundary condition. Finally, we consider the following $\widehat{\mathcal{L}}_{\text{RAPC}}(\phi)$ as our practical training loss with $\lambda_{\text{anchor}} > 0$ balance the two terms

$$\widehat{\mathcal{L}}_{\text{RAPC}}(\phi) = \widehat{\mathcal{L}}_{\text{pol}}(\phi) + \lambda_{\text{anchor}} \widehat{\mathcal{L}}_{\text{anchor}}(\phi). \tag{14}$$

During our training process, the π_{ref} in π_ϕ is freezing, and Z_ϕ is directly computed as in (9). Besides that, since the $p_\phi(s_t)$ is modeling by a transformer model, each $p_\phi(s_t)$ can be parallelly computed by using causal mask (Radford et al., 2019).

As discussed in Section 4.2, sampling error of policy gradient accumulated during training leads to bandit-like behavior of the policy, resulting in a dominated response pattern for each prompt. We show in the next proposition that π_ϕ will not output a dominated response due to its structure in (9).

Proposition 1 (Preservation of next-token probability bounds)**.** *Let y_{t+1} and y'_{t+1} be two candidate next tokens at the same prefix s_t. Suppose$c_{\text{low}} \le \frac{\pi_{\text{ref}}(y_{t+1}|s_t)}{\pi_{\text{ref}}(y'_{t+1}|s_t)} \le c_{\text{up}}$, where $0 < c_{\text{low}} \le c_{\text{up}} < \infty$. Then the token policy in* (9) *satisfies*

$$e^{-1/\beta} c_{\text{low}} \le \frac{\pi_\phi(y_{t+1} \mid s_t)}{\pi_\phi(y'_{t+1} \mid s_t)} \le e^{1/\beta} c_{\text{up}}. \tag{15}$$

Algorithm 1 Optional process-guided step sampling

Require: Initial prefix $s = (x)$; proposal count K; frozen π_{ref}; trained estimator for the GGP p_ϕ; step length L; regularization parameter β

while s is not terminal **do**
  for $k = 1, \dots, K$ **do**
    Sample a step in length L, $z_k \sim \pi_{\text{ref}}(\cdot \mid s)$ up to the next step boundary.
    $\tilde{s}_k \leftarrow (s, z_k)$, $w_k \leftarrow 1 + p_\phi(\tilde{s}_k)\big(e^{1/\beta} - 1\big)$
  end for
  $\bar{w}_k \leftarrow w_k / \sum_{j=1}^{K} w_j$ for $k = 1, \dots, K$
  Sample $I \sim \text{Categorical}(\bar{w}_1, \dots, \bar{w}_K)$ and set $s \leftarrow \tilde{s}_I$
end while
return the response y contained in s

The proof is in Appendix D.5. The bounds show that the correction cannot make one candidate next token arbitrarily more likely than another unless such an imbalance is already present in the reference policy, up to a multiplicative factor determined by β.

5.2 Inference with π_ϕ

The optimal transition probability $\pi^*(y_{t:t+k} \mid s_t)$ defines a step level sampling strategy for $k > 1$ (π_ϕ is directly computed for $k = 1$). By plugging p_ϕ into (3) with $s_{t+k} = s_t, y_{t:t+k}$, we obtain

$$\pi_\phi(y_{t:t+k} \mid s_t) \propto \pi_{\text{ref}}(y_{t:t+k} \mid s_t)(1 + p_\phi(s_{t+k})(e^{1/\beta} - 1)), \tag{16}$$

Thus, when additional inference compute is available, we can sample step $y_{t:t+k}$ by following the similar pipeline in (Li et al., 2026b; Cardoso et al., 2022; Hammersley, 2013). The complete sampling algorithm refers to Algorithm 1.

The algorithm can be explained as follows: π_{ref} is the proposal distribution, which generates K candidate steps. The estimator predicts the GGP p_ϕ, which scores each candidate, and the categorical-resampling step in Algorithm 1 selects the next step according to their normalized weights. Such sampling strategy can intervene the generation process during the intermediate steps resulting dense and effective guidance signals as in Monte Carlo Tree Search (MCTS) (Browne et al., 2012; Park et al., 2025) but with much slighter computational cost owing to our $p_\phi(s_{t+k})$ which parallelly evaluate generated candidates. Notably, such sampling strategy preserves the structure of π_{ref} which has improved output diversity compared to post-trained model (Chen et al., 2026). Therefore, we can expect our model has better output diversity, and this is verified in Section 6.

6 Experiments

Next, we evaluate our proposed RAPC from four perspectives: reasoning performance, reward–KL efficiency, structural alignment with the target optimum (3), and the output diversity. Unless otherwise stated, all methods use the same evaluation protocol described in Appendix A.1.

6.1 Main Results

Set up. We evaluate RAPC against GRPO (Shao et al., 2024), GRPO with entropy regularization, and $Q^\sharp$ (Zhou et al., 2026) using Qwen2.5-Math-7B (Yang et al., 2024) and Qwen3-4B (Yang et al., 2025). GRPO and its entropy-regularized variant provide policy-gradient baselines; $Q^\sharp$ provides a representative Q-learning baseline for LLM post-training. In the Qwen2.5-Math-7B experiments, the policy and frozen reference share the same initialization, while RAPC trains a Qwen2.5-3B estimator for the GGP on Math12K (Lightman et al., 2024). We measure single-sample accuracy on GSM8K (Cobbe et al., 2021), MATH-500 (Lightman et al., 2024), OlympiadBench (He et al., 2024a), and SuperGPQA-500 (Du et al., 2026), and Pass@32 on AMC23 and AIME 2024–2026 (Zhang and Math-AI, 2024; 2025; Ahmadian et al., 2024). These benchmarks test mathematical reasoning and transfer to graduate-level knowledge questions. Implementation and evaluation details are provided in Appendix A.

Table 1: Benchmark results.

Method	Benchmark accuracy (%)				pass@32 accuracy (%)				Avg.
	GSM8K	MATH500	Olympiad Bench	SuperGPQA 500	AMC23	AIME24	AIME25	AIME26	
Qwen2.5-7B	67.93	64.60	29.04	19.20	82.50	43.33	30.00	23.33	44.99
+ GRPO	89.01	79.60	39.26	22.40	87.50	46.67	**33.33**	26.67	53.06
+ GRPO with entropy regularization	87.64	78.40	38.96	22.00	85.00	46.67	**33.33**	26.67	52.33
+ $Q^\sharp$	70.20	67.20	31.41	19.60	85.00	40.00	30.00	23.33	45.84
+ RAPC	**90.52**	**79.80**	**41.04**	**24.80**	**90.00**	**50.00**	**33.33**	**33.33**	**55.35**
Qwen3-4B	81.88	70.00	36.44	21.40	87.50	46.67	26.67	26.67	49.65
+ GRPO	91.13	78.00	41.93	25.20	87.50	**50.00**	36.67	30.00	55.05
+ GRPO with entropy regularization	91.58	**79.60**	40.15	24.60	90.00	43.33	30.00	**40.00**	54.91
+ $Q^\sharp$	83.47	72.40	39.11	22.00	87.5	43.33	26.67	26.67	50.14
+ RAPC	**91.81**	78.00	**42.37**	**25.40**	**92.50**	43.33	**40.00**	33.33	**55.84**

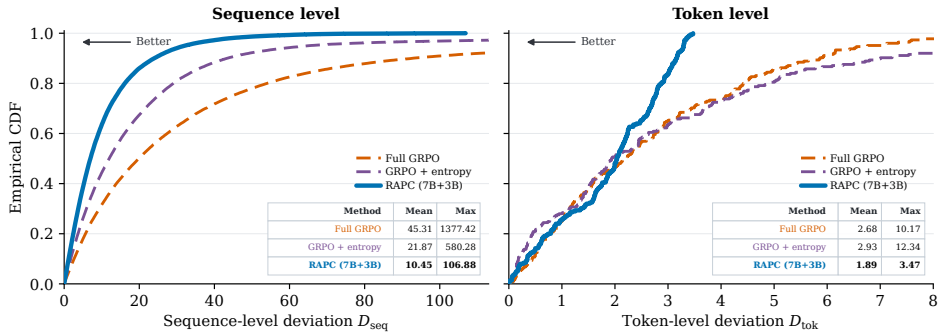


Figure 3: Empirical CDFs of sequence-level and token-level deviations from the Gibbs target. Higher CDF at the same deviation is better; the solid curve is RAPC and dashed curves are baselines. The legend reports the mean deviation.

