# PRINCIPLED REINFORCEMENT LEARNING WITH VERIFIABLE REWARDS: UNDERSTANDING LLM REASONING VIA DIFFUSIVE FLOW

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

## ABSTRACT

Large language models (LLMs) increasingly rely on reinforcement learning with verifiable rewards (RLVR) for post-training on complex reasoning tasks. While most RLVR methods are policy-based, value-based approaches have shown promise but remain underexplored in stability, formulation, and theory. In this work, we revisit value-based RLVR through a smoothed consistency framework that stabilizes LLM fine-tuning. With reward augmentation, our framework generalizes the standard KL-regularized objective, connects value-based and policy-based methods, and yields a practical joint policy–value objective using only terminal verifiable rewards. To explain its effectiveness for LLM reasoning, we develop a reasoning-flow characterization over a reasoning-directed acyclic graph (DAG), showing how the learned RLVR objective induces latent stepwise credit assignment and reshapes probability mass over reasoning trajectories. This characterization reveals how the fine-tuned LLM favors high-reward reasoning paths while preserving diversity in structured reasoning. On the theoretical side, we establish a rigorous regret analysis showing that our algorithm achieves sample-complexity guarantees and is robust to the LLM policy drift. Experiments on mathematical reasoning benchmarks show consistent gains over pretrained LLMs and demonstrate that our algorithm is competitive with strong baselines.

## 1 INTRODUCTION

Large language models (LLMs) have emerged as the central foundation of modern natural language understanding and generation (Brown et al., 2020; Ouyang et al., 2022) with growing applications in tasks that demand complex reasoning (Cobbe et al., 2021; Hendrycks et al., 2021; Uesato et al., 2022; Shao et al., 2024) in which success depends not only on the final response but also on the quality of the intermediate reasoning path that leads to it. This has made reinforcement learning (RL) a natural paradigm for post-training, especially in domains with verifiable rewards, where correctness can be assessed by explicit rules rather than by a separately trained reward model. Therefore, reinforcement learning with verifiable rewards (RLVR) (Lambert et al., 2024; Guo et al., 2025; Yue et al., 2025a) becomes a fundamental approach to incentivize the reasoning capacity of LLMs.

Although most of the existing RLVR methods are policy-based algorithms (Bai et al., 2022; Ouyang et al., 2022; Shao et al., 2024; Guo et al., 2025), they are often sensitive to rollout quality, rely on complex engineering heuristics such as density-ratio clipping, and make only limited use of the stepwise value structure of intermediate reasoning steps. In comparison, value-based RLVR methods have recently attracted attention (Wang et al., 2025; Yuan et al., 2025a;b; Yue et al., 2025b). The fundamental principle behind this streamlining of the works is to rely on Bellman-type consistency equations to learn a value function, then use it to guide the fine-tuning of LLMs. Despite the great successes of value-based methods, this direction remains underexplored in a principled manner for LLM reasoning. The major possible reasons include: (1) Value-function methods, which rely on iterative Bellman updates, can be more prone to instability than policy-gradient methods, which directly optimize a policy objective; (2) Although many of them start with the rigorous Bellman-type consistency equations, the actual training objective is a biased surrogate due to the inevitable adaptation to LLM settings. Therefore, there remain unresolved challenges in developing value-based RLVR methods that are simultaneously principled, stable, and theoretically justified.



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.

Motivated by the aforementioned challenges, we revisit the value-based RLVR algorithm from a new perspective and reformulate Bellman-type consistency equations through a smoothed consistency framework, thereby improving optimization stability during LLM fine-tuning. Building on this view, we further introduce a reward augmentation technique and demonstrate that the developed smoothed consistency framework is a general extension of the widely used Kullback–Leibler (KL) regularized post-training objective for LLM reasoning in both value-based and policy-based methods. This provides a route to incorporate key policy-constraint techniques, e.g., trust-region policy updates, from policy-based methods to value-based methods to stabilize autoregressive reasoning updates and permit more aggressive learning without abrupt policy drift. For the terminal-reward setting of LLM reasoning, this construction admits a trajectory-level consistency relation that jointly characterizes the policy and value function without requiring observable stepwise rewards, yielding a practical joint learning objective. The resulting trajectory consistency connects to classical path-consistency formulations in entropy- and KL-regularized RL (Nachum et al., 2017a), while here it arises from a smoothed, reward-augmented formulation of value-based RLVR and provides the basis for our reasoning-flow and regret analyses.

To understand why our algorithm — broadly, value-based RLVR — is effective at incentivizing reasoning in LLMs, we develop a reasoning-directed acyclic graph (DAG) representation to characterize the reasoning flow induced by our RLVR objective. The proposed framework characterizes the LLM reasoning processes as diffusive flows, which shed light on the role of the value-based method: the value function provides latent stepwise credit assignment, and the reasoning flow policy in the reasoning DAG, i.e., a fine-tuned LLM policy, learns to allocate probability mass toward high-reward solutions under prior flow guidance. This explains that our algorithm learns a structured, reward-shaped distribution over plausible reasoning flows to enable structured reasoning diversity, so-called "divergent reasoning with rules", with adaptive diversity-preserving and mode-seeking behavior across multiple desired flows.

- We introduce a smoothed consistency reasoning formulation with reward augmentation. This new formulation generalizes the KL-regularized objective, bridges value-based and policy-based RLVR methods, and derives a practical joint policy–value objective using only terminal verifiable rewards.
- We develop a reasoning-flow characterization of value-based RLVR over a reasoning DAG. Starting from the learned RLVR objective rather than imposing an a priori flow-matching objective, we provide insight into how value-based RLVR incentivizes reasoning in LLMs.
- We establish rigorous regret analyses for the proposed algorithm, the gold-standard criterion for evaluating RL-based algorithms. We prove that our algorithm learns a fine-tuned LLM policy competitive with a near-optimal fine-tuned policy, and also theoretically justify the necessity of the two major components in our algorithm related to policy constraint.
- We validate the proposed algorithm on diverse mathematical reasoning benchmarks, showing consistent gains over pretrained LLMs and competitiveness with strong baselines. We further conduct ablation studies to verify the contribution of the algorithmic components.

## 2 PRELIMINARIES

We consider LLM reasoning tasks as conditional generation (Yu et al., 2025). Let $\rho$ denote a prompt distribution and $x \sim \rho$ a sampled prompt. We model autoregressive generation as a deterministic episodic decision process with a fixed maximum horizon $H$. In this work, we write $\pi(\cdot \mid s)$ for the token-level LLM policy and $\pi(\cdot \mid x)$ for the induced distribution over length-$H$ trajectories starting from $s_0$, where $s_0 = (x, \varnothing)$ is the prompt-induced initial state. Given $x$, generation induces a token sequence $y = (a_0, \ldots, a_{H-1})$ with autoregressive factorization $\pi(y \mid x) = \prod_{i=0}^{H-1} \pi(a_i \mid s_i)$, where $s_i = (x, a_{<i})$ is the state at step $i$. Let $\mathcal{S}_{\mathrm{ar}}$ denote the set of prompt-conditioned autoregressive states, and let $\bar{\mathcal{S}} = \mathcal{S}_{\mathrm{ar}} \cup \perp$ be the extended state space, where $\perp$ is an auxiliary absorbing state. The LLM policy acts only on states in $\mathcal{S}_{\mathrm{ar}}$. Once <EOS> token is produced or the maximum horizon is reached, the process transitions to an auxiliary absorbing state $\perp$, used only for terminal normalization; after absorption, rewards and KL contributions are zero. During fine-tuning of LLM policy, we consider a LLM policy class $\Pi$ and a state-value function class $\mathcal{V}$, where each $V \in \mathcal{V}$ maps the extended state space $\bar{\mathcal{S}}$ to a scalar and satisfies the terminal normalization $V(\perp) = 0$. For ease of presentation,

we write $V(x)$ as shorthand for $V(s_0)$ whenever unambiguous. We use $|\Pi|$ and $|\mathcal{V}|$ to denote the complexity of the policy and value classes, respectively, both of which are represented by LLMs. In RLVR, we typically observe a terminal verifiable reward $R(x, y) \in \{0, 1\}$ only after the generated completion is evaluated. Let $\pi_{\text{ref}}$ denote a fixed reference LLM policy. Given $R(x, y)$, the standard KL-regularized RLVR objective (Guo et al., 2025; Zheng et al., 2025) is

$$J(\pi) = \mathbb{E}_{x\sim\rho,\, y\sim\pi(\cdot|x)}\left[R(x, y) - \beta \sum_{i=0}^{H-1} \log \frac{\pi(a_i \mid s_i)}{\pi_{\text{ref}}(a_i \mid s_i)}\right], \tag{2.1}$$

where $\beta > 0$ is a regularization parameter. Given the training data, the fine-tuned LLM policy is learned by maximizing a sample estimate of $J(\pi)$.

## 3 Main Algorithm

In this section, we introduce a novel view of the value-based methods that enables the development of our algorithm. To do so, we first revisit the value-based method for LLM reasoning. Value-based RLVR methods differ in implementation but share the same principle: solve a Bellman-type consistency equation to learn a (near)-optimal value function and use it to guide LLM policy updates.

$$V(s) = \max_{\pi\in\Pi} \mathbb{E}_{a\sim\pi(\cdot|s)}[r(s, a) + \gamma\mathbb{E}_{s'|s,a}(V(s'))]. \tag{3.1}$$

In particular, to solve equation 3.1, a straightforward approach is to minimize the discrepancy between the two sides of the equation using an $L_2$ loss function with LLM reasoning training data. However, we find that such a paradigm will easily lead to optimization difficulties due to the non-smoothness of the $\max$ operator, particularly when performing non-convex fine-tuning over a massive set of LLM parameters. To mitigate this issue, we draw on the Nesterov smoothing technique (Nesterov, 2005; Dai et al., 2018) to address the non-smoothness in the Bellman-type consistency equation 3.1. Specifically, we propose to add a smoothing component $\phi(\cdot)$ into the RHS of the equation 3.1, i.e.,

$$\max_{\pi\in\Pi} \mathbb{E}_{a\sim\pi(\cdot|s)} \left[r(s, a) + \gamma\mathbb{E}_{s'|s,a} \left[V(s')\right] + \beta\phi(\pi(a|s))\right], \tag{3.2}$$

where $\beta \geq 0$ controls the degree of smoothing. It has been shown that when $\phi(\cdot)$ satisfies some conditions, the hard-$\max$ operation in the above formulation can be smoothed by the Fenchel transformation theorem (Hiriart-Urruty & Lemaréchal, 2012). If we take a close look at the smooth Bellman-type consistency equation 3.2, it is easy to find that the stepwise reward function $r(s, a)$, although it is non-observable in LLM reasoning, as we can only observe the terminal reward, the reward function is invariant to the equation itself. That is, any transformation of the reward function does not invalidate the equation. This observation implies that we could use reward-augmentation techniques to modify the reward function to induce desired behavior during training.

Surprisingly, we find that, when we specify the $\phi(\cdot)$ as the Shannon entropy, and we further augment the reward function with an accumulative log-prob for the reference LLM policy, i.e., $R^\star(x, y) := R(x, y)+\beta\sum_{i=0}^{H-1} \log \pi_{\text{ref}}(a_i \mid s_i)$, it gives us an equivalence between the solver of the widely-used LLM reasoning post-training objective equation 2.1 and the solver of the smoothed equation 3.2. In other words, under this specified setup, our developed smoothing framework is a general extension of the widely used training objective in both policy-based and value-based methods.

This finding reveals a bridge between the policy-based method and value-based methods within our framework, and that some key techniques used only in the policy-based method can be incorporated into the value-based method via reward augmentation. One important example is the trust-region technique. In autoregressive LLM reasoning, such stabilization is crucial because a small change in token probabilities at an early step may alter the entire downstream completion, while the reward is observed only at the terminal step. As a result, without explicit control of iteration-to-iteration LLM policy drift, optimization can become unstable and cause abrupt changes in reasoning behavior. A trust-region mechanism (Schulman et al., 2015) is therefore desirable, since it constrains the new policy relative to the previous iterate and promotes more conservative updates throughout training. To achieve this, we further augment the previously augmented reward function with an additional accumulative log probability for the last-iteration policy $\pi_{t-1}$, where $t$ is the training iteration step:

$$R^\star(x, y) := R(x, y) + \beta \sum_{i=0}^{H-1} \log \pi_{\text{ref}}(a_i \mid s_i) + \zeta \sum_{i=0}^{H-1} \log \pi_{t-1}(a_i \mid s_i).$$

Following a similar principle, we show that this reward augmentation leads us to maximizing the following objective:

$$J_t(\pi) = \mathbb{E}_{x\sim\rho, y\sim\pi(\cdot|x)}\left[R(x,y) - \beta\sum_{i=0}^{H-1}\log\frac{\pi(a_i \mid s_i)}{\pi_{\text{ref}}(a_i \mid s_i)} - \zeta\sum_{i=0}^{H-1}\log\frac{\pi(a_i \mid s_i)}{\pi_{t-1}(a_i \mid s_i)}\right], \tag{3.3}$$

where the last term in the formulation explicitly performs the trust-region effect via the Lagrangian transformation. Here, although the last two terms are yielded via the reward augmentation, they play distinct roles in training and serve different purposes. The combined effects of both terms are beneficial: the first term helps maintain overfitting under the reference model, while the trust-region term improves stability and allows for a faster learning rate. In Section 5, we show that the trust-region term influences not only the stability of the optimization but also the regret guarantee.

