GAIT: GEOMETRY-BASED CREDIT ASSIGNMENT FROM HIDDEN-STATE TRAJECTORIES FOR RLVR

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) has become a popular approach for improving LLM reasoning. However, credit assignment in mainstream RLVR methods remains coarse-grained: a sequence-level reward is typically propagated across the entire response, regardless of which reasoning steps are essential for the final outcome. Existing approaches to step-level credit assignment often rely on additional supervision that is costly to obtain, such as process reward models, or computationally expensive Monte Carlo sampling that branches from intermediate steps. In this work, we propose **G**eometry-based Credit **A**ssignment from H**I**dden-state **T**rajectories (GAIT), a plug-and-play method that derives step-level credits directly from the model's hidden-state trajectories, without requiring any external step-level supervisor. To compare the geometry of responses with different lengths and reasoning paces, GAIT aligns each response with successful responses from the same rollout group using dynamic time warping, an algorithm for matching temporal sequences that may differ in length. It then measures stepwise divergence in a lower-dimensional representation space, and uses temporal changes in this divergence to reweight the sequence-level advantage across reasoning steps. Intuitively, this assigns stronger negative updates to steps in incorrect responses that increasingly diverge from successful trajectories, and stronger positive updates to atypical but successful transitions in correct responses. Across model scales and families, GAIT outperforms existing RLVR baselines with and without fine-grained credit assignment. On challenging AIME 2024–2026 benchmarks with Qwen3-8B-Base as the base model, GAIT improves over DAPO by 2.38 points and reaches DAPO's peak performance with 1.31$\times$ fewer GPU hours, while adding negligible computational overhead to each RL update.

1 INTRODUCTION

Large language models (LLMs) (OpenAI, 2024; Guo et al., 2025) increasingly rely on inference-time scaling to solve complex reasoning tasks, generating long chains of thought (CoTs) (Wei et al., 2022) that may contain hundreds or even thousands of intermediate reasoning steps. Reinforcement learning (RL) with verifiable rewards (RLVR) has emerged as a popular approach for developing such long-form reasoning capabilities. Typical RLVR algorithms, such as Group Relative Policy Optimization (GRPO) (Shao et al., 2024) and Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO) (Yu et al., 2025), sample a group of responses from the current policy for each prompt, evaluate them using a verifier, and optimize the policy based on the resulting rewards.

A key limitation of RLVR is its coarse-grained credit assignment. Their verifiers are typically outcome-based, and the sequence-level advantage derived from the final reward is broadcast uniformly to all tokens in a response. As long as two reasoning steps belong to the same response and share a single final reward, they receive the same learning signal, regardless of how differently they contribute to the outcome. For example, within an incorrect response, a critical step introducing an error is penalized no more strongly than a correct step or an inconsequential step. Similarly, within a correct response, a pivotal step receives no more credit than a routine one. Assigning finer-grained credits is therefore an important challenge for RLVR. Existing approaches to step-level credit assignment broadly fall into two categories. Process reward models (PRMs) (Lightman et al., 2023; Wang et al., 2024; Zhang et al., 2025b) rely on step-level supervision obtained through costly human annotation, strong LLM judges, or Monte Carlo (MC) sampling. Meanwhile, tree-based RL meth-

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

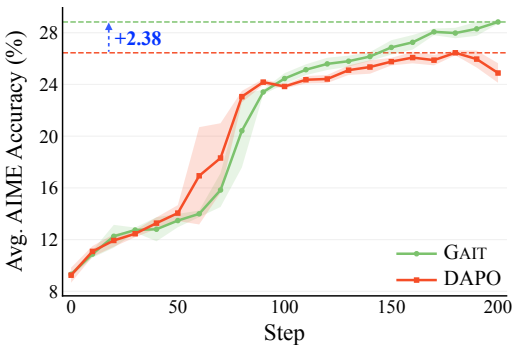


Figure 1: **Training dynamics of DAPO and our method, GAIT.** During RL training, GAIT consistently outperforms DAPO on three AIME benchmarks (AIME 2024–2026) from step 100 onward. Ultimately, GAIT exceeds DAPO's peak performance by 2.38 points.

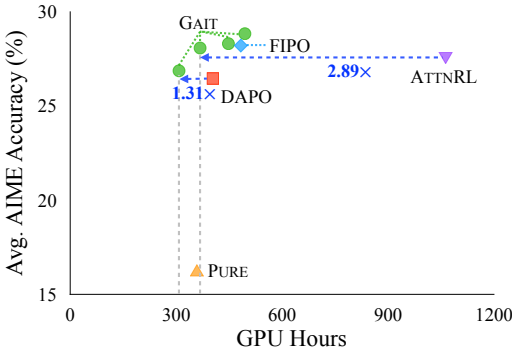


Figure 2: **Training efficiency.** For each baseline, we report its best AIME accuracy and corresponding GPU hours. For GAIT, we report AIME performance and GPU hours across different RL steps. GAIT achieves higher performance with fewer GPU hours than all baselines.

ods (Hou et al., 2025; Liu et al., 2026) estimate the values of reasoning prefixes via additional MC sampling during rollouts. Both introduce substantial overhead in data collection and computation.