Reasoning Performance. On the Qwen2.5-7B backbone, RAPC reaches an average score of 55.35, a 4.3% relative improvement over GRPO (53.06) and a 5.8% relative improvement over GRPO with entropy regularization (52.33). The gain is also visible on the harder multi-sample evaluations: on AIME26, RAPC improves Pass@32 from 26.67 to 33.33. Thus, the reference-relative correction improves both average reasoning accuracy and multi-sample coverage without updating the frozen reference model. We analyze the earlier offline token-level soft-Q experiments and distinguish them from the iterative data aggregation used by Q♯ (Zhou et al., 2026) in Appendix C.

Mismatch to Target Optimum. We evaluate deviation from the Gibbs target using the sequence-level metric D_{seq} and token-level metric D_{tok} defined in Section 4.1, with evaluation details in Appendix B. As shown in Figure 3, RAPC consistently yields smaller deviations than Full GRPO and GRPO with entropy regularization. Its mean D_{seq} is 10.45, versus 45.31 and 21.87, while its mean D_{tok} is 1.89, versus 2.68 and 2.93, respectively. This indicates closer alignment with the target Gibbs structure, consistent with Proposition 1.

Table 2: Token- and sequence-level entropy.

Method	Token entropy	Successful-sequence entropy
Full GRPO	0.204 [0.186, 0.222]	0.470 [0.301, 0.657]
GRPO + entropy	0.296 [0.275, 0.319]	0.488 [0.314, 0.675]
RAPC	**0.420** [**0.397**, **0.444**]	**0.651** [**0.462**, **0.852**]

Token- and Sequence-Level Entropy. Proposition 1 predicts that bounded reweighting preserves diversity among successful responses. We measure token-level policy entropy on each method's own Math12K generations. For sequence-level entropy, each method scores the same 32 reference-policy candidates per prompt for 160 prompts and computes entropy over verifier-accepted responses; details are provided in Appendix A.2. Table 2 shows that RAPC leads at both levels (0.420 and 0.651), indicating greater diversity across successful paths.

6.2 Process-Guided Step Sampling

As described in Section 5.1, p_ϕ estimates continuation correctness and scores candidate steps. We evaluate this use of the GGP with the sampling procedure in Algorithm 1, varying step length L.

Table 3: Test-time inference results. Accuracy is reported in %. Relative latency is normalized to single-sample inference. Oracle pass32 gives the upper bound of the sampled candidate pool.

Inference procedure	Benchmark accuracy (%)			Avg.	Rel. latency(×)
	MATH500	Olympiad Bench	SuperGPQA 500		
Reference sampling	50.60	19.85	13.40	27.95	1.00
Oracle pass@32	93.00	68.74	75.00	78.91	n/a
Self-consistency @32	81.20	46.22	26.00	51.14	8.20
p_ϕ as ORM @32	82.20	45.48	28.40	52.03	7.46
Beam search	65.40	36.44	21.80	41.21	9.50
PRM-weighted step sampling	72.80	38.67	21.80	44.42	4.75
p_ϕ**PRM Algorithm1**	**74.60**	**38.67**	**22.60**	**45.29**	5.29

Comparisons with TTS Methods. Using Qwen2.5-7B as the reference model and a 3B estimator for the GGP p_ϕ, we compare RAPC with selection-based and intermediate-branching methods on MATH500, OlympiadBench, and SuperGPQA500. The PRM-weighted baseline uses Skywork-PRM (He et al., 2024b). As shown in Table 3, p_ϕ-weighted step sampling achieves the best average accuracy among branching methods (45.29%), outperforming beam search and PRM-weighted step sampling by 4.08 and 0.87 points, respectively. Compared with selection-based methods, it offers a better accuracy–latency balance, achieving substantially lower latency with competitive accuracy.

More Explorations on Step-Level Guidance. At $\beta = 0.3$, Figure 4 shows that step-level guidance improves MATH500 accuracy with modest latency growth. For $K = 4$, reducing L from 512 to 128 raises accuracy from 63.0% to 74.6%, while latency increases from 4.95× to 5.29×. Further reducing L to 64 yields a 0.4-point gain to 75.0%, while latency increases only to 5.35×, indicating diminishing returns. In contrast, increasing K from 4 to 8 nearly doubles latency, with accuracy gains ranging from 1.2 to 3.0 percentage points. Overall, $K = 4, L = 128$ provides a favorable accuracy–latency trade-off.

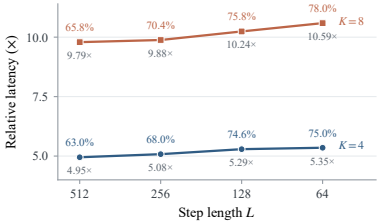


Figure 4: MATH500 step guidance ($\beta = 0.3$); accuracy is shown above each point.

6.3 Ablation Studies

Table 4: Effect of model scale.

Size	Val.	$D_{\text{seq}} \downarrow$	$D_{\text{tok}} \downarrow$	Entropy
1.5B	75.6	11.312	1.977	0.623
3B	**79.8**	10.475	1.888	**0.651**
7B	**79.8**	**3.258**	**1.037**	0.535

Table 5: Effect of inference-time β.

β	0.1	0.2	0.3	0.5	1.0
Acc. (%)	69.8	71.4	**74.6**	70.6	66.0

Process-model scale. We first investigate the effect of the model scale of the correctness model p_ϕ. Specifically, we compare 1.5B, 3B, and 7B variants while keeping the remaining experimental settings unchanged. The results are summarized in Table 4. The 7B model achieves the lowest D_{seq} and D_{tok}, indicating smaller distributional deviation, while the 3B model obtains the best validation performance. Considering both performance and model scale, we choose the 3B model as p_ϕ in our main experiments.

Inference-time guidance strength. We further study the effect of the regularization parameter β during inference. Specifically, we vary β while keeping the other inference settings fixed and report the results in Table 5. Performance is sensitive to the guidance strength: a moderate β provides a better trade-off, whereas overly weak or strong regularization degrades performance.

7 Conclusion

We showed that practical RLVR policies can deviate from the Gibbs-form optimum of the KL-regularized objective. To address this gap, RAPC learns a bounded reward-dependent correction to a frozen reference model. Across reasoning benchmarks, RAPC improves accuracy, more closely matches the target distribution, and preserves greater diversity among successful solutions.

AI USE DISCLOSURE

We used generative AI tools to translate portions of the manuscript, improve the readability of the English text, and assist in writing experimental code. The authors reviewed the AI-assisted text and code and examined the experimental outputs when preparing the manuscript. We take responsibility for the final content of this work.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce-style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267, 2024.

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

Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

Gabriel Cardoso, Sergey Samsonov, Achille Thin, Eric Moulines, and Jimmy Olsson. Br-snis: bias reduced self-normalized importance sampling. *Advances in Neural Information Processing Systems*, 2022.

Hongzhan Chen, Tao Yang, Shiping Gao, Ruijun Chen, Xiaojun Quan, Hongtao Tian, and Ting Yao. Discriminative policy optimization for token-level reward models. *arXiv preprint arXiv:2505.23363*, 2025.

Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *Advances in Neural Information Processing Systems*, 38:57654–57689, 2026.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

George Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314, 1989.

Xeron Du, Yifan Yao, Kaijing Ma, Bingli Wang, Tianyu Zheng, Minghao Liu, Yiming Liang, Xiaolong Jin, Zhenlin Wei, Chujie Zheng, et al. Supergpqa: Scaling llm evaluation across 285 graduate disciplines. *Advances in Neural Information Processing Systems*, 38, 2026.

Mingyuan Fan, Weiguang Han, Daixin Wang, Cen Chen, Zhiqiang Zhang, and Jun Zhou. When sharpening becomes collapse: Sampling bias and semantic coupling in rl with verifiable rewards. *arXiv preprint arXiv:2601.15609*, 2026.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *International conference on machine learning*, pages 2160–2169. PMLR, 2019.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pages 1352–1361. PMLR, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr, 2018.

John Hammersley. *Monte carlo methods*. Springer Science & Business Media, 2013.

Seungwook Han, Idan Shenfeld, Akash Srivastava, Yoon Kim, and Pulkit Agrawal. Value augmented sampling for language model alignment and personalization. *arXiv preprint arXiv:2405.06639*, 2024.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, 2024a.