The careful reader may find that the developed reward-augmented smoothed equation, i.e., which replaces the $r(s,a)$ with the augmented reward $r^\star(s,a)$ which is induced by $R^\star(x,y)$ in equation 3.2, is infeasible to solve directly. For most LLM reasoning tasks, responses are generated autoregressively, token by token, whereas the reward is typically observed only at the end of the full completion. Directly solving the constructed reward-augmented smoothed equation requires observations of the step-wise rewards, which is not practical.

Fortunately, our smoothed reward-augmentation algorithm can be equivalently characterized at the trajectory level, yielding three properties particularly useful for terminal-reward LLM reasoning. It simultaneously characterizes the optimal LLM policy and value function in a single equation, avoiding the standard two-stage procedure in value-based methods that first estimates the value function and then fine-tunes the LLM policy; its solution agrees with the optimum of the smoothed reward-augmentation equation; and its trajectory-level form naturally accumulates stepwise rewards as a counterpart to the terminal reward, making the resulting objective solvable for LLM reasoning tasks. We formalize this trajectory-level characterization in the following theorem.

**Theorem 3.1.** *Let $(\pi_t^\star, V_t^\star)$ denote the optimal LLM policy in the objective equation 3.3 and its corresponding optimal value function. Then, for any prompt $x$ and any completion $y$ sampled from $\pi_t^\star(\cdot \mid x)$, the following consistency equation holds*

$$V_t^\star(x) - V_t^\star(\perp) = R(x,y) - \beta\sum_{i=0}^{H-1}\log\frac{\pi_t^\star(a_i \mid s_i)}{\pi_{\text{ref}}(a_i \mid s_i)} - \zeta\sum_{i=0}^{H-1}\log\frac{\pi_t^\star(a_i \mid s_i)}{\pi_{t-1}(a_i \mid s_i)}. \tag{3.4}$$

It is easy to see that the equation in Theorem 3.1 depends entirely on the terminal reward instead of the step-wise reward. Theorem 3.1 is related to classical multi-step path-consistency conditions in entropy- and KL-regularized RL Nachum et al. (2017a;b). Here, the consistency relation is derived from our smoothed Bellman formulation with reward augmentation for value-based RLVR and explicitly incorporates both a fixed reference policy and an iteration-wise policy constraint while depending only on terminal verifiable rewards. This result subsequently yields the practical joint policy–value objective above and serves as the foundation for the reasoning-flow and regret analyses in Sections 4 and 5. Motivated by this theoretical result, we further propose to solve it by minimizing the discrepancy between the two sides of the equation, with the collected LLM reasoning training trajectories $\{x_k, y_k\}_{k=1}^n$. By some algebra, we have our training objective to minimize:

$$\hat{\mathcal{L}}_t(\pi, V) = \frac{1}{n}\sum_{k=1}^{n}\Big(V(x^k) - R(x_k,y_k) + \beta\sum_{i=0}^{H-1}\log\frac{\pi(a_i^k \mid s_i^k)}{\pi_{\text{ref}}(a_i^k \mid s_i^k)} + \zeta\sum_{i=0}^{H-1}\log\frac{\pi(a_i^k \mid s_i^k)}{\pi_{t-1}(a_i^k \mid s_i^k)}\Big)^2.$$

# 4 Diffusive Reasoning Flow

## 4.1 Reasoning-Flow Characterization of RLVR

In this section, to explain why our proposed value-based smoothed consistency algorithm can work effectively for LLM reasoning problems, we introduce a reasoning-DAG representation to derive and characterize the reasoning flow induced by our RLVR objective, including its latent credit assignment and terminal reasoning distribution. We begin by defining the reasoning DAG to establish routes for quantifying diffusive pathways.

**Definition 1** (Reasoning Directed Acyclic Graph)**.** *Fix an input prompt $x$. A reasoning directed acyclic graph (DAG) is $\mathcal{G}_x = (\mathcal{S}_x, \mathcal{E}_x)$, where $\mathcal{S}_x$ is the set of reasoning nodes and $\mathcal{E}_x$ is the set of valid transitions. It has root $s_0 = (x, \varnothing)$ and absorbing state $\perp$. Each node is a partial reasoning path $s_t = (x, z_{1:t})$, where $z_{1:t}$ is a sequence of thoughts, and terminal nodes $\mathcal{X}_x \subseteq \mathcal{S}_x$ represent completed solutions. For any nonterminal node $s_t \in \mathcal{S}_x \setminus \mathcal{X}_x$, each admissible thought $a \in \mathcal{Z}(s_t)$ induces the deterministic transition $s_{t+1} = \mathrm{append}(s_t, a) = (x, z_{1:t} \circ a), (s_t \to s_{t+1}) \in \mathcal{E}_x$. For any terminal node $\bar{x} = (x, z_{1:T}) \in \mathcal{X}_x$, the unique action* STOP *transitions to $\perp$.*

**Example** (Arithmetic Reasoning). Consider the prompt

$$x = \text{“Use the numbers } \{3, 4, 6, 11\} \text{ and basic arithmetic operations to obtain } 24.\text{”}$$

The root node is $s_0 = (x, \varnothing)$. One valid reasoning path is $\{11 - 3 = 8,\ 8 - 4 = 4,\ 6 \times 4 = 24\}$, which induces the trajectory $s_0 \to s_1 \to s_2 \to \bar{x}$, where $s_1 = (x, z_1)$, $s_2 = (x, z_{1:2})$, and $\bar{x} = (x, z_{1:3}) \in \mathcal{X}_x$ is a terminal state. This illustrates that nodes represent partial reasoning states, edges represent admissible next-thought transitions, and terminal states represent completed solutions, followed by a STOP transition to the absorbing state $\perp$.

With the definition of the reasoning DAG, we are able to treat intermediate reasoning steps as flow-carrying nodes, and the amount of flow passing through a node quantifies how strongly the policy relies on that step. We further leverage this idea by defining *reasoning flow* as the propagation of unnormalized probability mass through the DAG along feasible reasoning paths.

**Definition 2** (Reasoning Flow)**.** *Given a prompt $x$ and the reasoning DAG $\mathcal{G}_x$, let $\mathcal{T}_x$ denote the set of complete trajectories $\tau = (s_0 \to s_1 \to \cdots \to s_n \to s_f)$ in $\mathcal{G}_x$. A “reasoning flow” is any non-negative function $\mathcal{F} : \mathcal{T}_x \to \mathbb{R}_+$. It induces a measure over $\Sigma = 2^{\mathcal{T}_x}$ by $\mathcal{F}(A) = \sum_{\tau \in A} \mathcal{F}(\tau), \forall A \subseteq \mathcal{T}_x$. The pair $(\mathcal{G}_x, \mathcal{F})$ is called a “reasoning flow network”.*

To characterize how to induce the desired reasoning flow in the reasoning DAG, we introduce a new notion of reasoning reward called the *latent stepwise reward*, which is distinct from the standard *terminal reward* in RLVR. Unlike the standard terminal reward in RLVR, which evaluates only the final completed solution, the latent stepwise reward assigns credit to intermediate reasoning steps.

**Lemma 4.1** (Latent Stepwise-Reward)**.** *There exists a latent stepwise-reward function defined as:*

$$r(s, s') = \begin{cases} \beta \log \pi_{\mathrm{ref}}(s' \mid s), & s \in \mathcal{S}_x \setminus \mathcal{X}_x, \\ \log R(x, s) + \zeta \log \pi_{t-1}(s), & s \in \mathcal{X}_x \text{ and } s' = \perp, \\ 0, & s = \perp\ . \end{cases}$$

*where $\pi_{\mathrm{ref}}$ is the reference policy, and $\pi_{t-1}$ is the last-iteration policy on $\mathcal{G}_x$.*

Intuitively, Lemma 4.1 explains how the credits of the final verifiable reward should be assigned backward along diffusive pathways. This distribution is guided by three sources: the terminal reward, prior-flow information from the reference policy, and stabilization from the last-iteration policy. Roughly speaking, it specifies how desirable reasoning flows are encouraged throughout the reasoning DAG, rather than only at terminal states. We formalize this connection in the following lemma by showing how the latent stepwise reward induces unnormalized flow over nodes and edges.

**Lemma 4.2.** *For any nonterminal state $s \notin \mathcal{X}_x$, the reasoning flow satisfies*

$$\mathcal{F}_\lambda(s) = \sum_{s' \in \mathcal{A}(s)} \exp\left(\frac{r(s, s')}{\lambda}\right) \mathcal{F}_\lambda(s'), \qquad \mathcal{F}_\lambda(s \to s') = \exp\left(\frac{r(s, s')}{\lambda}\right) \mathcal{F}_\lambda(s'),$$

*where $\mathcal{A}(s)$ is a admissible successor set. For any $\bar{x} \in \mathcal{X}_x$, $\mathcal{F}_\lambda(\bar{x}) = \left(\pi_{t-1}(\bar{x})^\zeta R(x, \bar{x})\right)^{1/\lambda}$.*

Lemma 4.2 formalized how each reasoning step carries reward-shaped continuation flow and how each outgoing edge receives a portion of that flow according to its latent stepwise reward. To connect this reasoning flow into a reasoning generation policy that we are interested in Section 3, we define *reasoning flow policy*, which normalizes outgoing flow into transition probabilities and corresponds to the target policy in our LLM reasoning objective we aim to learn in the last section.

**Definition 3** (Reasoning Flow Policy)**.** *A reasoning flow policy is a Markov kernel $\pi_\theta(\cdot \mid s)$ supported on outgoing edges: for each nonterminal $s \in \mathcal{S}_x \setminus \mathcal{X}_x$, $\pi_\theta(\cdot \mid s)$ is a distribution over*

*admissible $\mathcal{A}(s)$, i.e., $\sum_{s' \in \mathcal{A}(s)} \pi_\theta(s' \mid s) = 1$. Further, the optimal policy connects conditional probabilities induced by reasoning flow $\mathcal{F}$ with the value function, i.e.,*

$$\pi^\star(s' \mid s) = \exp\Big(\tfrac{1}{\lambda} A^*_\lambda(s, s')\Big) = \frac{\mathcal{F}_\lambda(s \to s')}{\mathcal{F}_\lambda(s)}, \qquad (s \to s') \in \mathcal{E}_x, \tag{4.1}$$

*where $A^*_\lambda(s, s') := r(s, s') + \lambda \log \mathcal{F}_\lambda(s') - \lambda \log \mathcal{F}_\lambda(s)$ is an advantage function on an edge $(s \to s') \in \mathcal{E}_x$. A terminal $\bar{x} \in \mathcal{X}_x$ corresponds to a unique path $\tau(\bar{x}) = (s_0 \to s_1 \to \cdots \to s_T = \bar{x})$, hence the induced terminal marginal is $d(\bar{x}) := \prod_{t=0}^{T-1} \pi(s_{t+1} \mid s_t)$ for $\bar{x} \in \mathcal{X}_x$.*

The reasoning flow policy turns the unnormalized reasoning flow into a distribution over complete reasoning paths, thereby inducing a terminal distribution over $\mathcal{X}_x$. The central question that follows immediately is what terminal distribution this reasoning flow policy should realize. Interestingly, the terminal distribution induced by the reasoning flow policy exactly matches the policy distribution in our training objective function, motivated by Theorem 3.1. That is, it is a reward-shaped distribution guided by *prior reasoning flows* from the reference policy and is stabilized by the last-iteration policy. Thus, high-reward terminals are favored, but only through reasoning paths that remain compatible with the prior flow and the trust-region update. The following theorem validates the exact form of the induced target terminal distribution.

**Theorem 4.3** (Target Terminal Distribution)**.** *For $\zeta, \beta \geq 0$ and $\lambda = \zeta + \beta > 0$, define the unnormalized target weight $w_\lambda(\bar{x}) = \pi_{\text{ref}}(\bar{x})^{\beta/\lambda}\, \pi_{t-1}(\bar{x})^{\zeta/\lambda}\, R(x, \bar{x})^{1/\lambda}, \bar{x} \in \mathcal{X}_x$, where $\pi_{\text{ref}}(\bar{x})$ and $\pi_{t-1}(\bar{x})$ denote the sequence probabilities along the unique path $\tau(\bar{x})$ following Definition 3. The induced prior reasoning flow for reference policy is $\mathcal{F}^\pi(\bar{x}) = Z\, \pi_{\text{ref}}(\bar{x})$ for any $Z > 0$, in analogy for $\pi_{t-1}$. Then, the target terminal distribution in $\bar{x}$ is $d^*_\lambda(\bar{x}) = w_\lambda(\bar{x}) / Z_\lambda$ with $Z_\lambda = \sum_{\bar{x} \in \mathcal{X}_x} w_\lambda(\bar{x})$.*

Theorem 4.3 makes the notion of "divergent reasoning with rules" explicit: the terminal reward favors successful reasoning outcomes, while the reference and previous-iterate policies constrain which high-reward reasoning paths receive substantial probability mass. Interestingly, under the reasoning DAG framework, we can draw a connection to a popular generative training framework: GFlowNets (Bengio et al., 2021; Tiapkin et al., 2024). Following Lemma 4.1 and 4.2, if we choose the latent step-wise rewards as the so-called "backward policy" in GFlowNets, i.e., $r(s, s') = \log P_B(s \mid s')$, our reasoning DAG degenerates into GFlowNets. We can show that the induced terminal in GFlowNets under our framework is analogous to

$$\pi^*_\lambda(\tau) \propto \exp(R(x, \bar{x})/\lambda). \tag{4.2}$$

Under this specialization, the induced reasoning flow reduces to the reward-proportional flow underlying standard GFlowNet formulations.