In this work, we explore an alternative source of step-level credit that is nearly free to obtain: the model's own *internal representations*. Our key intuition is that the geometry of hidden-state trajectories reveals how reasoning evolves over time. In particular, (i) *an incorrect trajectory may initially resemble successful reasoning before diverging as erroneous reasoning develops*, whereas (ii) *a correct trajectory may deviate from common solution paths, reflecting useful alternative reasoning strategies*. Motivated by these observations, we introduce **G**eometry-based Credit **A**ssignment from HIdden-state **T**rajectories (GAIT), a novel plug-and-play method that reweights sequence-level credit across reasoning steps without requiring any external step-level supervisor. The method adds only negligible computational overhead to each RL step (see Figure 8).

Our approach proceeds in three stages (see Figure 3): (i) reasoning trajectory alignment, (ii) divergence computation, and (iii) advantage reweighting. Comparing the geometry of reasoning trajectories is challenging because responses differ in both length and reasoning pace, making position-wise comparisons unreliable. Therefore, we first project step-level hidden states into a lower-dimensional space and align each response with correct responses from the same rollout group using dynamic time warping (DTW) (Sakoe & Chiba, 1990), an algorithm for aligning sequences that progress at different rates. Second, we measure each step's divergence from the aligned correct trajectories and compute how this divergence changes across consecutive steps. Finally, we use the divergence changes across steps to reweight the sequence-level advantage at the step level. For incorrect responses, steps that increasingly diverge from correct reasoning receive stronger negative updates; for correct responses, atypical but successful reasoning transitions receive stronger positive updates.

Experiments across different model scales and families show that GAIT consistently outperforms RLVR baselines spanning different granularities of credit assignment. Using Qwen3-8B-Base (Yang et al., 2025) as the base model and AIME 2024–2026 (Hugging Face H4, 2024; MathArena, 2025; 2026) for evaluation, GAIT outperforms DAPO by 2.38 points, as shown in Figure 1. Moreover, Figure 2 shows that GAIT achieves superior training efficiency and lies on the performance–compute Pareto frontier. For example, it surpasses DAPO's peak performance using 1.31× fewer GPU hours and surpasses ATTNRL (Liu et al., 2026), a tree-based RL method, using 2.89× fewer GPU hours.

2 Related Work

RL for LLMs. RL is a key component of LLM post-training. In RLHF (Ouyang et al., 2022), Proximal Policy Optimization (PPO) (Schulman et al., 2017) is widely used. It relies on a learned value network to estimate token-level state values and compute advantages. More recently, RLVR has emerged as a popular paradigm for improving LLM reasoning. Critic-free methods such as

GRPO (Shao et al., 2024), DAPO (Yu et al., 2025), and GSPO (Zheng et al., 2025) reduce the computational overhead of PPO by estimating advantages from groups of sampled responses. However, these methods typically assign the same sequence-level advantage to all tokens within a response, providing only coarse-grained credit for intermediate reasoning steps. Fine-grained credit assignment therefore remains a central challenge for effective and efficient RLVR.

Credit Assignment for RLVR. Existing approaches to fine-grained credit assignment primarily operate at the reasoning-step or token level. Step-level methods typically rely on PRMs or tree-based rollouts. PRM-based methods evaluate intermediate reasoning steps using either specially trained reward models or prompted LLM judges. Trained PRMs require step-level supervision obtained from human annotations, LLM judges, or MC sampling (Lightman et al., 2023; Wang et al., 2024; Zhang et al., 2025b; Zou et al., 2025), incurring substantial costs for data generation and reward-model training. They may also suffer from policy distribution shift and reward hacking (Cheng et al., 2025). Alternatively, methods such as CAPO (Xie et al., 2025) and SRAR (Xie et al., 2026) use prompted LLMs as generative PRMs, avoiding dedicated reward-model training but often requiring a stronger LLM to provide reliable step-level evaluation. Tree-based methods, such as TREERL (Hou et al., 2025) and ATTNRL (Liu et al., 2026), estimate step-level credit by branching from intermediate reasoning states and evaluating sampled MC continuations. While avoiding separate PRMs, they require tree construction and repeated continuation rollouts.

Token-level methods instead derive credit signals from policy statistics. GTPO (Tan et al., 2026) constructs token-specific advantages using entropy-weighted rewards, while FIPO (Ma et al., 2026) reweights outcome-based advantages using future Kullback–Leibler (KL) signals between the current and rollout policies. Because these signals are derived from policy statistics rather than directly evaluating intermediate reasoning states, their connection to reasoning progress is unclear.

Internal Representations for Reasoning. Recent work suggests that LLM internal representations encode informative signals about reasoning behavior and quality. Probing studies show that hidden states contain information about reasoning correctness (Zhang et al., 2025a) and truthfulness (Hu et al., 2026), while trajectory-based approaches (Jiang et al., 2026; Gjølbye et al., 2026) examine how representations evolve throughout the reasoning process. In particular, Jiang et al. (2026) find that correct and incorrect reasoning exhibit distinct geometric properties in hidden-state trajectories, such as displacement and curvature. These findings motivate our use of internal representations for step-level credit assignment. Notably, our geometric signal differs from those examined before.