Jujie He, Tianwen Wei, Rui Yan, Jiacai Liu, Chaojie Wang, Yimeng Gan, Shiwen Tu, Chris Yuhao Liu, Liang Zeng, Xiaokun Wang, Boyang Wang, Yongcong Li, Fuxiang Zhang, Jiacheng Xu, Bo An, Yang Liu, and Yahui Zhou. Skywork-o1 open series, November 2024b. URL `https://doi.org/10.5281/zenodo.16998085`.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Jean Kaddour. Target policy optimization. *arXiv preprint arXiv:2604.06159*, 2026.

Olav Kallenberg. *Foundations of modern probability*. Springer, 1997.

Germán Kruszewski, Pierre Erbacher, Jos Rozen, and Marc Dymetman. Whatever remains must be true: Filtering drives reasoning in llms, shaping diversity. In *International Conference on Learning Representations*, volume 2026, pages 85291–85316, 2026.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven Chu Hong Hoi. Coderl: Mastering code generation through pretrained models and deep reinforcement learning. *Advances in Neural Information Processing Systems*, 35:21314–21328, 2022.

Long Li, Zhijian Zhou, Jiaran Hao, Jason Liu, Yanting Miao, Wei Pang, Xiaoyu Tan, Wei Chu, Zhe Wang, Shirui Pan, et al. The choice of divergence: A neglected key to mitigating diversity collapse in reinforcement learning with verifiable reward. In *International Conference on Learning Representations*, volume 2026, pages 64742–64768, 2026a.

Xiuyu Li, Jinkai Zhang, Mingyang Yi, Yu Li, Longqiang Wang, Yue Wang, and Ju Fan. Ets: Energy-guided test-time scaling for training-free rl alignment. In *Proceedings of the Conference on Machine Learning*, 2026b.

Yu Li, Xiuyu Li, Mingyang Yi, Jiaxing Wang, Liangxu Zhang, Zhaolong Xing, and Zhen Chen. Turning off-policy tokens on-policy: A plug-in approach for improving llm alignment. *arXiv preprint arXiv:2607.04728*, 2026c.

Yu Li, Mingyang Yi, Xiuyu Li, Ju Fan, Fuxin Jiang, Binbin Chen, Peng Li, Jie Song, and Tieying Zhang. Reasoning and tool-use compete in agentic rl: From quantifying interference to disentangled tuning. Preprint arXiv:arXiv:2602.00994, 2026d.

Zhaochun Li, Mingyang Yi, Yue Wang, Shisheng Cui, and Yong Liu. Towards a theoretical understanding to the generalization of rlhf. Preprint arXiv:2601.16403, 2026e.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. *arXiv preprint arXiv:2310.10505*, 2023.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601, 2024.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Jianqiao Lu, Zhiyang Dou, Hongru Wang, Zeyu Cao, Jianbo Dai, Yingjia Wan, Yunlong Feng, and Zhijiang Guo. Autopsv: Automated process-supervised verifier. *Advances in Neural Information Processing Systems*, 37:79935–79962, 2024.

Sidharth Mudgal, Jong Lee, Harish Ganapathy, YaGuang Li, Tao Wang, Yanping Huang, Zhifeng Chen, Heng-Tze Cheng, Michael Collins, Trevor Strohman, et al. Controlled decoding from language models. *arXiv preprint arXiv:2310.17022*, 2023.

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

Sungjin Park, Xiao Liu, Yeyun Gong, and Edward Choi. Ensembling large language models with process reward-guided tree search for better complex reasoning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10256–10277, 2025.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

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. *Advances in neural information processing systems*, 36:53728–53741, 2023.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pages 1889–1897. Pmlr, 2015.

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

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

Xuerui Su, Yue Wang, Jinhua Zhu, Mingyang Yi, Feng Xu, Zhiming Ma, and Yuting Liu. Reveal the mystery of dpo: The connection between dpo and rl algorithms. *arXiv preprint arXiv:2502.03095*, 2025.

Richard S Sutton, Andrew G Barto, and Andrew Barto. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Chaojie Wang, Yanchen Deng, Zhiyi Lyu, Liang Zeng, Jujie He, Shuicheng Yan, and Bo An. Q*: Improving multi-step reasoning for llms with deliberative planning. *arXiv preprint arXiv:2406.14283*, 2024a.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, 2024b.

Wei Xiong, Hanze Dong, Chenlu Ye, Ziqi Wang, Han Zhong, Heng Ji, Nan Jiang, and Tong Zhang. Iterative preference learning from human feedback: Bridging theory and practice for rlhf under kl-constraint. *arXiv preprint arXiv:2312.11456*, 2023.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Huining Yuan, Zelai Xu, Huaijie Wang, Xiangmin Yi, Jiaxuan Gao, Xiao-Ping Zhang, Yu Wang, Chao Yu, and Yi Wu. Verifiable process rewards for agentic reasoning. *arXiv preprint arXiv:2605.10325*, 2026.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels. *arXiv preprint arXiv:2412.01981*, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025, 2025.

Yining Zhang, Mingyang Yi, Chen Wang, Xuwen Xiang, Tianhe Jia, Zedong Dan, Chengqing Zong, and Yue Wang. Htam: Hierarchical transition-attended memory for operator optimization. *arXiv preprint arXiv:2605.29734*, 2026a.

Yuxin Zhang, Meihao Fan, Ju Fan, Mingyang Yi, Yuyu Luo, Guoliang Li, Bin Wu, and Wenchao Zhou. Reward-sql: Boosting text-to-sql via stepwise execution-aware reasoning and process-supervised rewards. *Proceedings of the ACM on Management of Data*, 2026b.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025. URL `https://arxiv.org/abs/2507.18071`.

Rui Zheng, Shihan Dou, Songyang Gao, Yuan Hua, Wei Shen, Binghai Wang, Yan Liu, Senjie Jin, Qin Liu, Yuhao Zhou, et al. Secrets of rlhf in large language models part i: Ppo. *arXiv preprint arXiv:2307.04964*, 2023.

Jin Zhou, Kaiwen Wang, Jonathan Chang, Zhaolin Gao, Nathan Kallus, Kilian Weinberger, Kianté Brantley, and Wen Sun. $Q^\sharp$: Provably optimal distributional rl for llm post-training. *Advances in Neural Information Processing Systems*, 38:32906–32942, 2026.

Xuekai Zhu, Daixuan Cheng, Dinghuai Zhang, Henry Li, Kaiyan Zhang, Che Jiang, Youbang Sun, Ermo Hua, Yuxin Zuo, Xingtai Lv, et al. Flowrl: Matching reward distributions for llm reasoning. In *International Conference on Learning Representations*, volume 2026, pages 151758–151780, 2026.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences. *arXiv preprint arXiv:1909.08593*, 2019.

A Experimental Details

This appendix records the settings needed to reproduce the main results in Section 6. We separate the shared main-experiment protocol from ablations and from the diagnostic study, whose construction is given in Appendix B.

A.1 Main Experiments

Training data and models. We use Math12K, a dataset converted from PRM800K (Lightman et al., 2024); . For Qwen2.5-Math-7B experiments, we use the original 12,000-example `train.parquet` split. For Qwen3-4B-Base experiments, we use a repaired 12,000-example training split in which the two examples with empty answers in the original split are assigned answer 0. Both configurations use the same 500 examples from `test.parquet` for validation. The Qwen2.5-Math-7B and Qwen3-4B-Base backbones remain frozen and provide the reference distribution. In both configurations, a trainable Qwen2.5-3B GGP estimatorl p_ϕ supplies the learned correction and is anchored by a frozen step-0 copy.

Training protocol. Both configurations use 256 prompts per update, $G = 8$ sampled completions per prompt, rollout temperature 1.0, one PPO epoch per rollout batch, standard-deviation normalization of group-relative advantages, policy clip ratios of 0.2 and 0.3, a dual-clip coefficient of 3.0, and gradient clipping at norm 1.0. We use AdamW with learning rate 10^{-6}, weight decay 0, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\epsilon = 10^{-8}$, with a constant learning-rate schedule and no warmup. Both configurations use 200 training updates and a maximum combined prompt-and-response length of 4096 tokens. For RAPC, we use correction strength $\beta = 0.3$, sigmoid scale $\alpha = 1.0$, and terminal anchor-loss weight $\lambda_{\text{anchor}} = 0.05$.

Evaluation protocol. All model configurations use the same 500-question Math12K validation set. We validate every 10 training updates and select the checkpoint with the highest validation accuracy. For benchmark evaluation, all methods use the same examples, prompts, and answer-scoring procedure. We report single-sample accuracy on GSM8K, MATH500, OlympiadBench, and SuperGPQA-500, and Pass@32 on AMC23 and AIME 2024–2026. Pass@32 uses 32 completions per problem. Both evaluation settings use temperature 0.2, top-p 1.0, and a maximum generation length of 4096 tokens.