### 4.2 Divergent Reasoning with Rules

In this section, based on our understanding of how our algorithm works via the reasoning DAG, we show that it enjoys several key properties for training reasoning LLMs. First, we show that our algorithm is mode-seeking, and recent studies have discussed the importance of mode-seeking behavior for sharpening LLMs toward top reasoning trajectories and improving their quality (Ji et al., 2024; Tajwar et al., 2024).

**Proposition 4.4** (Mode-Seeking)**.** *Over the reasoning DAG, recall that $d^*_\lambda$ denotes the target terminal distribution, and $d_\theta$ denotes the terminal distribution induced by the fine-tuned LLM. The expected gradient of our objective matches that of the reverse KL divergence:*

$$\nabla_\theta D_{\text{KL}}(d_\theta \,\|\, d^*_\lambda) = \frac{1}{2\lambda^2}\, \mathbb{E}_{x \sim d_\theta}\Big[\nabla_\theta \hat{\mathcal{L}}_t(\pi_\theta, V)\Big].$$

Since the reverse KL divergence is associated with mode-seeking behavior, Proposition 4.4 shows that our model inherits this mode-seeking property. Beyond its mode-seeking behavior, our algorithm also mitigates mode collapse, a common issue in RLVR methods that can cause rare but valid reasoning paths to be overlooked. Recent work shows that existing RLVR methods have limited ability to elicit genuinely novel reasoning due to such collapse (Yue et al., 2025a). Our algorithm addresses this issue through its intrinsic alignment with terminal rewards: it assigns probability mass to diverse high-reward reasoning paths rather than collapsing onto a small set of frequently sampled trajectories, as established in Theorem 4.3. We further formalize this diversity-preserving behavior.

**Proposition 4.5** (Diversity-Preserving). *Consider a multimodal terminal space where $\mathcal{X}_x$ is decomposed into $K$ disjoint modes $\mathcal{M}_1, \ldots, \mathcal{M}_K$. Recall $w_\lambda(x)$ is the unnormalized target weight over the reasoning DAG, and define score $U(\lambda, x) := \lambda \log w_\lambda(x)$. Then the target terminal distribution $d^*_\lambda$ assigns nontrivial probability mass to every mode. Specifically, for each $i = 1, \ldots, K$,*

$$\sum_{x \in \mathcal{M}_i} d^*_\lambda(x) \;\geq\; \frac{\exp\big(\frac{1}{\lambda} \max_{x \in \mathcal{M}_i} U(\lambda, x)\big)}{\sum_{j=1}^{K} |\mathcal{M}_j| \exp\big(\frac{1}{\lambda} \max_{x \in \mathcal{M}_j} U(\lambda, x)\big)}.$$

Additionally, we show that our algorithm enjoys a useful property, which we call *conditional diversity*. That is, the diversity induced by our algorithm is conditional: it is diversity within the set of paths that remain within the trust region and align with the reference policy, rather than unrestricted diversity over all possible reasoning paths. We formalize the *conditional diversity* as follows:

**Proposition 4.6** (Conditional Diversity). *Let $d^*_\lambda$ be the target terminal distribution in Theorem 4.3, guided by $\pi_{\text{ref}}$ and $\pi_{t-1}$. Suppose $m$ terminals $x_1, \ldots, x_m \in \mathcal{X}_x$ have near-optimal target weights, $w_\lambda(x_j) \geq \alpha \max_{x \in \mathcal{X}_x} w_\lambda(x)$ for $j = 1, \ldots, m$ and some relative quality threshold $\alpha \in (0, 1]$. Thus, these terminals receive mass at least $m\alpha\big(|\mathcal{X}_x| + m\alpha - m\big)^{-1}$.*

To explain why our algorithm has this property, we study the following adaptive notion.

**Proposition 4.7** (Temperature-Controlled Adaptive Mixture). *Over the reasoning DAG, the target terminal distribution is the softmax distribution evaluated at $\bar{x}$, i.e.,*

$$d^*_\lambda(\bar{x}) = \left[\text{softmax}_{x' \in \mathcal{X}_x} \left(U(\lambda, x')/\lambda\right)\right]_{\bar{x}}.$$

*Thus, $\lambda$ acts as a* temperature *controlling the concentration of $d^*_\lambda$: as $\lambda$ goes to 0, $d^*_\lambda$ concentrates on the set of score maximizers; as $\lambda$ increases, $d^*_\lambda$ becomes progressively more uniform over $\mathcal{X}_x$, assigning non-trivial probability to more diverse terminal solutions.*

Proposition 4.7 provides a direct interpretation of conditional diversity. The temperature $\lambda$ not only makes the target distribution more or less diverse, but also controls this diversity after terminal paths have been shaped by $\pi_{\text{ref}}$ and $\pi_{t-1}$. In summary, the characteristics of our objective can be summarized as "divergent reasoning with rules." Divergent reasoning arises naturally from alignment with terminal rewards in the reasoning DAG, while the rules are imposed by prior flows that constrain diverse reasoning paths and keep them aligned with the policy priors.

## 5 Regret Guarantee

Section 4 characterizes the distributional role of the two policy priors; we now show that the same structure also has a learning-theoretic consequence. In this section, we provide the theoretical regret analysis for our algorithm, which is the gold standard in reinforcement learning. First, we show that there exists a class of LLM policies fine-tuned without prior policy constraint via vanilla flow–based generative training, i.e., $\pi^*_\lambda$ in equation 4.2, that can not guarantee performance, which demonstrates the necessity of the policy constraints in our algorithm.

**Theorem 5.1.** *Suppose there exist a prompt $x^\star$ with $\rho(x^\star) > 0$, a reward-maximizing trajectory $y^\star = (a^\star_0, \ldots, a^\star_{H-1}) \in \arg\max_y R(x^\star, y)$, and a step $i$ such that $\pi_{\text{ref}}(a^\star_i \mid s^\star_i) = 0$, where $s^\star_i = (x^\star, a^\star_{<i})$. If every prompt $x$ admits at least one reward-maximizing trajectory, then there exists a policy $\pi^\star \in \arg\max_\pi J^0(\pi)$ such that $J(\pi^\star) = -\infty$.*

Theorem 5.1 indicates the necessity of LLM policy constraints, which are introduced by our reward augmentation: they affect not only the optimization stability but also the performance guarantee of the reasoning-incentivized LLM. Before presenting our regret guarantee, we first introduce some auxiliary LLM policy classes induced by the two prior policies and our working policy. For any radius $\varepsilon \geq 0, \delta \geq 0$, we have two token-wise KL divergence induced, i.e., $\text{KL}_{\text{token}}(\pi \,\|\, \mu) := \mathbb{E}_{x \sim \rho,\, y \sim \pi(\cdot|x)}\Big[\sum_{i=0}^{H-1} \log \frac{\pi(a_i|s_i)}{\mu(a_i|s_i)}\Big]$, policy classes:

$$\mathcal{B}_{\text{ref}}(\varepsilon) := \{\pi \in \Pi : \text{KL}_{\text{token}}(\pi \,\|\, \pi_{\text{ref}}) \leq \varepsilon\}, \quad \mathcal{B}_{t-1}(\delta) := \{\pi \in \Pi : \text{KL}_{\text{token}}(\pi \,\|\, \pi_{t-1}) \leq \delta\}.$$

Accordingly, our working LLM policy class at each iteration is $\Pi_t := \Pi \cap \mathcal{B}_{\text{ref}}(\varepsilon_1) \cap \mathcal{B}_{t-1}(\delta_t)$ for $\varepsilon_1 \geq 0$. To establish the regret guarantee for our algorithm, we discuss the coverage condition, which

is related to the drift of the LLMs during training and thus influences the regret guarantee (Song et al., 2024; Huang et al., 2024). In the standard settings, the so-called global coverage condition is required to control the drift of LLMs, i.e., $\sup_{\pi\in\Pi}\sup_{s,a}\frac{\pi(a|s)}{\pi_{\mathrm{ref}}(a|s)} < C$, for some positive constant $C$. By contrast, we show that our algorithm can work under a much weaker condition compared to the global coverage condition, due to the prior policy constraints via the reward augmentation.

**Assumption 1** (Local policy regularity)**.** *For any radius $u \geq 0$, there exist finite coefficients $C_{\mathrm{ref}}(u) \geq 1$ and $C_{t-1}(u) \geq 1$ such that*

$$\sup_{\pi\in\mathcal{B}_{\mathrm{ref}}(u)}\sup_{s,a}\left|\log\frac{\pi(a\mid s)}{\pi_{\mathrm{ref}}(a\mid s)}\right| \leq \log C_{\mathrm{ref}}(u), \quad \sup_{\pi\in\mathcal{B}_{t-1}(u)}\sup_{s,a}\left|\log\frac{\pi(a\mid s)}{\pi_{t-1}(a\mid s)}\right| \leq \log C_{t-1}(u).$$

*In addition, the locally constrained class $\Pi_t$ satisfies $\sup_{\pi\in\Pi_t}\sup_{x,y}\frac{\pi(y|x)}{\pi_{t-1}(y|x)} \leq C_{t-1}(\delta_t)$.*

Assumption 1 is imposed only on the locally constrained class $\Pi_t$, rather than on $\Pi$ globally. It ensures bounded trajectory residuals, while the trajectory-level coverage condition directly controls the distribution shift from the data-collection policy. More importantly, this condition implies that for the algorithm that can effectively control the drift via KL divergence, it is possible to obtain a strong regret guarantee. Intuitively, the two policy-prior constraints in our reward augmentation can provide such control, motivating the following theorem.

**Theorem 5.2.** *Suppose Assumption 1 holds, $\Pi_t$ and $\mathcal{V}$ are finite, and $|V(x)| \leq B_V$ for every $V \in \mathcal{V}$. Let $(\widehat{\pi}_t, \widehat{V}_t) \in \arg\min_{\pi\in\Pi_t, V\in\mathcal{V}} \widehat{\mathcal{L}}_t(\pi, V)$, let $\mathcal{L}_t(\pi, V)$ denote the population counterpart of $\widehat{\mathcal{L}}_t(\pi, V)$, and let $\pi^\dagger \in \arg\max_{\pi\in\Pi} J(\pi)$ satisfy $\pi^\dagger \in \cap_{t=1}^T \Pi_t$. Define $\alpha_t := \inf_{V\in\mathcal{V}} \mathcal{L}_t(\pi^\dagger, V)$ and $B_{\Delta_t} := B_V + 1 + \beta H \log C_{\mathrm{ref}}(\varepsilon_1) + \zeta H \log C_{t-1}(\delta_t)$. Suppose for $V_t^\dagger \in \arg\min_{V\in\mathcal{V}} \mathcal{L}_t(\pi^\dagger, V)$, $\mathbb{E}_\rho[\widehat{V}_t(x)] \geq \mathbb{E}_\rho[V_t^\dagger(x)] - \eta_t$. Then, with probability at least $1-\delta$,*

$$\sum_{t=1}^T \left[J(\pi^\dagger) - J(\widehat{\pi}_t)\right] \lesssim \sum_{t=1}^T \left[\eta_t + \sqrt{C_{t-1}(\delta_t)}\left(\sqrt{\alpha_t} + B_{\Delta_t}\sqrt[4]{\frac{\log(T|\Pi_t||\mathcal{V}|/\delta)}{n}}\right) + \zeta\delta_t\right].$$

Theorem 5.2 shows that our algorithm achieves a sample-complexity guarantee scaling with the localized trajectory-level coverage coefficients of both the reference policy and the last-iteration policy. This mainly follows from our reward-augmentation design, which captures the mutual effects of these two policy constraints. As a result, our algorithm is robust to LLM policy drift, and the output fine-tuned policy can compete with the optimal policy $\pi^\dagger$ when it is sufficiently covered by local KL neighborhoods.

Lastly, we show that the necessity of LLM policy constraint over $\pi_{t-1}$ is not only to stabilize the training but also to improve the regret guarantee. To facilitate our discussion, we define $\Pi_{\varepsilon_2} := \Pi \cap \mathcal{B}_{\mathrm{ref}}(\varepsilon_2)$ for $\varepsilon_2 \geq 0$, which is the policy class without effect term from $\pi_{t-1}$, and the corresponding coefficient $\widetilde{C}_t := \sup_{\pi\in\Pi_{\varepsilon_2}}\sup_{x,y}\frac{\pi(y|x)}{\pi_{t-1}(y|x)}$. For the comparison below, we further assume that $\pi^\dagger \in \Pi_{\varepsilon_2}$ and that $\Pi_{\varepsilon_2}$ is finite..

**Theorem 5.3.** *If $n$ is moderately large, the value-selection errors are comparable, and $\zeta\delta_t$ is lower order, then the regret bound in Theorem 5.2 is tighter than the one yielded by the training objective without the LLM policy constraint on $\pi_{t-1}$ whenever $\widetilde{C}_t \gtrsim \varepsilon_{\mathrm{ref},t}^{-1}\inf_{V\in\mathcal{V}}\mathcal{L}_t(\pi^\dagger, V)C_{t-1}(\delta_t)$ where $\varepsilon_{\mathrm{ref},t} := \inf_{V\in\mathcal{V}} \mathbb{E}_{\substack{x\sim\rho \\ y\sim\pi_{t-1}(\cdot|x)}}\left[V(x) - R(x,y) + \beta\sum_{i=0}^{H-1}\log\frac{\pi^\dagger(a_i|s_i)}{\pi_{\mathrm{ref}}(a_i|s_i)}\right]^2$.*

Theorem 5.3 identifies a regime in which the policy constraint over $\pi_{t-1}$ yields a strict improvement in the regret bound. Intuitively, when the possible drift toward the reference LLM policy is not too large compared with the policy constraint effect around $\pi_{t-1}$, the LLM policy constraint over $\pi_{t-1}$ is not only for training stabilization but also to improve the model performance.