3 PRELIMINARIES

GRPO. GRPO (Shao et al., 2024) is a critic-free RLVR algorithm commonly used for LLM post-training. Given a prompt q, GRPO samples a group of G responses $\mathcal{G} = \{o_i\}_{i=1}^{G}$ from the old policy $\pi_{\theta_{\text{old}}}$. A verifier assigns each response o_i a sequence-level reward r_i. For example, a binary correctness verifier assigns $r_i = 1$ if the answer extracted from o_i matches the ground truth, and $r_i = -1$ otherwise. We denote the subsets of correct and incorrect responses in the group by $\mathcal{G}^+ = \{o_i \in \mathcal{G} : r_i = 1\}$ and $\mathcal{G}^- = \{o_i \in \mathcal{G} : r_i = -1\}$, respectively. GRPO normalizes rewards within each group to compute a sequence-level advantage:

$$\hat{A}_i = \frac{r_i - \text{mean}\left(\{r_k\}_{k=1}^{G}\right)}{\text{std}\left(\{r_k\}_{k=1}^{G}\right)}. \tag{1}$$

This advantage is shared by all tokens in o_i. GRPO then maximizes the clipped surrogate objective:

$$\begin{aligned}\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\substack{q\sim\mathcal{D}, \\ \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)}} \Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\Big(\min\big(\rho_{i,t}(\theta)\hat{A}_i, \\ \text{clip}\left(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_i\big) - \beta D_{\text{KL}}\left(\pi_\theta \,\|\, \pi_{\text{ref}}\right)\Big)\Bigg],\end{aligned} \tag{2}$$

where $\rho_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}$ is the token-level importance ratio, ϵ determines the clipping range, and β controls the KL regularization toward a reference policy π_{ref}. Notably, GRPO first averages the token-level losses within each response and then averages across responses, giving each response equal weight regardless of its length.

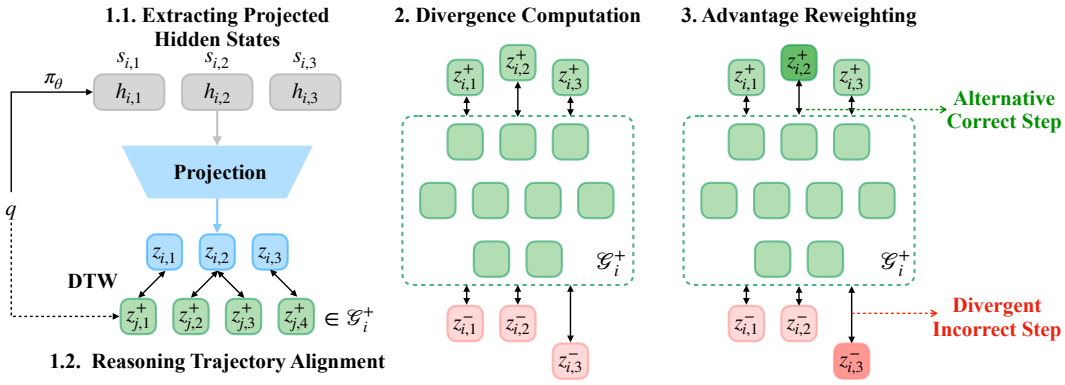


Figure 3: **An Overview of GAIT.** Superscript "+" and green denote steps in correct responses, while superscript "−" and red denote steps in incorrect responses. (1.1) We represent each reasoning step $s_{i,t}$ by its projected hidden state $z_{i,t}$, and (1.2) align the resulting trajectory with correct reference trajectories in $\mathcal{G}_i^+$ (Section 4.1). (2) We compute step-level divergences from the aligned correct references and their temporal changes (Section 4.2). (3) We use these temporal divergence changes to reweight the sequence-level advantage across reasoning steps (Section 4.3).

DAPO. DAPO (Yu et al., 2025) modifies GRPO with token-level loss aggregation, dynamic sampling, and asymmetric clipping, while removing KL regularization. Dynamic sampling discards groups with identical rewards, because they provide no group-relative learning signal (*i.e.*, $\hat{A}_i = 0$). It resamples until the training batch is filled with groups containing both correct and incorrect responses (*i.e.*, $\mathcal{G}^+ \neq \emptyset$ and $\mathcal{G}^- \neq \emptyset$). As in GRPO, all tokens in a response share the same sequence-level advantage. We provide additional details and the optimization objective in Appendix A.

4 Our Credit Assignment Approach: GAIT

Typical RLVR methods assign the same sequence-level advantage $\hat{A}_i$ to every token in a response o_i, even though different reasoning steps may contribute differently to the final outcome. To provide finer-grained credit, we propose GAIT, a step-level credit assignment approach based on the geometry of hidden-state trajectories.

GAIT is motivated by two intuitions. First, *an incorrect trajectory may initially resemble successful reasoning in the hidden state space before diverging as erroneous reasoning develops*. Thus, the hidden state trajectories of successful reasoning provide natural references for identifying where incorrect reasoning begins to depart, without requiring an external judge or PRM. Second, *a correct trajectory may also depart from common successful reasoning paths in the hidden state space, reflecting a useful alternative reasoning strategy*. Thus, hidden states can help identify atypical steps in correct responses, which we can reinforce to encourage reasoning diversity. Together, these intuitions suggest that the temporal evolution of divergence from successful reasoning can provide informative step-level credit. Accordingly, GAIT uses correct responses within the same rollout group as references and reweights the sequence-level advantage across reasoning steps according to their hidden-state geometry.