Implementation. Training uses BF16 with FSDP on four NVIDIA A100-SXM4-80GB GPUs.

A.2 Token- and Sequence-Level Entropy

Token-level entropy. For each method, we independently generate 16 responses for each of 160 randomly selected Math12K prompts, using temperature 1.0, top-$p = 1.0$, and a maximum response length of 8192 tokens. At each sampled position, we compute the Shannon entropy of the policy's next-token distribution. We average over tokens within each response and then over responses within each prompt. The table reports the mean across prompts; its 95% confidence intervals use prompt-level bootstrap resampling.

Successful-sequence entropy. In a separate fixed-candidate analysis, the reference policy generates 32 responses for each of 160 Math12K prompts using temperature 1.0, top-$p = 1.0$, and a maximum length of 8192 tokens, for 5,120 candidates in total. All three evaluated methods score exactly the same candidate token sequences. For a prompt and method, let $\mathcal{C}^+$ be the set of correct candidates and let p_i be the model's sequence probability for candidate i. We renormalize these probabilities within the correct set, $q_i = p_i / \sum_{j \in \mathcal{C}^+} p_j$, and compute $H^+(x) = -\sum_{i \in \mathcal{C}^+} q_i \log q_i$. Prompts with fewer than two correct candidates are excluded. We report the mean of the prompt-level values and an empirical 95% interval across prompts, giving each prompt equal weight.

A.3 Inference-Time Sampling and Ablations

All ablations in Section 6.3 retain the same 7B acting model, frozen 7B reference model, prompts, verifier, optimizer, update budget, and evaluation protocol. The estimator-scale ablation varies only

the GGP estimator size (1.5B, 3B, or 7B), while keeping the acting and reference models fixed; all other training details follow Appendix A.1.

The test-time-sampling comparison in Section 6.2 uses MATH500, OlympiadBench, and SuperGPQA-500. All compared methods use temperature 0.4, top-$p = 1.0$, and a maximum sequence length of 8192 tokens. The beam-search baseline uses beam width 4. The GGP-weighted sampler uses $K = 4$, $L = 128$, and $\beta = 0.3$ in this comparison. For the MATH500 step-length sweep, we vary $K \in \{4, 8\}$ and $L \in \{64, 128, 256, 512\}$ with $\beta = 0.3$. The guidance-strength sweep fixes $K = 4$ and $L = 128$ and varies $\beta \in \{0.1, 0.2, 0.3, 0.5, 1.0\}$. Relative latency is normalized to the time for one reference sample.

For fixed-candidate selection, every rule receives the same 32 reference candidates for each problem. The reported additional time includes scoring and selection after generation, but excludes the shared reference-candidate generation time.

B Observational Diagnostics of Structural Deviation

This appendix specifies the experiments behind the sequence-level and token-level diagnostics in Section 4.1. Both test the Gibbs structure using a GRPO policy and its original reference model, but at different levels: complete responses and individual next-token decisions.

B.1 Shared Models and Reward

The evaluated policy π_θ is the Qwen2.5-Math-7B GRPO checkpoint at training step 200; the reference policy π_{ref} is the original Qwen2.5-Math-7B model. For both diagnostics, the reward is binary and equals one when the final mathematical answer is correct and zero otherwise. We apply the answer parser to the cleaned response and compare the extracted answer with the gold answer.

B.2 Sequence-Level Diagnostic

We sample 64 prompts from the Math12K training set (Lightman et al., 2024). For each prompt, the reference policy generates 32 complete responses using temperature 1.0, top-$p = 1.0$, and a maximum response length of 4096 tokens, for 2048 responses in total. We score these fixed responses under both π_θ and π_{ref} by teacher forcing the same token sequence through each model and summing its token log probabilities.

For each prompt, we form every unordered pair of responses within each reward group, separately for $R = 1$ and $R = 0$. This yields 5,326 correct-response pairs and 14,471 incorrect-response pairs, or 19,797 same-reward pairs in total. We compute D_{seq} from Eq. (6) for each pair and plot the pair-level empirical CDF. Since responses recur across pairs, the pairs are dependent; any confidence intervals for this analysis should resample prompts as clusters.

B.3 Token-Level Diagnostic

We sample 100 prompts from the Math12K training set and generate one reference trajectory per prompt using temperature 1.0 and top-$p = 0.99$. Within the first 160 generated tokens of each trajectory, we select the eight positions with the highest next-token entropy under π_θ; ties are broken using the top-1/top-2 probability margin. This gives 800 prefixes. Thus, the prefixes come from reference-policy trajectories, while the selection of positions depends on the evaluated GRPO policy.

At each prefix, we take the 32 highest-probability next tokens under π_θ, discard special tokens from this pool, and retain the four highest-probability remaining candidate actions. We evaluate each forced action by appending it to the prefix and sampling 64 independent continuations from π_{ref}, with temperature 1.0, top-$p = 0.99$, and a maximum complete-response length of 2048 tokens. The reward of each continuation is its binary answer-correctness score. This gives 3,200 forced actions and 204,800 reference rollouts.

For each prefix, the four candidate actions form six unordered pairs. For each action we compute its policy-to-reference log-probability adjustment from π_θ and π_{ref}; we compare the difference in adjustments with the difference in estimated log continuation potentials, as in Eq. (7). The resulting

4,800 pairwise D_{tok} values define the token-level empirical CDF. We also compare the learned and rollout-estimated gaps directly, reporting Pearson and Spearman correlations as in Figure 2.

C Analysis of $Q^\sharp$ Value Guidance

Equation (3) states that, at the token level, the reference probability should be corrected by the value of the resulting prefix. $Q^\sharp$ (Zhou et al., 2026) is a direct value-based realization of this idea: a learned token-level value guides decoding from the reference policy. We evaluate $Q^\sharp$ alongside GRPO under the protocol of Table 1, then examine the value signal used for guidance.

C.1 Comparison with GRPO

On Qwen2.5-7B, $Q^\sharp$ raises the average benchmark score from 44.99 to 45.84, whereas GRPO reaches 53.06. On Qwen3-4B, the corresponding scores are 49.65, 50.14, and 55.05. Thus, $Q^\sharp$ improves over each reference model in our evaluation, but its gains are smaller than those of GRPO (Figure 5, upper left). The original $Q^\sharp$ study reports MATH pass@1 of 58.6% for its reference model and 61.9% with $Q^\sharp$; it also reports 68.4% for PPO and 71.1% for PPO with $Q^\sharp$ guidance (Zhou et al., 2026). The GRPO comparison here uses our matched evaluation.

C.2 Diagnosing the Value Estimates

We first examine whether decreasing fitting loss yields a useful local value signal. In two token-level value-fitting runs, training loss decreases substantially (Figure 5, upper right). In the first run, validation loss falls from 0.1544 to 0.0765, yet the final mean value scores for tokens from correct and incorrect responses differ by only 0.028. The second Math run ends with a larger difference of 0.208. Loss reduction and aggregate score separation therefore vary across runs; neither alone shows that the estimator ranks alternative tokens correctly at the same prefix.

We next inspect the available supervision. A diagnostic reference-policy collection contains eight responses for each of 7,497 Math questions. All eight are correct for 56.6% of questions and all eight are incorrect for 16.4%, so 73.0% provide no within-question contrast in terminal reward (Figure 5, lower left). When the same outcome labels every token of a response, such data provide limited evidence about which alternative next token is better at a shared prefix. The pooled mean value score is 0.747 for tokens from correct responses and 0.377 for tokens from incorrect responses, but these averages combine different questions and prefixes. In a separate local Math evaluation with 4,566 examples, the reference accuracy is 71.48%, while value-guided decoding reaches 70.98% for $\lambda = 0.03$–0.1 and 70.89% for $\lambda = 0.2$.

Finally, we inspect token ranking at a shared prefix. In a later diagnostic checkpoint, we locate the first token where a reference and a guided answer diverge, then compare the token scores under their common prefix. We retain pairs with different final correctness outcomes. The token leading to the correct answer receives the higher score in 33.3%–57.1% of such pairs, with 7–20 pairs per setting (Figure 5, lower right). All 95% Wilson intervals include 50%. Each branch contributes one completed answer, so this diagnostic measures observed ranking rather than the true conditional expected reward of each candidate token.

These observations point to a possible weakness of offline value fitting: terminal labels can distinguish complete responses while leaving the relative value of next-token choices at a particular prefix poorly constrained. Guidance acts on precisely these local comparisons. The small gains of $Q^\sharp$ relative to GRPO in our evaluation are consistent with this mechanism, although the diagnostics do not identify it as the sole cause.