## 6 Experimental Results

**Training and Evaluation Setup.** We conduct experiments to showcase the performance of the proposed algorithm on mathematical reasoning tasks, with comparisons against the popular baselines PPO (Schulman et al., 2017), GRPO (Guo et al., 2025), and the value-based baseline OREO

(Wang et al., 2025). We use the training set released with DAPO (Yu et al., 2025) (Apache License 2.0). We conduct training on two model scales, using the math-specialized model DeepSeek-Distill-1.5B (Guo et al., 2025) and the general-purpose model Qwen3-4B (Yang et al., 2025). For each model scale, the policy model $\pi_\theta$ and the critic model $V_\phi$ are initialized from the same pretrained checkpoint. The reference model $\pi_{\text{ref}}$ is set to the corresponding fixed pretrained model. We evaluate all methods on six challenging mathematical reasoning benchmarks: AIME24, AIME25, AMC23, MATH500 (Hendrycks et al., 2021), Minerva Math (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024). For more details on training configurations, please refer to the Appendix D.

Table 1: Evaluation on mathematical reasoning benchmarks. For *broad math reasoning* benchmarks (MATH500, Minerva Math, and OlympiadBench), we generate a single response per problem, denoted as Avg@1. For harder *competition-style* benchmarks (AIME24, AIME25, and AMC23), we generate 32 responses per problem and average the resulting accuracies, denoted as Avg@32.

| **Method** | **MATH500** Avg@1 | **Minerva Math** Avg@1 | **OlympiadBench** Avg@1 | **AIME24** Avg@32 | **AIME25** Avg@32 | **AMC23** Avg@32 | **Avg** |
|---|---|---|---|---|---|---|---|
| **DeepSeek-Distill-1.5B** | 63.0 | 26.5 | 27.7 | 8.7 | 6.8 | 38.1 | 28.4 |
| PPO ($n = 1$) | 79.0 | 30.2 | <u>46.4</u> | <u>17.0</u> | **16.6** | 63.1 | 42.0 |
| GRPO ($n = 4$) | **82.4** | **35.3** | **50.4** | 16.6 | 15.6 | <u>63.3</u> | **43.9** |
| OREO ($n = 1$) | 77.6 | <u>31.6</u> | 43.0 | 15.7 | 15.8 | 56.7 | 40.1 |
| Ours ($n = 1$) | <u>80.2</u> | 31.3 | 45.3 | **18.4** | <u>16.2</u> | **65.3** | <u>42.8</u> |
| **Qwen3-4B** | 72.4 | 35.3 | 37.6 | 17.1 | 16.4 | 53.3 | 38.7 |
| PPO ($n = 1$) | 80.4 | 40.4 | 45.8 | 16.5 | 18.6 | 61.1 | 43.8 |
| GRPO ($n = 4$) | <u>87.0</u> | <u>43.8</u> | <u>54.1</u> | **31.7** | <u>23.1</u> | <u>67.0</u> | <u>51.1</u> |
| OREO ($n = 1$) | 76.6 | 40.1 | 39.9 | 18.1 | 18.9 | 56.6 | 41.7 |
| Ours ($n = 1$) | **90.4** | **44.9** | **55.4** | <u>30.5</u> | **24.2** | **77.6** | **53.8** |

**Main Results.** Table 1 shows that our method outperforms the corresponding base models across all six benchmarks at both scales. Using a single rollout response per prompt during training ($n = 1$), it achieves the highest average accuracy among single-response methods, including PPO and OREO, on both models. On DeepSeek-Distill-1.5B, our method also outperforms GRPO ($n = 4$) on the competition-style benchmarks on average. On Qwen3-4B, it leads on five of six benchmarks and achieves the best overall average accuracy (53.8%), surpassing GRPO by 2.7 percentage points while using one-quarter as many rollout responses per prompt.

Table 2: Ablation study on the number of rollout responses per prompt $n$.

| **Method** | **MATH500** Avg@1 | **Minerva Math** Avg@1 | **OlympiadBench** Avg@1 | **AIME24** Avg@32 | **AIME25** Avg@32 | **AMC23** Avg@32 | **Avg** |
|---|---|---|---|---|---|---|---|
| Ours ($n = 1$) | 80.2 | 31.3 | 45.3 | 18.4 | 16.2 | **65.3** | 42.8 |
| Ours ($n = 4$) | **83.40** | **33.1** | **48.3** | **18.5** | **19.9** | 64.1 | 44.6 |

**Effect of Trust-Region.** Figure 1 shows that removing the KL regularization from the last iterate leads to severe instability and repeated reward collapse, whereas the full method remains stable. This result shows the effect of the trust-region term during LLM reasoning training.

**Rollout Ablation.** As shown in Table 2, increasing the number of rollout responses from $n = 1$ to $n = 4$ improves the average accuracy from 42.8 to 44.6 using DeepSeek-Distill-1.5B, indicating that our method benefits from additional rollout responses while remaining effective with a single rollout.

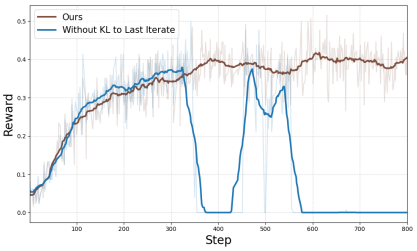


Figure 1: Ablation study on the KL regularization to the last iterate on DeepSeek-Distill-1.5B.

## 7 Conclusion

We develop a smoothed consistency framework for RLVR with reward augmentation. To explain its effectiveness for LLM reasoning, we develop a reasoning-flow characterization over a reasoning-directed acyclic graph (DAG). We validate our algorithm comprehensively through both theoretical analysis and empirical evaluation.

## AI USE STATEMENT

In this work, we used generative AI tools for [assisting in polishing writing, assisting in the writing of proofs]. We have not used generative AI tools for [generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, proposing or refining hypotheses, designing or providing feedback on research methodology or experiments, interpreting results, implementing methods, assisting with translation, cleaning and reformatting datasets, supporting qualitative and thematic data analysis, supporting qualitative and thematic data analysis, formulate questions for surveys or interviews, creating or modifying scientific figures or images, suggesting experimental parameters, creating artifacts, summarizing or analyzing existing literature, formatting references] that are not applicable to this work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

## REFERENCES

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.

Leemon Baird. Residual algorithms: Reinforcement learning with function approximation. In *Machine learning proceedings 1995*, pp. 30–37. Elsevier, 1995.

Emmanuel Bengio, Moksh Jain, Maksym Korablyov, Doina Precup, and Yoshua Bengio. Flow network based generative models for non-iterative diverse candidate generation. *Advances in neural information processing systems*, 34:27381–27394, 2021.

Yoshua Bengio, Salem Lahlou, Tristan Deleu, Edward J Hu, Mo Tiwari, and Emmanuel Bengio. Gflownet foundations. *Journal of Machine Learning Research*, 24(210):1–55, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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.

Bo Dai, Albert Shaw, Lihong Li, Lin Xiao, Niao He, Zhen Liu, Jianshu Chen, and Le Song. SBEED: Convergent reinforcement learning with nonlinear function approximation. In Jennifer Dy and Andreas Krause (eds.), *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pp. 1125–1134. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/dai18c.html.

Tristan Deleu, Padideh Nouri, Yoshua Bengio, and Doina Precup. Relative trajectory balance is equivalent to trust-pcl. *arXiv preprint arXiv:2509.01632*, 2025.

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, 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*, pp. 1861–1870. Pmlr, 2018.

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)*, pp. 3828–3850, 2024.

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. *arXiv preprint arXiv:2103.03874*, 2021.

Jean-Baptiste Hiriart-Urruty and Claude Lemaréchal. *Fundamentals of Convex Analysis*. Springer Science & Business Media, 2012.

Edward J Hu, Moksh Jain, Eric Elmoznino, Younesse Kaddar, Guillaume Lajoie, Yoshua Bengio, and Nikolay Malkin. Amortizing intractable inference in large language models. *arXiv preprint arXiv:2310.04363*, 2023.

Audrey Huang, Wenhao Zhan, Tengyang Xie, Jason D Lee, Wen Sun, Akshay Krishnamurthy, and Dylan J Foster. Correcting the mythos of kl-regularization: Direct alignment without overoptimization via chi-squared preference optimization. *arXiv preprint arXiv:2407.13399*, 2024.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Haozhe Ji, Cheng Lu, Yilin Niu, Pei Ke, Hongning Wang, Jun Zhu, Jie Tang, and Minlie Huang. Towards efficient exact optimization of language model alignment. *arXiv preprint arXiv:2402.00856*, 2024.

Vijay Konda and John Tsitsiklis. Actor-critic algorithms. *Advances in neural information processing systems*, 12, 1999.

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.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

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.

Kanika Madan, Jarrid Rector-Brooks, Maksym Korablyov, Emmanuel Bengio, Moksh Jain, Andrei Cristian Nica, Tom Bosc, Yoshua Bengio, and Nikolay Malkin. Learning gflownets from partial episodes for improved convergence and stability. In *International Conference on Machine Learning*, pp. 23467–23483. PMLR, 2023.

Nikolay Malkin, Moksh Jain, Emmanuel Bengio, Chen Sun, and Yoshua Bengio. Trajectory balance: Improved credit assignment in gflownets. *Advances in Neural Information Processing Systems*, 35:5955–5967, 2022a.

Nikolay Malkin, Salem Lahlou, Tristan Deleu, Xu Ji, Edward Hu, Katie Everett, Dinghuai Zhang, and Yoshua Bengio. Gflownets and variational inference. *arXiv preprint arXiv:2210.00580*, 2022b.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533, 2015.

Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Bridging the gap between value and policy based reinforcement learning. *Advances in neural information processing systems*, 30, 2017a.

Ofir Nachum, Mohammad Norouzi, Kelvin Xu, and Dale Schuurmans. Trust-pcl: An off-policy trust region method for continuous control. *arXiv preprint arXiv:1707.01891*, 2017b.

Yu Nesterov. Smooth minimization of non-smooth functions. *Mathematical programming*, 103(1): 127–152, 2005.

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.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 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.

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. *arXiv preprint arXiv: 2409.19256*, 2024.

Yuda Song, Gokul Swamy, Aarti Singh, J Bagnell, and Wen Sun. The importance of online data: Understanding preference fine-tuning via coverage. *Advances in Neural Information Processing Systems*, 37:12243–12270, 2024.

Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3 (1):9–44, 1988.

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

Fahim Tajwar, Anikait Singh, Archit Sharma, Rafael Rafailov, Jeff Schneider, Tengyang Xie, Stefano Ermon, Chelsea Finn, and Aviral Kumar. Preference fine-tuning of llms should leverage suboptimal, on-policy data. *arXiv preprint arXiv:2404.14367*, 2024.

Gerald Tesauro et al. Temporal difference learning and td-gammon. *Communications of the ACM*, 38(3):58–68, 1995.

Daniil Tiapkin, Nikita Morozov, Alexey Naumov, and Dmitry P Vetrov. Generative flow networks as entropy-regularized rl. In *International Conference on Artificial Intelligence and Statistics*, pp. 4213–4221. PMLR, 2024.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Siddarth Venkatraman, Moksh Jain, Luca Scimeca, Minsu Kim, Marcin Sendera, Mohsin Hasan, Luke Rowe, Sarthak Mittal, Pablo Lemos, Emmanuel Bengio, et al. Amortizing intractable inference in diffusion models for vision, language, and control. *Advances in neural information processing systems*, 37:76080–76114, 2024.

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

Christopher JCH Watkins and Peter Dayan. Q-learning. *Machine learning*, 8(3):279–292, 1992.

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.

Fangxu Yu. Flow of reasoning: Efficient training of llm policy with divergent thinking. *arXiv (Cornell University)*, 2024.

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

Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. What's behind ppo's collapse in long-cot? value optimization holds the secret. *arXiv preprint arXiv:2503.01491*, 2025a.

Yurun Yuan, Fan Chen, Zeyu Jia, Alexander Rakhlin, and Tengyang Xie. Trajectory bellman residual minimization: A simple value-based method for llm reasoning. *arXiv preprint arXiv:2505.15311*, 2025b.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025a.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025b.

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

Xuekai Zhu, Daixuan Cheng, Dinghuai Zhang, Hengli Li, Kaiyan Zhang, Che Jiang, Youbang Sun, Ermo Hua, Yuxin Zuo, Xingtai Lv, et al. Flowrl: Matching reward distributions for llm reasoning. *arXiv preprint arXiv:2509.15207*, 2025.