In GAIT, we split each response o_i into T_i steps $\{s_{i,1}, \ldots, s_{i,T_i}\}$ at double newlines ("\n\n") and assign a separate advantage to each reasoning step. For response o_i, we denote its set of correct references by $\mathcal{G}_i^+ = \{o_j^+ \in \mathcal{G}^+, j \neq i\}$. Figure 3 shows that GAIT has three stages. First, we represent each reasoning step $s_{i,t}$ by the projected hidden state $z_{i,t}$ of its final token and align the resulting trajectory with each correct reference in $\mathcal{G}_i^+$. Second, we measure how much each step diverges from the geometry of the aligned correct trajectories and track how this divergence changes over time. Finally, we use these temporal changes to reweight the sequence-level advantage across reasoning steps, and broadcast each step-level advantage to all tokens within that step. We also summarize the complete process in Algorithm 1.

Importantly, GAIT relies only on hidden states that are nearly free to obtain during training and requires no external step-level supervision. In the following sections, we first describe trajectory

alignment in Section 4.1 and define our divergence signal in Section 4.2. We then present pilot experiments that examine the motivating intuitions in Section 4.2.1. Finally, building on these observations, we introduce our advantage reweighting scheme in Section 4.3.

4.1 Reasoning Trajectory Alignment

We represent each response as a trajectory in hidden-state space. For the t-th reasoning step $s_{i,t}$ of a response o_i sampled from the old policy $\pi_{\theta_{\text{old}}}$, let $h_{i,t}^{(l)}$ denote the layer-l hidden state of its final token under $\pi_{\theta_{\text{old}}}$. We omit the layer index l hereafter for brevity. To obtain a compact trajectory representation, we then apply HARP (Hu et al., 2026) projection:

$$z_{i,t} = \text{HARP}(h_{i,t}), \tag{3}$$

retaining 5% of the original dimensions. We use this projection to suppress variation that is less relevant to the reasoning progress and reduce the computational cost of subsequent alignment.

Directly comparing steps with the same index is unreliable because different responses may contain different numbers of steps and may progress through the solution at different rates. We therefore align their hidden-state trajectories using dynamic time warping (DTW) (Sakoe & Chiba, 1990). Consider a query response o_i with T_i steps and a correct reference $o_j^+ \in \mathcal{G}_i^+$ with T_j steps. For a query step $s_{i,t} \in o_i$ and a reference step $s_{j,k}^+ \in o_j^+$, we define their distance in the projected hidden-state space as: $c(s_{i,t}, s_{j,k}^+) = \|z_{i,t} - z_{j,k}^+\|_2$. DTW uses dynamic programming to find the monotonic alignment path that minimizes the total pairwise distance between o_i and o_j^+:

$$P^*\left(o_i, o_j^+\right) = \underset{P \in \mathcal{P}(T_i, T_j)}{\arg\min} \sum_{(t,k)\in P} c\left(s_{i,t}, s_{j,k}^+\right), \tag{4}$$

where $\mathcal{P}\left(T_i, T_j\right)$ is the set of valid paths, consisting of step index pairs (t, k) that connect $(1, 1)$ to (T_i, T_j) with transitions $(+1, 0)$, $(0, +1)$, or $(+1, +1)$. Because DTW may align one query step with one or multiple reference steps, we define

$$\mathcal{K}\left(s_{i,t}, o_j^+\right) = \{k : (t,k) \in P^*\left(o_i, o_j^+\right)\},$$
$$a\left(s_{i,t}, o_j^+\right) = \frac{1}{|\mathcal{K}\left(s_{i,t}, o_j^+\right)|} \sum_{k \in \mathcal{K}\left(s_{i,t}, o_j^+\right)} c\left(s_{i,t}, s_{j,k}^+\right). \tag{5}$$

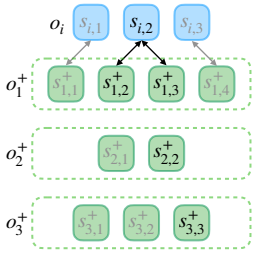


Figure 4: **An example of trajectories aligned by DTW.** The query response o_i is aligned with its correct reference set $\mathcal{G}_i^+ = \{o_1^+, o_2^+, o_3^+\}$, where $s_{i,2}$ in o_i and steps aligned with it are highlighted.

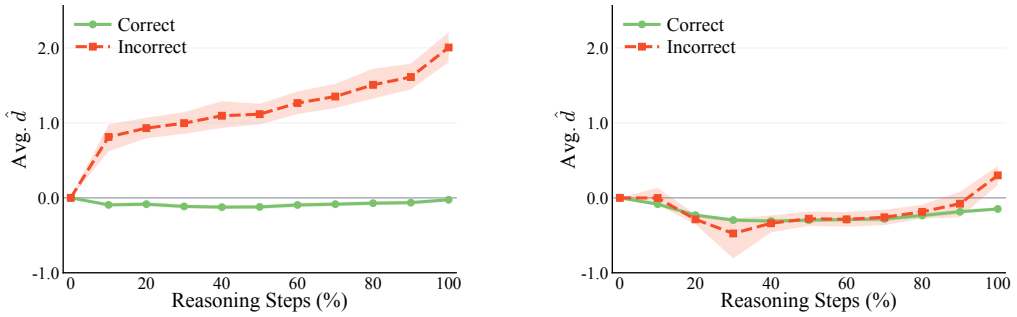
Here, $\mathcal{K}(s_{i,t}, o_j^+)$ contains the indices of reference steps aligned with $s_{i,t}$, while $a(s_{i,t}, o_j^+)$ measures the average distance between $s_{i,t}$ and its aligned portion of reference trajectory o_j^+. For example, in Figure 4, the alignment path between o_i and o_1^+ after DTW is $P^*(o_i, o_1^+) = \{(1,1), (2,2), (2,3), (3,4)\}$. Thus, $\mathcal{K}(s_{i,2}, o_1^+) = \{2, 3\}$ because $s_{1,2}^+$ and $s_{1,3}^+$ are both aligned with $s_{i,2}$, and $a(s_{i,2}, o_1^+) = (\|z_{i,2} - z_{1,2}^+\|_2 + \|z_{i,2} - z_{1,3}^+\|_2)/2$.