D Detailed Theoretical Proofs

This appendix provides the derivations supporting the target structure, finite-sample analysis, and method guarantees in the main text. We use natural logarithms. For a prompt x, $\mathcal{Y}_x$ denotes the set of complete responses that terminate with an end-of-sequence token. The reference policy is denoted by $q(y \mid x) = \pi_{\mathrm{ref}}(y \mid x)$. We assume that q has positive mass on every response under consideration,

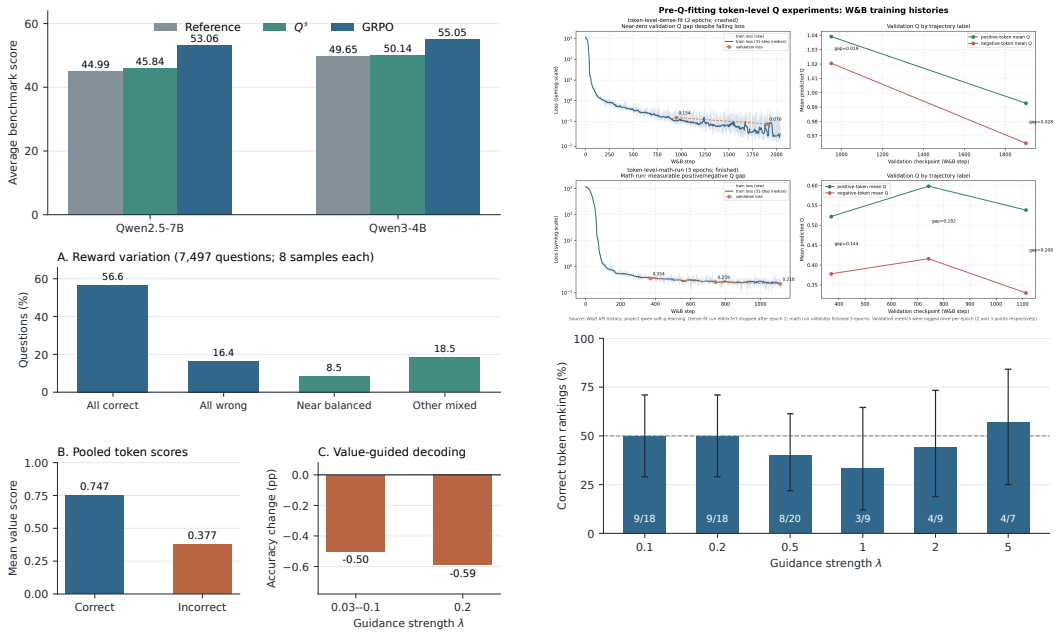


Figure 5: Results and diagnostics for $Q^\sharp$. Left: benchmark scores (top) and reference-response and guided-decoding diagnostics (bottom). Right: value-fitting histories (top) and same-prefix token ranking (bottom). The diagnostics use separate runs.

that $\beta > 0$, and that all partition functions below are finite. The same calculations hold with sums replaced by integrals for a continuous response space.

D.1 Sequence-level Gibbs optimum

Theorem D.1. *For a fixed prompt x and $\beta > 0$, consider*

$$J_x(\pi) = \mathbb{E}_{y\sim\pi(\cdot|x)}\big[R(x,y)\big] - \beta D_{\mathrm{KL}}(\pi(\cdot \mid x) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x)), \tag{D.1}$$

over policies satisfying $\sum_y \pi(y \mid x) = 1, \pi(y \mid x) = 0$ whenever $\pi_{\mathrm{ref}}(y \mid x) = 0$. Then the unique maximizer is

$$\pi^\star(y \mid x) = \frac{\pi_{\mathrm{ref}}(y \mid x)\exp(R(x,y)/\beta)}{Z(x)}, \tag{D.2}$$

where

$$Z(x) = \sum_y \pi_{\mathrm{ref}}(y \mid x)\exp\left(\frac{R(x,y)}{\beta}\right). \tag{D.3}$$

Proof. Fix a prompt x. We optimize $J_x(\pi)$ over policies satisfying

$$\sum_y \pi(y \mid x) = 1, \qquad \pi(y \mid x) = 0 \quad \text{whenever} \quad \pi_{\mathrm{ref}}(y \mid x) = 0.$$

For notational simplicity, write

$$\pi_y = \pi(y \mid x), \qquad \pi_y^{\mathrm{ref}} = \pi_{\mathrm{ref}}(y \mid x), \qquad R_y = R(x,y).$$

On the support of $\pi_{\mathrm{ref}}(\cdot \mid x)$, the constrained optimization problem has Lagrangian

$$\mathcal{L}(\pi,\lambda) = \sum_y \pi_y R_y - \beta \sum_y \pi_y \log\frac{\pi_y}{\pi_y^{\mathrm{ref}}} + \lambda\left(1 - \sum_y \pi_y\right). \tag{D.4}$$

At an optimum, the first-order condition with respect to each π_y gives

$$R_y - \beta\left(\log\frac{\pi_y}{\pi_y^{\mathrm{ref}}} + 1\right) - \lambda = 0. \tag{D.5}$$

Hence

$$\pi_y = \pi_y^{\mathrm{ref}} \exp\left(\frac{R_y}{\beta}\right) \exp\left(-\frac{\lambda}{\beta} - 1\right). \tag{D.6}$$

Thus the optimizer must be proportional to the reference policy exponentially tilted by the reward:

$$\pi_y \mathbb{P}opto \pi_y^{\mathrm{ref}} \exp\left(\frac{R_y}{\beta}\right). \tag{D.7}$$

The proportionality constant is determined by normalization. Imposing $\sum_y \pi_y = 1$ yields

$$\exp\left(-\frac{\lambda}{\beta} - 1\right) = \left[\sum_y \pi_y^{\mathrm{ref}} \exp\left(\frac{R_y}{\beta}\right)\right]^{-1} = \frac{1}{Z(x)}. \tag{D.8}$$

Therefore,

$$\pi^\star(y \mid x) = \frac{\pi_{\mathrm{ref}}(y \mid x) \exp(R(x,y)/\beta)}{Z(x)}, \tag{D.9}$$

where

$$Z(x) = \sum_y \pi_{\mathrm{ref}}(y \mid x) \exp\left(\frac{R(x,y)}{\beta}\right). \tag{D.10}$$

Finally, the reward term is linear in π, while $-D_{\mathrm{KL}}(\pi \| \pi_{\mathrm{ref}})$ is strictly concave in π on the support of π_{ref}. Since $\beta > 0$ and the feasible set is convex, $J_x(\pi)$ is strictly concave. Hence the stationary point derived above is the unique global maximizer. □

D.2 Prefix-extension characterization and soft-Q connection

Theorem D.2. *For a fixed prompt x and $\beta > 0$, let $\pi^\star$ be the sequence-level Gibbs optimum in Theorem D.1. For any prefix s_t with $\pi_{\mathrm{ref}}(s_t \mid x) > 0$ and any valid extension $y_{t:t+k}$, the optimal transition probability is*

$$\pi^\star(y_{t:t+k} \mid s_t) = \pi_{\mathrm{ref}}(y_{t:t+k} \mid s_t) \frac{h^\star(s_{t+k})}{h^\star(s_t)}, \tag{D.11}$$

where s_{t+k} is the resulting extended prefix and

$$h^\star(s) = \mathbb{E}_{y \sim \pi_{\mathrm{ref}}(\cdot|s)} \left[\exp\left(\frac{R(x,y)}{\beta}\right)\right]. \tag{D.12}$$

Proof. We first derive the optimal one-step transition from a prefix s_t. Let $V^\star(s_t)$ denote the optimal continuation value from s_t:

$$V^\star(s_t) = \sup_{\pi(\cdot|s_t)} \mathbb{E}_{y \sim \pi(\cdot|s_t)} \left[R(x,y) - \beta \log \frac{\pi(y \mid s_t)}{\pi_{\mathrm{ref}}(y \mid s_t)}\right], \tag{D.13}$$

where y denotes a complete response extending s_t.

Starting from s_t, the policy first selects the next token y_{t+1} and then continues from the resulting prefix $s_{t+1} = \{x, y_1, \ldots, y_t, y_{t+1}\}$. Accordingly, the continuation objective admits the following soft dynamic programming recursion:

$$V^\star(s_t) = \sup_{\pi(\cdot|s_t)} \left\{ \sum_{y_{t+1}} \pi(y_{t+1} \mid s_t) V^\star(s_{t+1}) - \beta D_{\mathrm{KL}}(\pi(\cdot \mid s_t) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid s_t)) \right\}, \tag{D.14}$$

where $s_{t+1} = \{x, y_1, \ldots, y_t, y_{t+1}\}$ is the successor prefix induced by y_{t+1}.