## A Technical Proofs for Section 3

***Proof of Theorem 3.1.*** Fix an iteration $t$. For a prompt $x$, let

$$y = (a_0, \dots, a_{H-1}) \sim \pi_t^\star(\cdot \mid x),$$
$$s_0 = (x, \emptyset), \quad s_{i+1} = (s_i, a_i), \quad i = 0, \dots, H-1,$$

be the completion and its induced reasoning states. After the last generated token, the process enters the auxiliary absorbing state, and we set $s_H = \perp$. To express the terminal reward in a stepwise form, define

$$r(s_h, a_h) := \begin{cases} 0, & h = 0, \dots, H-2, \\ R(x, y), & h = H-1, \end{cases} \qquad \text{so that} \qquad \sum_{h=0}^{H-1} r(s_h, a_h) = R(x, y).$$

Now define the optimal regularized value function $V_t^\star(s)$ from an intermediate state $s$ and let $Q_t^\star(s, a)$ denote the corresponding optimal action-value function. Since the transition is deterministic under autoregressive and $\pi_t^\star$ is optimal, we have

$$\begin{aligned} V_t^\star(s) &:= \sup_{\pi \in \Pi} \mathbb{E}\left[ \sum_{j=i}^{H-1} \left( r_j - \beta \log \frac{\pi(a_j \mid s_j)}{\pi_{\text{ref}}(a_j \mid s_j)} - \zeta \log \frac{\pi(a_j \mid s_j)}{\pi_{t-1}(a_j \mid s_j)} \right) \middle| \, s_i = s \right] \\ &= \max_{\pi \in \Pi} \sum_a \pi(a|s) \left[ Q_t^\star(s, a) - \beta \log \frac{\pi(a|s)}{\pi_{\text{ref}}(a \mid s)} - \zeta \log \frac{\pi(a|s)}{\pi_{t-1}(a \mid s)} \right], \end{aligned}$$

where $Q_t^\star(s_i, a_i) = r(s_i, a_i) + V_t^\star(s_{i+1}), i = 0, \dots, H-1$. Now we consider the Lagrangian

$$\mathcal{L}(\pi, \lambda) = \sum_a \pi(a|s) \left[ Q_t^\star(s, a) - \beta \log \frac{\pi(a|s)}{\pi_{\text{ref}}(a \mid s)} - \zeta \log \frac{\pi(a|s)}{\pi_{t-1}(a \mid s)} \right] + \lambda \left( \sum_a \pi(a|s) - 1 \right).$$

For any action $a$ with $\pi_t^\star(a|s) > 0$, the first-order optimality condition yields

$$\begin{aligned}
0 &= Q_t^\star(s,a) - \beta\left(\log\frac{\pi_t^\star(a|s)}{\pi_{\mathrm{ref}}(a \mid s)} + 1\right) - \zeta\left(\log\frac{\pi_t^\star(a|s)}{\pi_{t-1}(a \mid s)} + 1\right) + \lambda \\
&= Q_t^\star(s,a) - \beta\log\frac{\pi_t^\star(a \mid s)}{\pi_{\mathrm{ref}}(a \mid s)} - \zeta\log\frac{\pi_t^\star(a \mid s)}{\pi_{t-1}(a \mid s)} - (\beta + \zeta - \lambda) \\
\beta + \zeta - \lambda &= Q_t^\star(s,a) - \beta\log\frac{\pi_t^\star(a \mid s)}{\pi_{\mathrm{ref}}(a \mid s)} - \zeta\log\frac{\pi_t^\star(a \mid s)}{\pi_{t-1}(a \mid s)} \\
&= \sum_a \pi_t^\star(a \mid s)\left[Q_t^\star(s,a) - \beta\log\frac{\pi_t^\star(a \mid s)}{\pi_{\mathrm{ref}}(a \mid s)} - \zeta\log\frac{\pi_t^\star(a \mid s)}{\pi_{t-1}(a \mid s)}\right] \\
&= V_t^\star(s),
\end{aligned}$$

where the last step is because the left-hand side is constant over all actions in the support of $\pi_t^\star(\cdot \mid s)$. Multiplying the above identity by $\pi_t^\star(a \mid s)$ and summing over $a$, we obtain the last step.

Since $y \sim \pi_t^\star(\cdot \mid x)$, each sampled action $a_i$ lies in the support of $(\pi_t^\star(\cdot \mid s_i)$ almost surely. Hence, for every $i = 0, \ldots, H-1$,

$$\begin{aligned}
V_t^\star(s_i) &= Q_t^\star(s_i,a_i) - \beta\log\frac{\pi_t^\star(a_i \mid s_i)}{\pi_{\mathrm{ref}}(a_i \mid s_i)} - \zeta\log\frac{\pi_t^\star(a_i \mid s_i)}{\pi_{t-1}(a_i \mid s_i)} \\
&= r(s_i,a_i) + V_t^\star(s_{i+1}) - \beta\log\frac{\pi_t^\star(a_i \mid s_i)}{\pi_{\mathrm{ref}}(a_i \mid s_i)} - \zeta\log\frac{\pi_t^\star(a_i \mid s_i)}{\pi_{t-1}(a_i \mid s_i)} \\
V_t^\star(s_i) - V_t^\star(s_{i+1}) &= r(s_i,a_i) - \beta\log\frac{\pi_t^\star(a_i \mid s_i)}{\pi_{\mathrm{ref}}(a_i \mid s_i)} - \zeta\log\frac{\pi_t^\star(a_i \mid s_i)}{\pi_{t-1}(a_i \mid s_i)}.
\end{aligned}$$

Summing over $i = 0, \ldots, H-1$, the value terms telescope:

$$\begin{aligned}
V_t^\star(s_0) - V_t^\star(s_\perp) &= \sum_{i=0}^{H-1} r(s_i,a_i) - \beta\sum_{i=0}^{H-1}\log\frac{\pi_t^\star(a_i \mid s_i)}{\pi_{\mathrm{ref}}(a_i \mid s_i)} - \zeta\sum_{i=0}^{H-1}\log\frac{\pi_t^\star(a_i \mid s_i)}{\pi_{t-1}(a_i \mid s_i)} \\
&= R(x,y) - \beta\sum_{i=0}^{H-1}\log\frac{\pi_t^\star(a_i \mid s_i)}{\pi_{\mathrm{ref}}(a_i \mid s_i)} - \zeta\sum_{i=0}^{H-1}\log\frac{\pi_t^\star(a_i \mid s_i)}{\pi_{t-1}(a_i \mid s_i)}.
\end{aligned}$$

This proves the stated multi-step consistency identity. Under the terminal condition $V_t^\star(s_\perp) = 0$, it further reduces to

$$\begin{aligned}
V_t^\star(x) &= R(x,y) - \beta\sum_{i=0}^{H-1}\log\frac{\pi_t^\star(a_i \mid s_i)}{\pi_{\mathrm{ref}}(a_i \mid s_i)} - \zeta\sum_{i=0}^{H-1}\log\frac{\pi_t^\star(a_i \mid s_i)}{\pi_{t-1}(a_i \mid s_i)} \\
&=: G_t(x,y;\pi_t^\star).
\end{aligned}$$

This completes the proof. □

## B TECHNICAL PROOFS FOR SECTION 4

***Proof of Theorem 4.3.*** For intermediate state $s$, define value function $V_\lambda^*(s)$ under latent stepwise reward function in Definition 4.1,

$$\begin{aligned}
V_\lambda^*(s) &= \sup_{\pi\in\Pi}\mathbb{E}_{\tau\sim\pi}\left[\sum_{i=0}^{T-1} r(s_i,s_{i+1}) - \lambda\log\pi(s_{i+1} \mid s_i)\right], \\
V_\lambda^*(\perp) &= 0, \qquad V_\lambda^*(\bar{x}) = \log R(x,\bar{x}) + \zeta\log\pi_{t-1}(\bar{x}),
\end{aligned}$$

where $\lambda = \zeta + \beta$, and terminal states $\bar{x} \in \mathcal{X}_x$. Define the *reasoning flow* $\mathcal{F}_\lambda$ in Definition 2 as,

$$\mathcal{F}_\lambda(s) := \exp\left(V_\lambda^*(s)/\lambda\right).$$

Thus, the reasoning flow at the terminal state and intermediate state is as follows,

$$\begin{aligned}
\text{For } \bar{x} \in \mathcal{X}_x,\ \mathcal{F}_\lambda(\bar{x}) &= \exp\left(\frac{1}{\lambda}\left(\log R(\bar{x}) + \zeta \log \pi_{t-1}(\bar{x})\right)\right) \\
&= \left(R(\bar{x})\, \pi_{t-1}(\bar{x})^{\zeta}\right)^{1/\lambda}, \\
\text{For } s \notin \mathcal{X}_x,\ \mathcal{F}_\lambda(s) &= \sum_{s' \in \mathcal{A}_s} \exp\left(\frac{1}{\lambda} r(s, s')\right) \mathcal{F}_\lambda(s') \\
&= \sum_{s' \in \mathcal{A}_s} \pi_{\text{ref}}(s' \mid s)^{\beta/\lambda}\, \mathcal{F}_\lambda(s'), \\
\mathcal{F}_\lambda(s \to s') &= \pi_{\text{ref}}(s' \mid s)^{\beta/\lambda}\, \mathcal{F}_\lambda(s').
\end{aligned}$$

Then the soft Bellman optimality equations imply the action-value function and the optimal reasoning flow policy,

$$\begin{aligned}
V^*_\lambda(s) &= \lambda \log \mathcal{F}_\lambda(s), \\
Q^*_\lambda(s, s') &= \lambda \log \mathcal{F}_\lambda(s \to s'), \\
\pi^*(s' \mid s) &= \exp\Big(\tfrac{1}{\lambda}\big(Q^*_\lambda(s, s') - V^*_\lambda(s)\big)\Big) \\
&= \frac{\mathcal{F}_\lambda(s \to s')}{\mathcal{F}_\lambda(s)}.
\end{aligned}$$

Consider an unique path $(s_0 \to s_1 \to \cdots \to s_T = \bar{x})$ with terminal $\bar{x} \in \mathcal{X}_x$, define

$$\begin{aligned}
d^*_\lambda(\bar{x}) &= \prod_{t=0}^{T-1} \pi^*(s_{t+1} \mid s_t) \\
&= \prod_{t=0}^{T-1} \frac{\mathcal{F}_\lambda(s_t \to s_{t+1})}{\mathcal{F}_\lambda(s_t)} \\
&= \frac{\mathcal{F}_\lambda(\bar{x})}{\mathcal{F}_\lambda(s_0)} \prod_{i=0}^{T-1} \pi_{\text{ref}}(s_{i+1} \mid s_i)^{\beta/\lambda} \\
&\propto\ \pi_{\text{ref}}(\bar{x})^{\beta/\lambda}\, \pi_{t-1}(\bar{x})^{\zeta/\lambda}\, R(\bar{x})^{1/\lambda}.
\end{aligned}$$

after telescoping cancels all intermediate $\mathcal{F}_\lambda(s_t)$. Immediately after unrolling $F_\lambda(s_0)$, we obtain normalization,

$$\begin{aligned}
Z &:= F_\lambda(s_0) \\
&= \exp\{V^*_\lambda(s_0)/\lambda\} \\
&= \sum_{\bar{x} \in \mathcal{X}_x} \pi_{\text{ref}}(\bar{x})^{\beta/\lambda}\, \pi_{t-1}(\bar{x})^{\zeta/\lambda}\, R(\bar{x})^{1/\lambda}.
\end{aligned}$$

We have shown the target terminal distribution in Theorem 4.3. Furthermore, (Deleu et al., 2025) provides some insights on the connection between GFlowNets and Reasoning DAG. □

***Proof of Proposition 4.4.*** We follow the specific proof of Proposition 1 in (Malkin et al., 2022b), which connects Trajectory Balance (TB) in GFlowNets to reverse KL. Define $c(x)$ for each terminal $x \in \mathcal{X}_x$,

$$\begin{aligned}
c(x) &:= \log \frac{d_\theta(x)}{w_\lambda(x)}, \\
\nabla_\theta c(x) &= \nabla_\theta \log d_\theta(x).
\end{aligned}$$

since weight $w_\lambda(x)$ does not depend on $\theta$. Given $d^*_\lambda(x) = w_\lambda(x)/Z_\lambda$ in Theorem 4.3, we are able to rewrite the reverse KL in terms of $c(x)$,

$$\begin{aligned}
D_{\text{KL}}(d_\theta \| d^*_\lambda) &= \mathbb{E}_{x \sim d_\theta}\left[\log \frac{d_\theta(x)}{d^*_\lambda(x)}\right] && \text{(B.1)} \\
&= \mathbb{E}_{x \sim d_\theta}[c(x)] + \log Z_\lambda, && \text{(B.2)}
\end{aligned}$$

Differentiate equation B.2 w.r.t. $\theta$:

$$\begin{aligned}\nabla_\theta D_{\mathrm{KL}}(d_\theta \| d_\lambda^*) &= \nabla_\theta \mathbb{E}_{x\sim d_\theta}[c(x)] \\ &= \mathbb{E}_{x\sim d_\theta}\left[\nabla_\theta \log d_\theta(x)\, c(x) + \nabla_\theta c(x)\right] \\ &= \mathbb{E}_{x\sim d_\theta}\left[\nabla_\theta \log d_\theta(x)(c(x) + \log Z)\right] + \mathbb{E}_{x\sim d_\theta}[\nabla_\theta c(x)] \\ &= \mathbb{E}_{x\sim d_\theta}\left[\nabla_\theta \log d_\theta(x)(c(x) + \log Z)\right] + \sum_{x\in\mathcal{X}_x} \nabla_\theta d_\theta(x) \\ &= \mathbb{E}_{x\sim d_\theta}\left[\nabla_\theta \log d_\theta(x)(c(x) + \log Z)\right] + \nabla_\theta 1 \\ &= \mathbb{E}_{x\sim d_\theta}\left[\nabla_\theta \log d_\theta(x)(c(x) + \log Z)\right].\end{aligned}$$

The second equality follows by using the score-function identity. The third equality follows from the fact that the score term remains unchanged after we add any scalar $Z > 0$ independent of $\theta$ and the equation in equation B.1.