4.2 Divergence Computation

We next quantify how strongly each query step in o_i diverges from the geometry of its correct reference set $\mathcal{G}_i^+$. For brevity, we write $j \in \mathcal{G}_i^+$ to index a correct reference trajectory $o_j^+ \in \mathcal{G}_i^+$. For each step $s_{i,t}$, we aggregate its alignment distances across $\mathcal{G}_i^+$ using a softmin-weighted average:

$$w\left(s_{i,t}, o_j^+\right) = \frac{\exp\left\{-a\left(s_{i,t}, o_j^+\right)\right\}}{\sum_{j' \in \mathcal{G}_i^+} \exp\left\{-a\left(s_{i,t}, o_{j'}^+\right)\right\}}, \quad d_{i,t} = \sum_{j \in \mathcal{G}_i^+} w\left(s_{i,t}, o_j^+\right) a\left(s_{i,t}, o_j^+\right). \tag{6}$$

Here, $d_{i,t}$ measures the raw divergence of $s_{i,t}$ with respect to its correct reference set. The softmin weighting gives greater influence to reference trajectories whose aligned steps are closer to $s_{i,t}$. Consequently, each query step is compared primarily against the successful reasoning paths that are most geometrically similar to it.



(a) Qwen3-8B-Base. (b) DAPO checkpoint.

Figure 5: **Average $\hat{d}$ throughout reasoning for correct and incorrect responses from (a) Qwen3-8B-Base and (b) its DAPO checkpoint.** For both models, incorrect trajectories increasingly diverge from successful trajectories, while correct trajectories remain relatively close. After RL training using DAPO, the divergence of incorrect trajectories is substantially reduced.

Because the absolute scale of $d_{i,t}$ can vary across reasoning stages, we construct a baseline $\mathcal{B}_{i,t}$ from geometrically aligned steps in correct references and normalize $d_{i,t}$ relative to this baseline:

$$\mathcal{B}_{i,t} = \left\{ d^+_{j,k} : j \in \mathcal{G}^+_i,\ k \in \mathcal{K}\left(s_{i,t}, o^+_j\right) \right\}, \quad \hat{d}_{i,t} = \frac{d_{i,t} - \text{mean}(\mathcal{B}_{i,t})}{\text{std}(\mathcal{B}_{i,t})}. \tag{7}$$

For example, in Figure 4, $s_{i,2}$ is aligned with $s^+_{1,2}$, $s^+_{1,3}$, $s^+_{2,2}$, and $s^+_{3,3}$, so its baseline is $\mathcal{B}_{i,2} = \{d^+_{1,2}, d^+_{1,3}, d^+_{2,2}, d^+_{3,3}\}$. Here, each $d^+_{j,k}$ is computed analogously for the correct step $s^+_{j,k}$ with respect to its own correct reference set $\mathcal{G}^+_j$. We define $\hat{d}_{i,t}$ as the normalized divergence of $s_{i,t}$. It measures how unusually divergent $s_{i,t}$ is relative to correct references at a comparable reasoning stage. Finally, we measure the temporal change in normalized divergence between consecutive steps:

$$\Delta\hat{d}_{i,t} = \hat{d}_{i,t} - \hat{d}_{i,t-1}, \quad \hat{d}_{i,0} = 0. \tag{8}$$

A large positive $\Delta\hat{d}_{i,t}$ indicates that step $s_{i,t}$ moves the trajectory of o_i farther from the geometry of correct references than the preceding step.

4.2.1 Revisiting Intuitions Behind Gait

We now examine the two intuitions motivating the use of $\Delta\hat{d}_{i,t}$ for step-level credit assignment.

Intuition 1: In hidden state space, incorrect responses gradually diverge from correct ones as errors develop. We examine this intuition by tracking normalized divergences $\hat{d}$ from each response's correct reference set. For Qwen3-8B-Base (Yang et al., 2025) and its DAPO checkpoint, we randomly select 200 rollout groups per model containing both correct and incorrect responses to prompts from a training set. We divide each response into 10 segments with equal numbers of steps, average $\hat{d}$ within each segment, and then aggregate separately over correct and incorrect responses. Figure 5 shows that normalized divergence remains relatively low throughout correct responses but increases as incorrect responses progress. This pattern holds both before and after RL, with incorrect responses exhibiting substantially lower normalized divergence after RL. We provide additional response-level diagnostics in Appendix B.1. These findings support our first intuition and motivate stronger negative updates for steps with $\Delta\hat{d}_{i,t} > 0$ in incorrect responses.

Intuition 2: In hidden state space, increasing divergence in correct responses may reflect useful alternative reasoning. We examine this intuition by estimating how adding a step with positive temporal divergence ($\Delta\hat{d}_{i,t} > 0$) changes downstream continuation accuracy. For each sampled step $s_{i,t}$, we compare MC-estimated continuation accuracy from the reasoning prefixes $\{s_{i,1}, \ldots, s_{i,t-1}\}$ and $\{s_{i,1}, \ldots, s_{i,t}\}$, respectively. In Figure 7a, continuation accuracy decreases on average after such steps in incorrect responses but increases in correct responses. Thus, increased divergence can reflect harmful departures in incorrect responses but useful alternative transitions in correct ones. These results support our second intuition and motivate stronger positive updates for steps with $\Delta\hat{d}_{i,t} > 0$ in correct responses.