To solve the one-step optimization, we formulate the corresponding Lagrangian as:

$$\mathcal{L}(\pi, \lambda) = \sum_{y_{t+1}} \pi(y_{t+1} \mid s_t) V^\star(s_{t+1}) - \beta \sum_{y_{t+1}} \pi(y_{t+1} \mid s_t) \log \frac{\pi(y_{t+1} \mid s_t)}{\pi_{\text{ref}}(y_{t+1} \mid s_t)} + \lambda \left(1 - \sum_{y_{t+1}} \pi(y_{t+1} \mid s_t)\right). \tag{D.15}$$

Differentiating the Lagrangian with respect to $\pi(y_{t+1} \mid s_t)$ and setting the derivative to zero gives

$$V^\star(s_{t+1}) - \beta \left(\log \frac{\pi(y_{t+1} \mid s_t)}{\pi_{\text{ref}}(y_{t+1} \mid s_t)} + 1\right) - \lambda = 0. \tag{D.16}$$

Solving for $\pi(y_{t+1} \mid s_t)$ yields

$$\pi^\star(y_{t+1} \mid s_t) \mathbb{P}opto \pi_{\text{ref}}(y_{t+1} \mid s_t) \exp\left(\frac{V^\star(s_{t+1})}{\beta}\right). \tag{D.17}$$

The normalization condition then gives

$$\pi^\star(y_{t+1} \mid s_t) = \frac{\pi_{\text{ref}}(y_{t+1} \mid s_t) \exp(V^\star(s_{t+1})/\beta)}{\sum_{y'} \pi_{\text{ref}}(y' \mid s_t) \exp(V^\star(s_{t+1}(y'))/\beta)}. \tag{D.18}$$

where $s_{t+1}(y')$ denotes the successor prefix obtained by appending the candidate token y' to s_t.

Substituting the optimal policy above back into the soft dynamic programming equation gives

$$V^\star(s_t) = \beta \log \mathbb{E}_{y_{t+1} \sim \pi_{\text{ref}}(\cdot \mid s_t)} \left[\exp\left(\frac{V^\star(s_{t+1})}{\beta}\right)\right]. \tag{D.19}$$

Equivalently,

$$\exp\left(\frac{V^\star(s_t)}{\beta}\right) = \mathbb{E}_{y_{t+1} \sim \pi_{\text{ref}}(\cdot \mid s_t)} \left[\exp\left(\frac{V^\star(s_{t+1})}{\beta}\right)\right]. \tag{D.20}$$

Recursively applying this relation over the remaining tokens, together with the terminal condition $V^\star(y) = R(x, y)$ for a completed response y, yields

$$\exp\left(\frac{V^\star(s_t)}{\beta}\right) = \mathbb{E}_{y \sim \pi_{\text{ref}}(\cdot \mid s_t)} \left[\exp\left(\frac{R(x, y)}{\beta}\right)\right]. \tag{D.21}$$

For convenience, denote the right-hand side by

$$h^\star(s_t) = \mathbb{E}_{y \sim \pi_{\text{ref}}(\cdot \mid s_t)} \left[\exp\left(\frac{R(x, y)}{\beta}\right)\right]. \tag{D.22}$$

Hence,

$$\exp\left(\frac{V^\star(s_t)}{\beta}\right) = h^\star(s_t). \tag{D.23}$$

Returning to the one-step optimal policy, its denominator is $\exp(V^\star(s_t)/\beta)$, while its numerator contains $\exp(V^\star(s_{t+1})/\beta)$. Therefore,

$$\pi^\star(y_{t+1} \mid s_t) = \pi_{\text{ref}}(y_{t+1} \mid s_t) \frac{h^\star(s_{t+1})}{h^\star(s_t)}. \tag{D.24}$$

Applying the one-step relation along $y_{t:t+k} = \{y_{t+1}, \ldots, y_{t+k}\}$ gives

$$\begin{aligned} \pi^\star(y_{t:t+k} \mid s_t) &= \mathbb{P}od_{i=1}^{k} \pi^\star(y_{t+i} \mid s_{t+i-1}) \\ &= \mathbb{P}od_{i=1}^{k} \pi_{\text{ref}}(y_{t+i} \mid s_{t+i-1}) \frac{h^\star(s_{t+i})}{h^\star(s_{t+i-1})} \\ &= \pi_{\text{ref}}(y_{t:t+k} \mid s_t) \frac{h^\star(s_{t+k})}{h^\star(s_t)}, \end{aligned} \tag{D.25}$$

where the intermediate $h^\star$ terms cancel. □

D.3 BINARY-REWARD GIBBS CORRECTION

Corollary D.3. *Suppose $R(x, y) \in \{0, 1\}$ and define*

$$p_{\mathrm{ref}}(s_t) = \mathbb{P}_{y\sim\pi_{\mathrm{ref}}(\cdot|s_t)}\big[R(x, y) = 1\big]. \tag{D.26}$$

Then

$$h^\star(s_t) = 1 + p_{\mathrm{ref}}(s_t)\left(e^{1/\beta} - 1\right). \tag{D.27}$$

Consequently, for any reference-supported extension $y_{t:t+k}$ with resulting prefix s_{t+k}, an unnormalized optimal correction is

$$g^\star(s_t, y_{t:t+k}) = \log h^\star(s_{t+k}) = \log\Big[1 + p_{\mathrm{ref}}(s_{t+k})\left(e^{1/\beta} - 1\right)\Big]. \tag{D.28}$$

Proof. By the definition of $h^\star$ and the binary reward $R(x, y) \in \{0, 1\}$,

$$h^\star(s_t) = \mathbb{E}_{y\sim\pi_{\mathrm{ref}}(\cdot|s_t)}\left[e^{R(x,y)/\beta}\right] = 1 + p_{\mathrm{ref}}(s_t)\left(e^{1/\beta} - 1\right). \tag{D.29}$$

For an extension $y_{t:t+k}$, Theorem D.2 gives the optimal correction factor $h^\star(s_{t+k})/h^\star(s_t)$. Since $h^\star(s_t)$ is constant across all extensions from the same prefix, the corresponding unnormalized log-correction is

$$g^\star(s_{t+k}) = \log h^\star(s_{t+k}) = \log\Big[1 + p_{\mathrm{ref}}(s_{t+k})\left(e^{1/\beta} - 1\right)\Big]. \tag{D.30}$$

□

D.4 THE BIAS CAUSED BY SAMPLING ERROR

Before starting our analysis, we first clarify some notations. For the training prompts $\{x_i\}_{i=1}^N$, consider the parameterized form of the KL-regularized objective in (1):

$$\mathcal{L}(\theta) := \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{y\sim\pi_\theta(\cdot|x_i)}\left[R(x_i, y) - \beta\log\frac{\pi_\theta(y \mid x_i)}{\pi_{\mathrm{ref}}(y \mid x_i)}\right]. \tag{D.31}$$

The score-function identity gives

$$\nabla_\theta\mathcal{L}(\theta) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}_{y\sim\pi_\theta(\cdot|x_i)}\left[\left(R(x_i, y) - \beta\log\frac{\pi_\theta(y \mid x_i)}{\pi_{\mathrm{ref}}(y \mid x_i)}\right)\nabla_\theta\log\pi_\theta(y \mid x_i)\right]. \tag{D.32}$$

We assume that the Gibbs optimum in (2) is realizable: there exists $\theta^\star$ such that $\pi_{\theta^\star}(\cdot \mid x_i) = \pi^\star(\cdot \mid x_i)$ for every i. Writing $\mathcal{L}^\star := \mathcal{L}(\theta^\star)$, which is the global optimal value over the parameterized policy class, and equality holds only when the induced policies agree with the target optimum on all training prompts.

Assumption 1 (Local PL)**.** *There exist $L > 0$, $\mu > 0$, and $\Delta_{\mathrm{PL}} > 0$ such that $\mathcal{L}$ is L-smooth and*

$$\frac{1}{2}\left\|\nabla_\theta\mathcal{L}(\theta)\right\|_2^2 \geq \mu\left(\mathcal{L}^\star - \mathcal{L}(\theta)\right) \tag{D.33}$$

whenever $\mathcal{L}^\star - \mathcal{L}(\theta) \leq \Delta_{\mathrm{PL}}$.