Let $\Delta_\theta(x; Z) := c(x) + \log Z$, and note that

$$\begin{aligned}\Delta_\theta(x; Z) &= \log(Z d_\theta(x)/w_\lambda(x)), \\ \nabla_\theta \Delta_\theta(x; Z) &= \nabla_\theta \log d_\theta(x).\end{aligned}$$

Therefore,

$$\begin{aligned}\nabla_\theta \hat{\mathcal{L}}_t(\pi_\theta, V_\phi) &= \nabla_\theta \Delta_\theta(x; Z)^2 && \text{(B.3)} \\ &= 2\lambda^2 \Delta_\theta(x; Z)\, \nabla_\theta \Delta_\theta(x; Z) && \text{(B.4)} \\ &= 2\lambda^2 \Delta_\theta(x; Z)\, \nabla_\theta \log d_\theta(x). && \text{(B.5)}\end{aligned}$$

Taking expectation of equation B.5 under $d_\theta$, our loss aligns with the reverse KL divergence in terms of expected gradients. □

***Proof of Proposition 4.5.*** We assume terminal set $\mathcal{X}_x$ can be decomposed into $K$ disjoint modes, and we denote them by $\{\mathcal{M}_1, \ldots, \mathcal{M}_K\}$. Let

$$\begin{aligned}U(\lambda, x) :=& \lambda \log w_\lambda(x) \\ =& \beta \log \pi_{\text{ref}}(x) + \zeta \log \pi_{t-1}(x) + \log R(x).\end{aligned}$$

Then, for every mode $\mathcal{M}_i, i \in \{1, \ldots, K\}$, we have

$$\begin{aligned}\mathbb{P}_\lambda(\mathcal{M}_i) &= \sum_{x\in\mathcal{M}_i} d_\lambda^*(x) \\ &= \frac{\sum_{x\in\mathcal{M}_i} w_\lambda(x)}{\sum_{x\in\mathcal{X}} w_\lambda(x)} \\ &= \frac{\sum_{x\in\mathcal{M}_i} \exp(U(\lambda, x)/\lambda)}{\sum_{x\in\mathcal{X}} \exp(U(\lambda, x)/\lambda)} \\ &\geq \frac{\exp(\max_{x\in\mathcal{M}_i} U(\lambda, x)/\lambda)}{\sum_{j=1}^K |\mathcal{M}_j| \exp(\max_{x\in\mathcal{M}_j} U(\lambda, x)/\lambda)},\end{aligned}$$

where $U_i = \max_{x\in\mathcal{M}_i} u(x)$. □

***Proof of Proposition 4.6.*** For any subset $A \subseteq \mathcal{X}_x$,

$$\begin{aligned}\mathbb{P}_\lambda(A) &= \sum_{x\in A} d_\lambda^*(x) && \text{(B.6)} \\ &= \frac{\sum_{x\in A} w_\lambda(x)}{\sum_{x'\in\mathcal{X}_x} w_\lambda(x')}. && \text{(B.7)}\end{aligned}$$

Equation equation B.7 follows immediately from Theorem 4.3. Let $w_{\max} := \max_{x\in\mathcal{X}_x} w_\lambda(x)$. Then, for some relative quality threshold $\alpha \in (0, 1]$,

$$\begin{aligned}\sum_{j=1}^m w_\lambda(x_j) &\geq m\alpha w_{\max}, \\ \sum_{x\in\mathcal{X}_x\setminus\{x_j\}} w_\lambda(x) &\leq (|\mathcal{X}_x| - m) w_{\max},\end{aligned}$$

which yields

$$\begin{aligned}\mathbb{P}_\lambda(\{x_1,\dots,x_m\}) &= \frac{\sum_{j=1}^m w_\lambda(x_j)}{\sum_{j=1}^m w_\lambda(x_j)+\sum_{x\notin\{x_j\}} w_\lambda(x)} \\ &\geq \frac{m\alpha w_{\max}}{m\alpha w_{\max}+(|\mathcal{X}_x|-m)w_{\max}} \\ &= \frac{m\alpha}{m\alpha+(|\mathcal{X}_x|-m)}.\end{aligned}$$

□

## C TECHNICAL PROOFS FOR SECTION 5

***Proof of Theorem 5.1.*** Construct a determinstic policy $\pi^\star$ such that, for each prompt $x$, places total mass on a reward-maximizing trajectory $y_x^\star$, and in particular outputs $y^\star$ when $x = x^\star$. Then:

$$\begin{aligned}J^0(\pi) &= \mathbb{E}_{x\sim\rho}\left[\mathbb{E}_{y\sim\pi(\cdot|x)}[R(x,y)]\right] \\ &\leq \mathbb{E}_{x\sim\rho}\left[\max_y R(x,y)\right], \\ J^0(\pi^\star) &= \mathbb{E}_{x\sim\rho}\left[\mathbb{E}_{y\sim\pi^\star(\cdot|x)}[R(x,y)]\right] \\ &= \mathbb{E}_{x\sim\rho}\left[R(x,y_x^\star)\right] \\ &= \mathbb{E}_{x\sim\rho}\left[\max_y R(x,y)\right],\end{aligned}$$

which implies $\pi^\star \in \arg\max_\pi J^0(\pi)$. On the other hand, since under $x^\star$ the policy $\pi^\star$ deterministically generates $y^\star$, we have $\pi^\star(a_i^\star \mid s_i^\star) = 1, \pi_{\mathrm{ref}}(a_i^\star \mid s_i^\star) = 0$, and therefore

$$\begin{aligned}J(\pi^\star) &= \mathbb{E}_{x\sim\rho,y\sim\pi^\star(\cdot|x)}[R(x,y)] - \beta\mathbb{E}_{x\sim\rho,y\sim\pi^\star(\cdot|x)}\left[\sum_{h=0}^{H-1}\log\frac{\pi^\star(a_h\mid s_h)}{\pi_{\mathrm{ref}}(a_h\mid s_h)}\right] \\ &\leq J^0(\pi^\star) - \beta\rho(x^\star)\log\frac{\pi^\star(a_i^\star\mid s_i^\star)}{\pi_{\mathrm{ref}}(a_i^\star\mid s_i^\star)} \\ &= J^0(\pi^\star) - \beta\rho(x^\star)\log\frac{1}{0} \\ &= -\infty.\end{aligned}$$

Hence, there exists a policy in the optimal policy class of $J^0$ whose value under the regularized objective $J$ is $-\infty$.

□

In the following, we provide the proof of the regret bound with the reference and the last-iteration policy regularization.

At iteration $t$, trajectories are collected from the previous iterate $\pi_{t-1}$. For any $\pi \in \Pi_t$ and $V \in \mathcal{V}$, define

$$
\begin{aligned}
G_t(x, y; \pi) &:= R(x, y) - \beta \sum_{i=0}^{H-1} \log \frac{\pi(a_i \mid s_i)}{\pi_{\mathrm{ref}}(a_i \mid s_i)} - \zeta \sum_{i=0}^{H-1} \log \frac{\pi(a_i \mid s_i)}{\pi_{t-1}(a_i \mid s_i)}, \\
\Delta_t(x, y; \pi, V) &:= V(x) - G_t(x, y; \pi), \\
\mathcal{L}_t(\pi, V) &:= \mathbb{E}_{x \sim \rho,\ y \sim \pi_{t-1}(\cdot|x)} \left[\Delta_t(x, y; \pi, V)^2\right], \\
\mathcal{L}_t^{\pi}(\pi, V) &:= \mathbb{E}_{x \sim \rho,\ y \sim \pi(\cdot|x)} \left[\Delta_t(x, y; \pi, V)^2\right], \\
\widehat{\mathcal{L}}_t(\pi, V) &:= \frac{1}{n} \sum_{k=1}^{n} \Delta_t(x_k, y_k; \pi, V)^2, \qquad x_k \sim \rho, \quad y_k \sim \pi_{t-1}(\cdot \mid x_k), \\
(\widehat{\pi}_t, \widehat{V}_t) &\in \arg \min_{\pi \in \Pi_t,\ V \in \mathcal{V}} \widehat{\mathcal{L}}_t(\pi, V), \\
V_t^{\dagger} &\in \arg \min_{V \in \mathcal{V}} \mathcal{L}_t(\pi^{\dagger}, V), \\
\alpha_t &:= \mathcal{L}_t(\pi^{\dagger}, V_t^{\dagger}) \\
&= \inf_{V \in \mathcal{V}} \mathcal{L}_t(\pi^{\dagger}, V).
\end{aligned}
$$

**Lemma C.1.** *For any $\pi \in \Pi_t$ and $V \in \mathcal{V}$,*

$$
|\Delta_t(x, y; \pi, V)| \le B_{\Delta_t}, \qquad B_{\Delta_t} := B_V + 1 + \beta H \log C_{\mathrm{ref}}(\varepsilon_1) + \zeta H \log C_{t-1}(\delta_t).
$$

*Consequently,* $0 \le \Delta_t(x, y; \pi, V)^2 \le B_{\Delta_t}^2$.

***Proof of Lemma C.1.*** For any $\pi \in \Pi_t = \Pi \cap \mathcal{B}_{\mathrm{ref}}(\varepsilon_1) \cap \mathcal{B}_{t-1}(\delta_t)$, the bounded log-ratio condition gives

$$
\left|\log \frac{\pi(a_i \mid s_i)}{\pi_{\mathrm{ref}}(a_i \mid s_i)}\right| \le \log C_{\mathrm{ref}}(\varepsilon_1), \qquad \left|\log \frac{\pi(a_i \mid s_i)}{\pi_{t-1}(a_i \mid s_i)}\right| \le \log C_{t-1}(\delta_t).
$$

Since $R(x, y) \in \{0, 1\}$, we have

$$
\begin{aligned}
|G_t(x, y; \pi)| &\le 1 + \beta \sum_{i=0}^{H-1} \left|\log \frac{\pi(a_i \mid s_i)}{\pi_{\mathrm{ref}}(a_i \mid s_i)}\right| + \zeta \sum_{i=0}^{H-1} \left|\log \frac{\pi(a_i \mid s_i)}{\pi_{t-1}(a_i \mid s_i)}\right| \\
&\le 1 + \beta H \log C_{\mathrm{ref}}(\varepsilon_1) + \zeta H \log C_{t-1}(\delta_t).
\end{aligned}
$$

Together with $|V(x)| \le B_V$, this implies

$$
\begin{aligned}
|\Delta_t(x, y; \pi, V)| &= |V(x) - G_t(x, y; \pi)| \\
&\le B_{\Delta_t}.
\end{aligned}
$$

The proof is completed. □

**Lemma C.2.** *With probability at least $1 - \delta$, for all $t = 1, \ldots, T$,*

$$
\sup_{\pi \in \Pi_t,\ V \in \mathcal{V}} \left|\widehat{\mathcal{L}}_t(\pi, V) - \mathcal{L}_t(\pi, V)\right| \le \epsilon_t,
$$

*where* $\epsilon_t := B_{\Delta_t}^2 \sqrt{\frac{\log \frac{2T|\Pi_t||\mathcal{V}|}{\delta}}{2n}}$.

***Proof of Lemma C.2.*** Let $\mathcal{F}_{t-1}$ denote the history before the data at iteration $t$ are collected. Conditioned on $\mathcal{F}_{t-1}$, the policy $\pi_{t-1}$ and the feasible class $\Pi_t$ are fixed, and $\{(x_k, y_k)\}_{k=1}^n$ are independently distributed according to $\rho(x)\pi_{t-1}(y \mid x)$. For any fixed $(\pi, V) \in \Pi_t \times \mathcal{V}$, Lemma C.1 and Hoeffding's inequality give

$$
\begin{aligned}
&0 \le \Delta_t(x, y; \pi, V)^2 \le B_{\Delta_t}^2, \\
&\mathbb{P}\left(\left|\widehat{\mathcal{L}}_t(\pi, V) - \mathcal{L}_t(\pi, V)\right| \ge \epsilon \,\middle|\, \mathcal{F}_{t-1}\right) \le 2 \exp\left(-\frac{2n\epsilon^2}{B_{\Delta_t}^4}\right).
\end{aligned}
$$

Taking a union bound over $(\pi, V) \in \Pi_t \times \mathcal{V}$ yields

$$\mathbb{P}\left(\sup_{\pi\in\Pi_t,\, V\in\mathcal{V}} \left|\widehat{\mathcal{L}}_t(\pi, V) - \mathcal{L}_t(\pi, V)\right| \geq \epsilon_t \,\middle|\, \mathcal{F}_{t-1}\right) \leq \frac{\delta}{T},$$

$$\epsilon_t := B_{\Delta_t}^2 \sqrt{\frac{\log \frac{2T|\Pi_t||\mathcal{V}|}{\delta}}{2n}}.$$

Taking expectations with respect to $\mathcal{F}_{t-1}$ and then applying a union bound over $t = 1, \ldots, T$ proves that the stated concentration event holds simultaneously for all iterations with probability at least $1 - \delta$. □