We observe similar patterns across different LLM families and provide results for Llama3.1-8B-Instruct (Grattafiori et al., 2024) and its corresponding RL checkpoint in Appendix B.2.

4.3 ADVANTAGE REWEIGHTING

Based on the observations above, we use the temporal divergence $\Delta\hat{d}_{i,t}$ to reweight the sequence-level advantage across reasoning steps. Specifically, for each step, we compute a clipped multiplier $m_{i,t}$ and reweight its advantage to $A_{i,t}$:

$$m_{i,t} = \text{clip}\left(1 + \alpha\Delta\hat{d}_{i,t}, 1 - \eta_{\text{low}}, 1 + \eta_{\text{high}}\right), \quad A_{i,t} = m_{i,t}\hat{A}_i. \tag{9}$$

Here, α controls the strength of the reweighting, while $[1 - \eta_{\text{low}}, 1 + \eta_{\text{high}}]$ bound the adjustment to stabilize training. The resulting $A_{i,t}$ is assigned to every token within step $s_{i,t}$. When $\Delta\hat{d}_{i,t} > 0$ and $\eta_{\text{high}} > 0$, the magnitude of the original sequence-level advantage is amplified. For an incorrect response with $\hat{A}_i < 0$, a step that newly diverges from successful reasoning receives a stronger negative update. For a correct response with $\hat{A}_i > 0$, a step that departs from common successful trajectories receives a stronger positive update, encouraging alternative successful reasoning strategies. Conversely, when $\Delta\hat{d}_{i,t} < 0$ and $\eta_{\text{low}} > 0$, the magnitude of the update is attenuated. In this way, GAIT reweights the sequence-level advantage according to where meaningful changes occur along the hidden-state trajectory, providing step-level credits without an external step-level supervisor.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Baselines. We compare GAIT with RLVR baselines spanning three levels of credit assignment: (i) **sequence-level**: GRPO (Shao et al., 2024) and DAPO (Yu et al., 2025); (ii) **step-level**: PURE (Cheng et al., 2025), which performs step-level credit assignment using a PRM trained on the human-annotated PRM800K dataset (Lightman et al., 2023), and ATTNRL (Liu et al., 2026), a tree-based RL method that uses attention-guided branching and MC rollouts; and (iii) **token-level**: FIPO (Ma et al., 2026), which reweights token-level advantages using future-KL signals.

Implementation details. We conduct our main experiments with Qwen3-8B-Base (Yang et al., 2025), initializing all methods from this base model and training on DeepMath-103K dataset (He et al., 2026). To assess generalization across model scales and families, we additionally evaluate DAPO, FIPO that is the best-performing baseline on Qwen3-8B-Base, and GAIT on Qwen3-4B-Base (Yang et al., 2025) and Llama3.1-8B-Instruct (Grattafiori et al., 2024). Each training run consists of 200 steps, with a learning rate of 1×10^{-6} and 10 linear warmup steps. We use a rollout group size of 16 for all methods except ATTNRL, which uses MC sampling, together with a training batch size of 256, a mini-batch size of 64, and a maximum response length of 12,288 tokens. We conduct all RL experiments on 8 H100 GPUs and repeat each experiment with three random seeds.

Our GRPO configuration follows the training recipe of JUSTRL (He et al., 2025). We integrate PURE's credit assignment into DAPO. For ATTNRL, we retain its original adaptive sampling mechanism, which serves a similar purpose to DAPO's dynamic sampling. FIPO is also originally built on top of DAPO.

Similarly, we plug GAIT into DAPO. For Qwen3-8B-Base, we extract hidden states from layer $l = 32$, corresponding to about 90% of model depth, and set $\alpha = 0.01$. We use $\eta_{\text{low}} = 0$ and $\eta_{\text{high}} = 0.2$. To ensure sufficient correct references for the normalization in Equation (7), we apply GAIT only to rollout groups containing at least four correct responses ($|\mathcal{G}^+| \geq 4$). For the remaining groups after DAPO's dynamic sampling, we fall back to the original DAPO advantages. We provide additional baseline details and hyperparameters for the other two models in Appendix C.

Evaluation and benchmarks. We primarily evaluate the trained models on three challenging AIME benchmarks: AIME 2024 (Hugging Face H4, 2024), AIME 2025 (MathArena, 2025), and AIME 2026 (MathArena, 2026), with additional evaluations on AMC23 (math-ai, 2023) and BeyondAIME (ByteDance-Seed, 2025). For every benchmark, following Hochlehnert et al. (2025)'s recommendation to tune decoding temperature per model, we evaluate all methods under the same

Table 1: **Pass@1 comparison between our method (GAIT), and baselines.** We report mean ± standard deviation over three RL runs with different random seeds. Best and second-best results are marked in **bold** and underlined, respectively. The green rows denote GAIT. Due to computational constraints, the other two models are evaluated only against DAPO and FIPO, the strongest baseline on Qwen3-8B-Base. Across model scales and families, GAIT achieves the best average Pass@1.