We assume initialization satisfies

$$\mathcal{L}^\star - \mathcal{L}(\theta_0) \leq \Delta_{\mathrm{PL}}. \tag{D.34}$$

Then, under the gradient ascent with learning rate η

$$\theta_{t+1} = \theta_t + \eta\nabla_\theta\mathcal{L}(\theta_t), \tag{D.35}$$

we have the following theorem, which indicates that the gradient descent can find target optimum.

Theorem D.4. *Under Assumption 1 the θ_t obtained by gradient ascent with $0 < \eta \leq \min\{1/L, 1/\mu\}$ satisfying*

$$\mathcal{L}^\star - \mathcal{L}(\theta_t) \leq (1 - \eta\mu)^t\left(\mathcal{L}^\star - \mathcal{L}(\theta_0)\right). \tag{D.36}$$

Proof. By L-smoothness and (D.35),

$$\mathcal{L}(\theta_{t+1}) \geq \mathcal{L}(\theta_t) + \eta\left(1 - \frac{L\eta}{2}\right)\|\nabla_\theta \mathcal{L}(\theta_t)\|_2^2 \tag{D.37}$$

$$\geq \mathcal{L}(\theta_t) + \frac{\eta}{2}\|\nabla_\theta \mathcal{L}(\theta_t)\|_2^2, \tag{D.38}$$

where the second inequality uses $\eta \leq 1/L$. Hence the objective is nondecreasing. The initialization condition (D.34) and induction therefore show that every iterate remains in the region where (D.33) holds. Combining that condition with (D.38) gives

$$\mathcal{L}^\star - \mathcal{L}(\theta_{t+1}) \leq (1 - \eta\mu)\left(\mathcal{L}^\star - \mathcal{L}(\theta_t)\right). \tag{D.39}$$

Iterating this inequality proves our conclusion. □

At iteration t, a finite-batch update draws $y_{i,1}, \ldots, y_{i,B}$ independently from $\pi_{\theta_t}(\cdot \mid x_i)$ for every i and uses

$$\hat{g}_t := \frac{1}{NB}\sum_{i=1}^{N}\sum_{b=1}^{B}\left(R(x_i, y_{i,b}) - \beta \log \frac{\pi_{\theta_t}(y_{i,b} \mid x_i)}{\pi_{\text{ref}}(y_{i,b} \mid x_i)}\right)\nabla_\theta \log \pi_{\theta_t}(y_{i,b} \mid x_i) \tag{D.40}$$

and estimate the gradient, resulting the update sequence

$$\theta_{t+1} = \theta_t + \eta\hat{.} \tag{D.41}$$

Let $\mathcal{F}_t$ contain the initialization and all randomness before the batch at iteration t is drawn, so that θ_t is $\mathcal{F}_t$-measurable. At every iteration, all NB responses are sampled afresh, independently conditional on $\mathcal{F}_t$. Thus the conditional law of $\hat{g}_t$ depends only on θ_t and is the same at every visit to the same parameter.

We write $\hat{g}(\theta)$ to denote the estimator (D.41) at a fixed parameter θ. We further assume differentiability on a common support, interchange of differentiation and expectation, and finite second moments wherever used below. In particular, we have the unbiased property of gradient estimator $\mathbb{E}[\hat{g}_t \mid \mathcal{F}_t] = \nabla\mathcal{L}(\theta_t)$. The following results shows that the condition unbiased gradient estimator has non-zero variance on optimal parameter θ^*.

Lemma D.5. *For each prompt x_t, let $C_i := \beta \log Z(x_i)$ with*

$$Z(x_i) = \mathbb{E}_{y\sim\pi_{\text{ref}}(\cdot|x_i)}[e^{R(x_i,y)/\beta}], \tag{D.42}$$

and define the Fisher information matrix

$$\mathcal{I}_i(\theta^\star) = \mathbb{E}_{y\sim\pi_{\theta^\star}(\cdot|x_i)}\left[\nabla_\theta \log \pi_{\theta^\star}(y \mid x_i)\nabla_\theta \log \pi_{\theta^\star}(y \mid x_i)^\mathsf{T}\right]. \tag{D.43}$$

We have

$$\text{Cov}(\hat{g}(\theta^\star)) = \frac{1}{N^2 B}\sum_{i=1}^{N} C_i^2 \mathcal{I}_i(\theta^\star). \tag{D.44}$$

Proof. From the optimal Gibbs policy (2) we have

$$R(x_i, y) - \beta \log \frac{\pi_{\theta^\star}(y \mid x_i)}{\pi_{\text{ref}}(y \mid x_i)} = \beta \log Z(x_i) = C_i. \tag{D.45}$$

Due to the log-trick, we have

$$\mathbb{E}_{\pi_{\theta^\star}(\cdot|x_i)}[\nabla_\theta \log \pi_{\theta^\star}(y \mid x_i)] = 0. \tag{D.46}$$

Then we have

$$\begin{aligned}\text{Cov}(\widehat{g}(\theta^\star)) = \frac{1}{N^2B^2}\Bigg[&\sum_{i,b} C_i^2 \mathbb{E}[\nabla_\theta \log \pi_{\theta^\star}(y_b \mid x_i)\nabla_\theta \log \pi_{\theta^\star}(y_b \mid x_i)^\mathsf{T}] \\ &+ \sum_{(i,b)\neq(j,c)} C_i C_j\, \mathbb{E}[\nabla_\theta \log \pi_{\theta^\star}(y_b \mid x_i)\nabla_\theta \log \pi_{\theta^\star}(y_c \mid x_j)^\mathsf{T}]\Bigg] \\ = \frac{1}{N^2 B}&\sum_{i=1}^{N} C_i^2 \mathcal{I}_i(\theta^*).\end{aligned} \tag{D.47}$$

The second equality is from the independence. Then we prove our conclusion. □

Next, we assume that there exists $C_i \neq 0$ and $\operatorname{tr} \mathcal{I}_i(\theta^\star) > 0$ for at least one prompt, then we have the non-zero gradient variance

$$\sigma_\star^2 = \mathbb{E}\left[\|\hat{g}(\theta^\star) - \nabla_\theta \mathcal{L}(\theta^\star)\|_2^2\right] = \frac{1}{N^2 B} \sum_{i=1}^{N} C_i^2 \operatorname{tr} \mathcal{I}_i(\theta^\star) > 0. \tag{D.48}$$

Next, we will show that θ^* is not a stationary point of gradient ascent (D.41), so that practical trained policy model will not converge to the target optimum. Before that, we need the following lemma.

Lemma D.6. *Suppose that $q_\star = \mathbb{P}(\hat{g}(\theta^\star) = 0) < 1$. Let τ be a stopping time such that $\theta_\tau = \theta^\star$ on $\{\tau < \infty\}$. Define*

$$T_\tau := \inf\{m \geq 1 : \theta_{\tau+m} \neq \theta^\star\}.$$

Then

$$\mathbb{P}(T_\tau < \infty \mid \mathcal{F}_\tau) = 1, \qquad \mathbb{E}[T_\tau \mid \mathcal{F}_\tau] = \frac{1}{1 - q_\star} < \infty. \tag{D.49}$$

Proof. Due to the positive variance (D.48), we have $q_\star < 1$. Since $\eta > 0$ in (D.41) we have

$$\mathbb{P}(\theta_{t+1} = \theta^* \mid \theta_t = \theta^*) = \mathbb{P}(\hat{g}(\theta^\star) = 0) = q_\star. \tag{D.50}$$

Thus, iteratively using this equality obtains that

$$\mathbb{P}(T_\tau > m \mid \mathcal{F}_\tau) = q_\star^m \tag{D.51}$$

Taking $m \to \infty$ proves our first conclusion in (D.49). Then by invoking that $E[X] = \sum_{m>0} \mathbb{P}(X > m)$ for positive random variable X, we have

$$\mathbb{E}[T_\tau \mid \mathcal{F}_\tau] = \sum_{m=0}^{\infty} q_\star^m = \frac{1}{1 - q_\star}.$$

which proves our second conclusion in (D.49). □

From the lemma, we know that the optimum parameter θ^* is therefore not an absorbing state. Next, we rule out the asymptotic convergence, saying that the iterate θ_t will explicitly stay away from θ^*.