**Lemma C.3.** *On the event in Lemma C.2, the empirical minimizer satisfies*

$$\mathcal{L}_t(\widehat{\pi}_t, \widehat{V}_t) \leq \alpha_t + 2\epsilon_t.$$

***Proof of Lemma C.3.*** By applying lemma C.2 and the definition of $\alpha_t$, we have that

$$\begin{aligned}
\mathcal{L}_t(\widehat{\pi}_t, \widehat{V}_t) &\leq \widehat{\mathcal{L}}_t(\widehat{\pi}_t, \widehat{V}_t) + \epsilon_t \\
&\leq \widehat{\mathcal{L}}_t(\pi^\dagger, V_t^\dagger) + \epsilon_t \\
&\leq \mathcal{L}_t(\pi^\dagger, V_t^\dagger) + 2\epsilon_t \\
&= \alpha_t + 2\epsilon_t.
\end{aligned}$$

The proof is completed. □

**Lemma C.4.** *For any $\pi \in \Pi_t$ and $V \in \mathcal{V}$,*

$$|\mathbb{E}_{x\sim\rho}[V(x)] - J_t(\pi)| \leq \sqrt{C_{t-1}(\delta_t)}\sqrt{\mathcal{L}_t(\pi, V)}.$$

***Proof of Lemma C.4.*** By the definitions of $\Delta_t$ and $J_t$,

$$\mathbb{E}_{\substack{x\sim\rho \\ y\sim\pi(\cdot|x)}} [\Delta_t(x, y; \pi, V)] = \mathbb{E}_{x\sim\rho}[V(x)] - J_t(\pi).$$

Therefore, by Cauchy–Schwarz,

$$|\mathbb{E}_{x\sim\rho}[V(x)] - J_t(\pi)| \leq \sqrt{\mathcal{L}_t^\pi(\pi, V)}.$$

Using Assumption 1,

$$\begin{aligned}
\mathcal{L}_t^\pi(\pi, V) &= \mathbb{E}_{\substack{x\sim\rho \\ y\sim\pi_{t-1}(\cdot|x)}} \left[\frac{\pi(y \mid x)}{\pi_{t-1}(y \mid x)} \Delta_t(x, y; \pi, V)^2\right] \\
&\leq C_{t-1}(\delta_t)\mathcal{L}_t(\pi, V).
\end{aligned}$$

Combining the two inequalities proves the result. □

**Lemma C.5.** *Conditioned on the event in Lemma C.2,*

$$J_t(\pi^\dagger) - J_t(\widehat{\pi}_t) \leq \eta_t + \sqrt{C_{t-1}(\delta_t)}\left(\sqrt{\alpha_t} + \sqrt{\alpha_t + 2\epsilon_t}\right).$$

***Proof of Lemma C.5.*** Adding and subtracting the expected values of $V_t^\dagger$ and $\widehat{V}_t$, and using the value-selection condition, gives

$$\begin{aligned}
J_t(\pi^\dagger) - J_t(\widehat{\pi}_t) &\leq \left|J_t(\pi^\dagger) - \mathbb{E}_{x\sim\rho}[V_t^\dagger(x)]\right| + \eta_t + \left|\mathbb{E}_{x\sim\rho}[\widehat{V}_t(x)] - J_t(\widehat{\pi}_t)\right| \\
&\leq \eta_t + \sqrt{C_{t-1}(\delta_t)}\left(\sqrt{\mathcal{L}_t(\pi^\dagger, V_t^\dagger)} + \sqrt{\mathcal{L}_t(\widehat{\pi}_t, \widehat{V}_t)}\right) \\
&\leq \eta_t + \sqrt{C_{t-1}(\delta_t)}\left(\sqrt{\alpha_t} + \sqrt{\alpha_t + 2\epsilon_t}\right),
\end{aligned}$$

where the second inequality follows from Lemma C.4, and the last follows from Lemma C.3. □

***Proof of Theorem 5.2.*** By Lemma C.5, simultaneously for all $t = 1, \dots, T$,

$$J_t(\pi^\dagger) - J_t(\widehat{\pi}_t) \le \eta_t + \sqrt{C_{t-1}(\delta_t)}\left(\sqrt{\alpha_t} + \sqrt{\alpha_t + 2\epsilon_t}\right).$$

Since $J_t(\pi) = J(\pi) - \zeta \mathrm{KL}_{\mathrm{token}}(\pi \| \pi_{t-1})$,

$$\begin{aligned} J(\pi^\dagger) - J(\widehat{\pi}_t) &= J_t(\pi^\dagger) - J_t(\widehat{\pi}_t) + \zeta \mathrm{KL}_{\mathrm{token}}(\pi^\dagger \| \pi_{t-1}) - \zeta \mathrm{KL}_{\mathrm{token}}(\widehat{\pi}_t \| \pi_{t-1}) \\ &\le J_t(\pi^\dagger) - J_t(\widehat{\pi}_t) + \zeta \delta_t. \end{aligned}$$

Therefore,

$$\sum_{t=1}^{T} \left[J(\pi^\dagger) - J(\widehat{\pi}_t)\right] \le \sum_{t=1}^{T} \left[\eta_t + \sqrt{C_{t-1}(\delta_t)}\left(\sqrt{\alpha_t} + \sqrt{\alpha_t + 2\epsilon_t}\right) + \zeta \delta_t\right].$$

Using $\sqrt{\alpha_t + 2\epsilon_t} \le \sqrt{\alpha_t} + \sqrt{2\epsilon_t}$ gives

$$\sum_{t=1}^{T} \left[J(\pi^\dagger) - J(\widehat{\pi}_t)\right] \le \sum_{t=1}^{T} \left[\eta_t + \sqrt{C_{t-1}(\delta_t)} \left(2\sqrt{\alpha_t} + \sqrt{2}\, B_{\Delta_t} \sqrt[4]{\frac{\log \frac{2T|\Pi_t||\mathcal{V}|}{\delta}}{2n}}\right) + \zeta \delta_t\right].$$

□

In the following, we study the regret bound with only reference regularization. Let $\Pi_{\varepsilon_2} := \Pi \cap \mathcal{B}_{\mathrm{ref}}(\varepsilon_2)$. For any $\pi \in \Pi_{\varepsilon_2}$ and $V \in \mathcal{V}$, define

$$\begin{aligned} G_{\mathrm{ref}}(x, y; \pi) &:= R(x, y) - \beta \sum_{i=0}^{H-1} \log \frac{\pi(a_i \mid s_i)}{\pi_{\mathrm{ref}}(a_i \mid s_i)}, \\ \Delta_{\mathrm{ref}}(x, y; \pi, V) &:= V(x) - G_{\mathrm{ref}}(x, y; \pi), \\ \mathcal{L}^{t-1}_{\mathrm{ref},t}(\pi, V) &:= \mathbb{E}_{\substack{x \sim \rho \\ y \sim \pi_{t-1}(\cdot|x)}} \left[\Delta_{\mathrm{ref}}(x, y; \pi, V)^2\right], \\ \mathcal{L}^{\pi}_{\mathrm{ref}}(\pi, V) &:= \mathbb{E}_{\substack{x \sim \rho \\ y \sim \pi(\cdot|x)}} \left[\Delta_{\mathrm{ref}}(x, y; \pi, V)^2\right], \\ \widehat{\mathcal{L}}_{\mathrm{ref},t}(\pi, V) &:= \frac{1}{n} \sum_{k=1}^{n} \Delta_{\mathrm{ref}}(x_k, y_k; \pi, V)^2, \\ (\widehat{\pi}_{\mathrm{ref},t}, \widehat{V}_{\mathrm{ref},t}) &\in \arg \min_{\pi \in \Pi_{\varepsilon_2},\, V \in \mathcal{V}} \widehat{\mathcal{L}}_{\mathrm{ref},t}(\pi, V), \\ V^\dagger_{\mathrm{ref},t} &\in \arg \min_{V \in \mathcal{V}} \mathcal{L}^{t-1}_{\mathrm{ref},t}(\pi^\dagger, V), \\ \varepsilon_{\mathrm{ref},t} &:= \mathcal{L}^{t-1}_{\mathrm{ref},t}(\pi^\dagger, V^\dagger_{\mathrm{ref},t}) = \inf_{V \in \mathcal{V}} \mathcal{L}^{t-1}_{\mathrm{ref},t}(\pi^\dagger, V). \end{aligned}$$

The corresponding trajectory-level coverage coefficient is

$$\widetilde{C}_t := \sup_{\pi \in \Pi_{\varepsilon_2}} \sup_{x,y} \frac{\pi(y \mid x)}{\pi_{t-1}(y \mid x)},$$

**Lemma C.6.** *For any $\pi \in \Pi_{\varepsilon_2}$ and $V \in \mathcal{V}$,*

$$|\Delta_{\mathrm{ref}}(x, y; \pi, V)| \le B_{\Delta_{\mathrm{ref}}}, \qquad B_{\Delta_{\mathrm{ref}}} := B_V + 1 + \beta H \log C_{\mathrm{ref}}(\varepsilon_2).$$

***Proof of Lemma C.6.*** For every $\pi \in \Pi_{\varepsilon_2}$,

$$\begin{aligned} |G_{\mathrm{ref}}(x, y; \pi)| &\le |R(x, y)| + \beta \sum_{i=0}^{H-1} \left|\log \frac{\pi(a_i \mid s_i)}{\pi_{\mathrm{ref}}(a_i \mid s_i)}\right| \\ &\le 1 + \beta H \log C_{\mathrm{ref}}(\varepsilon_2). \end{aligned}$$

Together with $|V(x)| \le B_V$, this gives $|\Delta_{\mathrm{ref}}(x, y; \pi, V)| \le B_{\Delta_{\mathrm{ref}}}$. □

**Lemma C.7.** *With probability at least $1-\delta$, simultaneously for all $t = 1, \dots, T$,*

$$\sup_{\pi \in \Pi_{\varepsilon_2},\, V \in \mathcal{V}} \left| \widehat{\mathcal{L}}_{\mathrm{ref},t}(\pi, V) - \mathcal{L}^{t-1}_{\mathrm{ref},t}(\pi, V) \right| \le \epsilon_{\mathrm{ref},t},$$

*where* $\epsilon_{\mathrm{ref},t} := B^2_{\Delta_{\mathrm{ref}}} \sqrt{\log(2T|\Pi_{\varepsilon_2}||\mathcal{V}|/\delta)/(2n)}$.

***Proof of Lemma C.7.*** Conditioned on $\mathcal{F}_{t-1}$, the samples are independent under $\rho(x)\pi_{t-1}(y \mid x)$. For every fixed $(\pi, V) \in \Pi_{\varepsilon_2} \times \mathcal{V}$,

$$\mathbb{P}\left( \left| \widehat{\mathcal{L}}_{\mathrm{ref},t}(\pi, V) - \mathcal{L}^{t-1}_{\mathrm{ref},t}(\pi, V) \right| \ge \epsilon \,\middle|\, \mathcal{F}_{t-1} \right) \le 2 \exp\left( -\frac{2n\epsilon^2}{B^4_{\Delta_{\mathrm{ref}}}} \right).$$

Taking a union bound over $\Pi_{\varepsilon_2} \times \mathcal{V}$ conditionally on $\mathcal{F}_{t-1}$, then taking expectations with respect to $\mathcal{F}_{t-1}$ and applying a union bound over $t = 1, \dots, T$, proves the claim. □

**Lemma C.8.** *On the event in Lemma C.7,*

$$\mathcal{L}^{t-1}_{\mathrm{ref},t}(\widehat{\pi}_{\mathrm{ref},t}, \widehat{V}_{\mathrm{ref},t}) \le \varepsilon_{\mathrm{ref},t} + 2\epsilon_{\mathrm{ref},t}.$$

***Proof of Lemma C.8.*** Since $\pi^\dagger \in \Pi_{\varepsilon_2}$,

$$\begin{aligned}
\mathcal{L}^{t-1}_{\mathrm{ref},t}(\widehat{\pi}_{\mathrm{ref},t}, \widehat{V}_{\mathrm{ref},t}) &\le \widehat{\mathcal{L}}_{\mathrm{ref},t}(\widehat{\pi}_{\mathrm{ref},t}, \widehat{V}_{\mathrm{ref},t}) + \epsilon_{\mathrm{ref},t} \\
&\le \widehat{\mathcal{L}}_{\mathrm{ref},t}(\pi^\dagger, V^\dagger_{\mathrm{ref},t}) + \epsilon_{\mathrm{ref},t} \\
&\le \mathcal{L}^{t-1}_{\mathrm{ref},t}(\pi^\dagger, V^\dagger_{\mathrm{ref},t}) + 2\epsilon_{\mathrm{ref},t} \\
&= \varepsilon_{\mathrm{ref},t} + 2\epsilon_{\mathrm{ref},t}.
\end{aligned}$$

□

**Lemma C.9.** *For any $\pi \in \Pi_{\varepsilon_2}$ and $V \in \mathcal{V}$,*

$$|\mathbb{E}_{x\sim\rho}[V(x)] - J(\pi)| \le \sqrt{\widetilde{C}_t} \sqrt{\mathcal{L}^{t-1}_{\mathrm{ref},t}(\pi, V)}.$$

***Proof of Lemma C.9.*** By the definitions of $\Delta_{\mathrm{ref}}$ and $J$,

$$\mathbb{E}_{\substack{x\sim\rho \\ y\sim\pi(\cdot|x)}} [\Delta_{\mathrm{ref}}(x, y; \pi, V)] = \mathbb{E}_{x\sim\rho}[V(x)] - J(\pi).$$