Method	AIME24	AIME25	AIME26	AMC23	BeyondAIME	Avg.
Qwen3-8B-Base						
GRPO	28.99±0.38	23.96±0.44	23.23±0.25	74.01±0.30	11.17±0.21	32.27±0.30
DAPO	30.42±0.89	24.31±0.60	24.62±0.05	78.13±1.64	12.94±1.04	34.08±0.40
PURE	16.39±1.35	17.50±0.92	14.62±0.83	65.55±0.89	9.21±0.19	24.65±0.20
ATTNRL	30.83±0.85	26.04±0.89	25.76±0.71	77.03±0.89	**13.84**±0.85	34.70±0.09
FIPO	31.84±0.94	25.97±0.39	**26.80**±0.21	77.86±0.97	13.04±0.67	35.11±0.31
GAIT	**32.43**±1.46	**27.50**±1.29	26.56±1.43	**79.04**±0.35	13.73±0.54	**35.85**±0.24
Δ vs. DAPO	**+2.01**	**+3.19**	**+1.94**	**+0.91**	**+0.79**	**+1.77**
Qwen3-4B-Base						
DAPO	22.88±0.60	19.51±0.85	17.43±1.32	67.93±1.27	**8.80**±0.32	27.31±0.42
FIPO	20.10±0.66	16.42±1.65	13.96±0.47	67.76±0.71	7.77±0.15	25.20±0.61
GAIT	**23.92**±0.72	**21.18**±0.94	**18.47**±1.36	**68.14**±1.32	8.72±0.36	**28.09**±0.24
Δ vs. DAPO	**+1.04**	**+1.67**	**+1.04**	**+0.21**	**-0.08**	**+0.78**
Llama3.1-8B-Instruct						
DAPO	10.03±0.77	**3.54**±0.81	2.61±0.23	39.48±1.05	1.94±0.11	11.52±0.18
FIPO	10.17±0.98	2.60±0.81	1.77±0.90	37.29±0.91	2.02±0.44	10.77±0.51
GAIT	**11.32**±0.87	3.06±0.72	**2.81**±0.37	**40.78**±2.38	**2.14**±0.23	**12.02**±0.73
Δ vs. DAPO	**+1.29**	**-0.48**	**+0.20**	**+1.30**	**+0.20**	**+0.50**

temperature sweep $\{0.6, 1.0\}$, sampling 32 responses per prompt at each temperature, and report the better Pass@1 (*i.e.*, Avg@32) for each model.

5.2 MAIN RESULTS

GAIT outperforms all baselines across model scales and families. Table 1 shows that GAIT achieves the highest average Pass@1 across all base models. On Qwen3-8B-Base, it improves over DAPO on all five benchmarks, raising average Pass@1 by 1.77 points, with an average gain of 2.38 points across three AIME benchmarks (see Figure 1 for training dynamics). It also outperforms FIPO, the strongest credit-assignment baseline on this model, by 0.74 points on average. Compared with the step-level credit assignment methods PURE and ATTNRL, GAIT achieves higher average Pass@1 without a separately trained PRM or additional MC rollouts for credit assignment. These gains extend to Qwen3-4B-Base and Llama3.1-8B-Instruct, where GAIT improves over DAPO on four of five benchmarks, increasing average Pass@1 by 0.78 and 0.50 points, respectively. In contrast, FIPO underperforms DAPO in both settings.

GAIT also achieves the highest average Pass@32. Besides Pass@1, the gains extend to Pass@32 across all base models (Table 4). On Qwen3-8B-Base, Qwen3-4B-Base, and Llama3.1-8B-Instruct, GAIT improves average Pass@32 over DAPO by 0.95, 2.89, and 0.85 points, respectively.

GAIT is highly training efficient and achieves higher Pass@1 with fewer GPU hours. We track average Pass@1 across the three AIME benchmarks throughout RL training with Qwen3-8B-Base. As shown in Figure 2, GAIT lies on the performance–compute Pareto frontier. It surpasses the best Pass@1 achieved by DAPO, PURE, ATTNRL, and FIPO while using fewer GPU hours than each baseline requires to reach its peak. For example, GAIT surpasses the peak performance of DAPO and ATTNRL using 1.31× and 2.89× fewer GPU hours, respectively.

GAIT requires little computation overhead for credit assignment. The runtime breakdown in Figure 8 shows that GAIT spends only 34.5 seconds on average per RL step on credit assignment, accounting for 3.1% of the measured total step time. Rollout generation and policy updates remain the dominant costs. These results show that GAIT improves both training performance and efficiency while allocating only a negligible fraction of each RL step to credit assignment.

Table 2: **Ablations of step-specific reweighting and amplification in incorrect and correct responses.** Random permutation shuffles temporal divergence signals across reasoning steps before advantage reweighting, yielding performance comparable to DAPO but substantially below GAIT. Incorrect-only and correct-only restrict advantage amplification to incorrect and correct responses, respectively. Both variants underperform GAIT that applies amplification to both response types.

Method	AIME24	AIME25	AIME26	Avg.
DAPO	30.42	24.31	24.62	26.45
GAIT w/ Random Permutation	29.58	26.15	24.06	26.60
GAIT w/ Incorrect-Only	31.53	25.94	24.38	27.28
GAIT w/ Correct-Only	29.52	24.82	26.11	26.82
GAIT	**32.43**	**27.50**	**26.56**	**28.83**

5.3 ABLATION STUDY

In this section, we ablate the main design choices of GAIT using Qwen3-8B-Base.