Theorem D.7. *Define $v(\theta) = \mathbb{E}[\|\hat{g}(\theta) - \nabla \mathcal{L}(\theta)\|_2^2]$. Suppose that v is continuous at $\theta^\star$, and there exists are a neighborhood U of $\theta^\star$ and a constant $M_4 < \infty$ such that*

$$\sup_{\theta \in U} \mathbb{E}\left[\|\hat{g}(\theta)\|_2^4\right] \leq M_4. \tag{D.52}$$

Then there exist $r > 0$ and $p_0 > 0$ such that, whenever $\|\theta_t - \theta^\star\|_2 \leq r$,

$$\mathbb{P}(\|\theta_{t+1} - \theta^\star\|_2 > r \,|\, \mathcal{F}_t) \geq p_0. \tag{D.53}$$

Moreover, we have

1. $\mathbb{P}(\|\theta_t - \theta^\star\|_2 > r$ *infinitely often*$) = 1$
2. $\mathbb{P}(\lim_{t\to\infty} \theta_t = \theta^\star) = 0$
3. $\limsup_{t\to\infty} \mathbb{P}(\|\theta_t - \theta^\star\|_2 > r) \geq \frac{p_0}{1+p_0} > 0$

Thus the iterates have zero probability of converging to this fixed optimal parameter and do not converge to it in probability.

Proof. By continuity, without loss of generality, we can assume U that

$$v(\theta) \geq \sigma^2 = \sigma_\star^2/2 > 0 \tag{D.54}$$

throughout U (otherwise we can shrink σ). Write $\xi_t := \hat{g}_t - \nabla \mathcal{L}(\theta_t)$. Then for $\theta_t \in U$, we have

$$\mathbb{E}[\|\hat{g}_t\|_2^2 \mid \mathcal{F}_t] = \|\nabla \mathcal{L}(\theta_t)\|_2^2 + v(\theta_t) \geq \sigma^2. \tag{D.55}$$

Apply the conditional Paley–Zygmund inequality (Kallenberg, 1997) to the nonnegative variable $= \|\hat{g}_t\|_2^2$ we obtain

$$\begin{aligned}
\mathbb{P}\left(\|\hat{g}_t\|_2 \geq \frac{\sigma}{\sqrt{2}} \,\middle|\, \mathcal{F}_t \right) &\geq \mathbb{P}\big(\|\hat{g}_t\|_2^2 \geq \tfrac{1}{2}\mathbb{E}[\|\hat{g}_t\|_2^2 \mid \mathcal{F}_t] \,\big|\, \mathcal{F}_t \big) \\
&\geq \frac{\big(\mathbb{E}[\|\hat{g}_t\|_2^2 \mid \mathcal{F}_t]\big)^2}{4\mathbb{E}[\|\hat{g}_t\|_2^4 \mid \mathcal{F}_t]} && \text{(D.56)} \\
&\geq \frac{\sigma^4}{4M_4} && \text{(D.57)} \\
&= p_0 && \text{(D.58)} \\
&> 0. && \text{(D.59)}
\end{aligned}$$

Thus, we prove our conclusion (D.53).

Next, we choose $r > 0$ such that $\overline{B}_r(\theta^\star) \subset U$ and $r \leq \eta\sigma/(4\sqrt{2})$, we can choose such r since we can shrink σ. Then if $\|\theta_t - \theta^\star\|_2 \leq r$ and the event (D.59) occurs, then

$$\|\theta_{t+1} - \theta^\star\|_2 \geq \eta\|\hat{g}_t\|_2 - \|\theta_t - \theta^\star\|_2 \geq \frac{\eta\sigma}{\sqrt{2}} - r \geq 3r > r. \tag{D.60}$$

This proves our conclusion (D.53).

Next, for a fixed integer $K \geq 0$, iterated conditioning shows that, on $\{\theta_K \in \overline{B}_r(\theta^\star)\}$,

$$\mathbb{P}\left(\bigcap_{j=1}^{m} \{\theta_{K+j} \in \overline{B}_r(\theta^\star)\} \,\middle|\, \mathcal{F}_K \right) \leq (1 - p_0)^m. \tag{D.61}$$

Letting $m \to \infty$ shows that the probability of remaining in the ball at all times from K onward is zero. A countable union over K therefore gives

$$\mathbb{P}\left(\bigcup_{K \geq 0} \bigcap_{t \geq K} \theta_t \in \overline{B}_r(\theta^\star) \right) = 0,$$

which proves our conclusions 1 and 2.

Finally, let

$$a_t = \mathbb{P}(\|\theta_t - \theta^\star\|_2 > r), \tag{D.62}$$

(D.53) implies that

$$a_{t+1} \geq p_0(1 - a_t). \tag{D.63}$$

Writing

$$a = \limsup_t a_t \tag{D.64}$$

then for any $\varepsilon > 0$ we have $a_t \leq a + \varepsilon$ eventually, and hence $a_{t+1} \geq p_0(1 - a - \varepsilon)$. Taking the upper limit and then $\varepsilon \downarrow 0$ yields $a \geq p_0(1 - a)$, which proves conclusion 3. □

The following proposition gives a token-level characterization of deviations from the sequence-level optimum.

Proposition 2. *Given prompt x. If $\pi_\theta(\cdot \mid x) \neq \pi^\star(\cdot \mid x)$ on the support of π_{ref}, then the token-level Gibbs condition probability* (3) *cannot hold at every prefix.*

Proof. Suppose instead that $D_{\mathrm{tok}} = 0$ for every candidate-token pair at every prefix with positive reference probability. (8) then implies that, at each prefix s_t, the ratio

$$\frac{\pi_\theta(y_{t+1} \mid s_t)}{\pi_{\mathrm{ref}}(y_{t+1} \mid s_t)h^\star(s_{t+1})} \tag{D.65}$$

is the same for every candidate token. Then (3.1) and the law of total expectation give $h^\star(s_t) = \sum_{y_{t+1}} \pi_{\text{ref}}(y_{t+1} \mid s_t) h^\star(s_{t+1})$. Normalization therefore makes the common ratio $1/h^\star(s_t)$. Therefore the policy satisfies the one-token case of (3) at every prefix. For any complete response $y = (y_1, \ldots, y_T)$, the autoregressive factorization then gives

$$\pi_\theta(y \mid x) = \prod_{t=0}^{T-1} \pi_{\text{ref}}(y_{t+1} \mid s_t) \frac{h^\star(s_{t+1})}{h^\star(s_t)} \tag{D.66}$$

$$= \pi_{\text{ref}}(y \mid x) \frac{h^\star(s_T)}{h^\star(x)} \tag{D.67}$$

$$= \pi_{\text{ref}}(y \mid x) \frac{\exp(R(x,y)/\beta)}{Z(x)} \tag{D.68}$$

$$= \pi^\star(y \mid x). \tag{D.69}$$

This contradicts the assumed sequence-level mismatch. Hence proves our conclusion. □

In particular, any same-reward pair with $D_{\text{seq}} > 0$ certifies the premise of Proposition 2 and therefore implies a token-level violation at some prefix.

D.5 Bounded reference-relative reweighting

Proposition 1 (Preservation of next-token probability bounds)**.** *Let y_{t+1} and y'_{t+1} be two candidate next tokens at the same prefix s_t. Suppose$c_{\text{low}} \le \frac{\pi_{\text{ref}}(y_{t+1}|s_t)}{\pi_{\text{ref}}(y'_{t+1}|s_t)} \le c_{\text{up}}$, where $0 < c_{\text{low}} \le c_{\text{up}} < \infty$. Then the token policy in* (9) *satisfies*

$$e^{-1/\beta} c_{\text{low}} \le \frac{\pi_\phi(y_{t+1} \mid s_t)}{\pi_\phi(y'_{t+1} \mid s_t)} \le e^{1/\beta} c_{\text{up}}. \tag{15}$$

Proof. For a candidate next token a at prefix s_t, write

$$w_\phi(s_t, a) = 1 + p_\phi(s_t, a)(e^{1/\beta} - 1).$$

Since $p_\phi(s_t, a) \in [0, 1]$, every weight lies in $[1, e^{1/\beta}]$. The two tokens y_{t+1} and y'_{t+1} share the same normalizer in (9), so

$$\frac{\pi_\phi(y_{t+1} \mid s_t)}{\pi_\phi(y'_{t+1} \mid s_t)} = \frac{\pi_{\text{ref}}(y_{t+1} \mid s_t)}{\pi_{\text{ref}}(y'_{t+1} \mid s_t)} \frac{w_\phi(s_t, y_{t+1})}{w_\phi(s_t, y'_{t+1})}.$$

The weight ratio lies in $[e^{-1/\beta}, e^{1/\beta}]$. Combining this with the assumed reference-policy ratio bounds proves the claim. □