Consequently,

$$\begin{aligned}
|\mathbb{E}_{x\sim\rho}[V(x)] - J(\pi)| &\le \sqrt{\mathcal{L}^{\pi}_{\mathrm{ref}}(\pi, V)}, \\
\mathcal{L}^{\pi}_{\mathrm{ref}}(\pi, V) &= \mathbb{E}_{\substack{x\sim\rho \\ y\sim\pi_{t-1}(\cdot|x)}} \left[ \frac{\pi(y \mid x)}{\pi_{t-1}(y \mid x)} \Delta_{\mathrm{ref}}(x, y; \pi, V)^2 \right] \\
&\le \widetilde{C}_t \mathcal{L}^{t-1}_{\mathrm{ref},t}(\pi, V).
\end{aligned}$$

Combining the two inequalities proves the result. □

**Lemma C.10.** *Suppose that $\mathbb{E}_{x\sim\rho}[\widehat{V}_{\mathrm{ref},t}(x)] \ge \mathbb{E}_{x\sim\rho}[V^\dagger_{\mathrm{ref},t}(x)] - \eta_{\mathrm{ref},t}$. On the event in Lemma C.7,*

$$J(\pi^\dagger) - J(\widehat{\pi}_{\mathrm{ref},t}) \le \eta_{\mathrm{ref},t} + \sqrt{\widetilde{C}_t} \left( \sqrt{\varepsilon_{\mathrm{ref},t}} + \sqrt{\varepsilon_{\mathrm{ref},t} + 2\epsilon_{\mathrm{ref},t}} \right).$$

***Proof of Lemma C.10.*** Using Lemmas C.9 and C.8,

$$\begin{aligned} J(\pi^\dagger) - J(\widehat{\pi}_{\mathrm{ref},t}) &\le \left| J(\pi^\dagger) - \mathbb{E}_{x\sim\rho}[V^\dagger_{\mathrm{ref},t}(x)] \right| + \eta_{\mathrm{ref},t} + \left| \mathbb{E}_{x\sim\rho}[\widehat{V}_{\mathrm{ref},t}(x)] - J(\widehat{\pi}_{\mathrm{ref},t}) \right| \\ &\le \eta_{\mathrm{ref},t} + \sqrt{\widetilde{C}_t} \left( \sqrt{\mathcal{L}^{t-1}_{\mathrm{ref},t}(\pi^\dagger, V^\dagger_{\mathrm{ref},t})} + \sqrt{\mathcal{L}^{t-1}_{\mathrm{ref},t}(\widehat{\pi}_{\mathrm{ref},t}, \widehat{V}_{\mathrm{ref},t})} \right) \\ &\le \eta_{\mathrm{ref},t} + \sqrt{\widetilde{C}_t} \left( \sqrt{\varepsilon_{\mathrm{ref},t}} + \sqrt{\varepsilon_{\mathrm{ref},t} + 2\epsilon_{\mathrm{ref},t}} \right). \end{aligned}$$

□

Consequently, summing Lemma C.10 over $t = 1, \ldots, T$ gives

$$\sum_{t=1}^{T} \left[ J(\pi^\dagger) - J(\widehat{\pi}_{\mathrm{ref},t}) \right] \le \sum_{t=1}^{T} \left[ \eta_{\mathrm{ref},t} + \sqrt{\widetilde{C}_t} \left( \sqrt{\varepsilon_{\mathrm{ref},t}} + \sqrt{\varepsilon_{\mathrm{ref},t} + 2\epsilon_{\mathrm{ref},t}} \right) \right].$$

***Proof of Theorem 5.3.*** Theorem 5.2 and Lemma C.10 yield the following per-iteration bounds:

$$\begin{aligned} \mathfrak{B}_t &:= \eta_t + \sqrt{C_{t-1}(\delta_t)} \left( \sqrt{\alpha_t} + \sqrt{\alpha_t + 2\epsilon_t} \right) + \zeta\delta_t, \\ \mathfrak{B}_{\mathrm{ref},t} &:= \eta_{\mathrm{ref},t} + \sqrt{\widetilde{C}_t} \left( \sqrt{\varepsilon_{\mathrm{ref},t}} + \sqrt{\varepsilon_{\mathrm{ref},t} + 2\epsilon_{\mathrm{ref},t}} \right). \end{aligned}$$

The proposed bound is tighter whenever $\mathfrak{B}_t \le \mathfrak{B}_{\mathrm{ref},t}$. Solving this inequality for $\widetilde{C}_t$ gives the exact sufficient condition

$$\widetilde{C}_t \ge C_{t-1}(\delta_t) \left( \frac{\sqrt{\alpha_t} + \sqrt{\alpha_t + 2\epsilon_t} + \frac{\eta_t - \eta_{\mathrm{ref},t} + \zeta\delta_t}{\sqrt{C_{t-1}(\delta_t)}}}{\sqrt{\varepsilon_{\mathrm{ref},t}} + \sqrt{\varepsilon_{\mathrm{ref},t} + 2\epsilon_{\mathrm{ref},t}}} \right)_+^2,$$

where $(z)_+ := \max\{z, 0\}$.

When $n$ is sufficiently large, $\epsilon_t$ and $\epsilon_{\mathrm{ref},t}$ are negligible. If $\eta_t$ and $\eta_{\mathrm{ref},t}$ are comparable and $\zeta\delta_t$ is lower order, then

$$\begin{aligned} \widetilde{C}_t &\gtrsim C_{t-1}(\delta_t) \left( \frac{2\sqrt{\alpha_t}}{2\sqrt{\varepsilon_{\mathrm{ref},t}}} \right)^2 \\ &= \frac{\alpha_t}{\varepsilon_{\mathrm{ref},t}} C_{t-1}(\delta_t). \end{aligned}$$

When $\alpha_t \asymp \varepsilon_{\mathrm{ref},t}$, this further reduces to $\widetilde{C}_t \gtrsim C_{t-1}(\delta_t)$. □

## D Implementation Details

**Training configurations.** We implement the proposed method and the baselines in the VERL framework (Sheng et al., 2024) (Apache license 2.0). All experiments are conducted on a shared compute cluster using four NVIDIA GPUs. All baselines follow the recommended hyperparameter settings. The prompt length is set to 1024 tokens and the response length to 2048 tokens. We use a training batch size of 128. Unless otherwise specified, PPO, OREO, and our method use one rollout per prompt ($n = 1$), whereas GRPO and the $n = 4$ ablation use four rollouts per prompt. All responses are sampled with a temperature of 1.0. For a fair comparison, all methods are trained for the same number of steps and are evaluated at convergence under identical step budgets. For system settings, we use vLLM rollout with tensor parallel size matched to the number of visible GPUs, and set gpu_memory_utilization $= 0.5$. Our anonymous code is available at `https://anonymous.4open.science/r/Anonymous_code_ICLR2027_1-FF4C/README.md`.

For Qwen3-4B, the learning rate is $1 \times 10^{-6}$. Both $\beta$ and $\zeta$ are set to $1 \times 10^{-3}$. The Qwen3-4B is trained for 200 steps. For DeepSeek-R1-Distill-Qwen-1.5B, the learning rate is set to $2 \times 10^{-6}$. $\beta$ is set to $1 \times 10^{-3}$, $\zeta$ is set to 0.3. The DeepSeek-R1-Distill-Qwen-1.5B is trained for 800 steps.

In implementation, we instantiate the policy and value models as $\pi = \pi_\theta$ and $V = V_\phi$, and minimize $\widehat{\mathcal{L}}_t(\pi_\theta, V_\phi)$ over the model parameters $(\theta, \phi)$. For practical implementation, we further modify our loss function to be the following form:

$$L_\pi^{\text{resp}}(\theta) = \frac{1}{n}\sum_{k=1}^{n}\Big(V_\phi(x_k) - V_\phi(s_{H-1}^k) - R(x_k, y_k) + \beta\sum_{i=0}^{H-1}\log\frac{\pi_\theta(a_i^k \mid s_i^k)}{\pi_{ref}(a_i^k \mid s_i^k)} + \zeta\sum_{i=0}^{H-1}\log\frac{\pi_\theta(a_i^k \mid s_i^k)}{\pi_{t-1}(a_i^k \mid s_i^k)}\Big)^2$$

$$L_V(\phi) = \frac{1}{n}\sum_{k=1}^{n}\frac{1}{H}\sum_{i=0}^{H-1}\Big(V_\phi(s_i^k) - R(x_k, y_k) + \beta\sum_{i=t}^{T}\log\frac{\pi_\theta(a_i^k \mid s_i^k)}{\pi_{ref}(a_i^k \mid s_i^k)}\Big)^2.$$

**Evaluation benchmarks.** We evaluate our method on six mathematical reasoning benchmarks. AIME24 and AIME25 consist of competition-level problems from the American Invitational Mathematics Examination, requiring multi-step reasoning in algebra, geometry, number theory, and combinatorics. AMC23 contains problems from the 2023 American Mathematics Competition and evaluates general high-school mathematical problem solving with a broader range of difficulty. MATH500 is a commonly used 500-problem subset of the MATH benchmark, covering diverse competition-style mathematical topics and requiring precise final-answer reasoning. Minerva Math is a challenging quantitative reasoning benchmark used to evaluate advanced mathematical problem-solving ability. OlympiadBench contains olympiad-level scientific and mathematical problems and is designed to test more difficult multi-step reasoning beyond standard school-level benchmarks. For AIME24, AIME25, and AMC23, we sample 32 responses per problem and report Avg@32 due to the small number of test problems. For MATH500, Minerva Math, and OlympiadBench, we generate one response per problem and report Avg@1.

# E Additional Works

## E.1 Value-based RL

The value-based and policy-based methods are two major categories of reinforcement learning algorithms. For value-based methods, we aim to learn an approximation of the optimal value/action-value function and choose actions by greedy maximization of these values. In contrast, policy-based methods parameterize the policy and maximize it directly using policy gradients. Classical value-based reinforcement learning methods are often framed through approximate dynamic programming and temporal-difference (TD) learning (Tesauro et al., 1995). Representative examples include Q-learning (Watkins & Dayan, 1992), SARSA (Sutton et al., 1998), and the Deep Q-Network (DQN) (Mnih et al., 2015), and entropy regularized learning (Haarnoja et al., 2018; Nachum et al., 2017a;b) along with variants that incorporate function approximation (Sutton, 1988) and residual updates (Baird, 1995). In particular, entropy-regularized reinforcement learning: PCL and Trust-PCL characterize optimal policies and values in entropy- and KL-regularized reinforcement learning through multi-step path-consistency conditions (Nachum et al., 2017a;b). More recently, Deleu et al. (2025) establish an equivalence between Relative Trajectory Balance and Trust-PCL, clarifying the connection between trajectory-balance objectives and KL-regularized RL. Their result establishes an objective-level equivalence; our analysis addresses a complementary structural question by characterizing the reasoning flow induced by the RLVR objective, including its latent credit assignment, target terminal distribution, and diversity properties. Combining strengths of policy-based and value-based methods, actor-critic algorithms (Konda & Tsitsiklis, 1999) use a critic trained via function approximation, estimating a value function to provide a learning signal that updates the actor's policy parameters toward improved performance.

## E.2 Reinforcement Learning with Verifiable Rewards

More recently, RL has been pushed from general alignment toward explicit reasoning improvements: OpenAI's O1 (Jaech et al., 2024) and DeepSeek's R1 (Guo et al., 2025) popularize framing response generation as an MDP and leveraging rule-based or verifier-style reward signals to train for correctness and reasoning quality. Following up, a growing body of "reasoning RL" studies predominantly

utilize policy-based methods such as PPO and GRPO (Shao et al., 2024; Guo et al., 2025), and have been shown to be effective in improving LLM reasoning capabilities. Some of the variants include (Yu et al., 2025; Zheng et al., 2025; Liu et al., 2025). In contrast to these policy-based approaches, several recent studies have explored value-based methods for LLM reasoning. Representative examples include TBRM (Yuan et al., 2025b), OREO (Wang et al., 2025), and VC-PPO (Yuan et al., 2025a) shows that value optimization is the key to making PPO work for long chain-of-thought reasoning, and VAPO (Yue et al., 2025b) extends this line into a stronger value-model-based reasoning RL framework.

### E.3 GFlowNet

Generative Flow Network (GFlowNet) (Bengio et al., 2021; 2023) with trajectory balance objectives (Malkin et al., 2022a; Venkatraman et al., 2024; Madan et al., 2023) is proposed to sequentially generate an object with probability proportional to a given positive reward for that object. Recent work has tried to connect GFlowNet to LLMs (Yu, 2024; Zhu et al., 2025), but these efforts primarily emphasize divergent solution generation and reward-distribution matching rather than constrained multi-step reasoning with an explicit reference policy. In particular, Flow of Reasoning Yu (2024) models the reasoning process as a DAG-structured Markovian flow and adapts GFlowNet objectives for reward-proportional divergent generation. In parallel, (Hu et al., 2023) interprets chain-of-thought as a latent-variable modeling problem within GFlowNet. Overall, much of the existing literature applies the GFlowNet framework to LLM reasoning and empirically validates it, but offers limited systematic analysis of why the flow-matching–based formulation is well-suited to reasoning or of what structural assumptions justify the underlying reasoning graph. Our reasoning DAG serves a different role: rather than supplying the training objective, it provides an analytical representation for characterizing the reasoning flow induced by our independently derived RLVR objective. This allows us to study how terminal rewards and policy priors jointly shape latent credit assignment and the resulting reasoning distribution.