Effect of HARP projection. We remove HARP projection and compute credit-assignment signals directly from the original hidden states. Table 5 shows that removing HARP reduces the average Pass@1 by 0.81 points. It also increases the average credit-assignment time per RL step from 34.5 to 135.3 seconds. Thus, HARP improves credit-assignment quality while providing a 3.92× speedup.

Importance of step-specific reweighting. We randomly permute the temporal divergence values $\Delta\hat{d}_{i,t}$ across steps before advantage reweighting in Equation (9). This preserves the divergence values but breaks their correspondence with the original steps. Table 2 shows that this variant achieves an average Pass@1 comparable to DAPO and 2.23 points lower than GAIT. This result indicates that $\Delta\hat{d}_{i,t}$ identifies reasoning steps that are particularly important for credit assignment.

Amplification in incorrect and correct responses. For steps with $\Delta\hat{d}_{i,t} > 0$, GAIT amplifies the magnitude of the sequence-level advantage, strengthening negative updates in incorrect responses and positive updates in correct responses (see Equation (9)). We compare this design with two variants that restrict amplification only to incorrect responses (incorrect-only) or only to correct responses (correct-only). Table 2 shows that their average Pass@1 accuracies are 27.28 and 26.82, respectively, both below the 28.83 achieved by GAIT. Moreover, the higher performance of incorrect-only over correct-only suggests that amplifying advantages of diverging steps in incorrect responses contributes more, while applying amplification to both response types yields the best performance.

Sensitivity to hyperparameters. We vary the hidden-state extraction layer l and the reweighting strength α in Equation (9). As shown in Figure 9, GAIT outperforms DAPO at all three tested layers, $l \in \{28, 32, 36\}$, with the highest Pass@1 at $l = 32$. The final layer ($l = 36$) performs worst among these choices, suggesting that its hidden states may contain more lexical noise and less reasoning-relevant information than those of other late layers.

Similarly, GAIT outperforms DAPO for all tested values $\alpha \in \{0.0075, 0.01, 0.0125\}$, with the highest Pass@1 at $\alpha = 0.01$ (Figure 10). These results demonstrate that GAIT is robust to the choice of hyperparameters, while the setting used in Section 5.1 yields the best performance.

6 CONCLUSION

We propose GAIT, a method for step-level credit assignment in RLVR using the geometry of hidden-state trajectories. By tracking changes in divergence from correct reasoning, GAIT reweights sequence-level advantages to strengthen negative feedback for emerging errors in incorrect responses and positive feedback for atypical but successful reasoning in correct responses. GAIT requires no external step-level supervision and introduces little computational overhead. Experiments show that GAIT outperforms existing RLVR baselines across different granularities of credit assignment. Meanwhile, GAIT achieves higher training efficiency. These results highlight the potential of internal representations as an effective and efficient source of fine-grained credit for RLVR.

AI USE STATEMENT

In this work, we used generative AI tools for interpreting results. We have not used generative AI tools for designing or providing feedback on research methodology or experiments, implementing methods, and supporting qualitative and thematic data analysis; and generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing critical ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, assisting with translation, cleaning and reformatting datasets are not applicable to this work. Additionally, we used generative AI tools for suggesting experimental parameters, creating or editing software code, and editing a research paper to improve readability. We have reviewed all AI-assisted work. LLM-generated code was verified and tested for correctness by the lead authors. LLM-generated text polishing is proofread by the lead authors. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

We describe our method in Section 4 and provide implementation details in Section 5.1 and Appendix C. The model obtained by training Qwen3-8B-Base with our approach, GAIT, is available at `https://huggingface.co/artifacts-7f3a/GAIT-Qwen3-8B`. We commit to open-sourcing our codebase upon publication.

REFERENCES

ByteDance-Seed. Beyondaime: Advancing math reasoning evaluation beyond high school olympiads, 2025. URL `https://huggingface.co/datasets/ByteDance-Seed/BeyondAIME`.

Jie Cheng, Gang Xiong, Ruixi Qiao, Lijun Li, Chao Guo, Junle Wang, Yisheng Lv, and Fei-Yue Wang. Stop summation: Min-form credit assignment is all process reward model needs for reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Anders Gjølbye, Lars Kai Hansen, and Sanmi Koyejo. Reasoning models don't just think longer, they move differently. *arXiv preprint arXiv:2605.15454*, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Bingxiang He, Zekai Qu, Zeyuan Liu, Yinghao Chen, Yuxin Zuo, Cheng Qian, Kaiyan Zhang, Weize Chen, Chaojun Xiao, Ganqu Cui, et al. Justrl: Scaling a 1.5 b llm with a simple rl recipe. *arXiv preprint arXiv:2512.16649*, 2025.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Andreas Hochlehnert, Hardik Bhatnagar, Vishaal Udandarao, Samuel Albanie, Ameya Prabhu, and Matthias Bethge. A sober look at progress in language model reasoning: Pitfalls and paths to reproducibility. In *Second Conference on Language Modeling*, 2025.

Zhenyu Hou, Ziniu Hu, Yujiang Li, Rui Lu, Jie Tang, and Yuxiao Dong. TreeRL: LLM reinforcement learning with on-policy tree search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025.

Junjie Hu, Gang Tu, Cheng Shengyu, JinXin Li, Jinting Wang, Rui Chen, Zhilong Zhou, and Dongbo Shan. HARP: Hallucination detection via reasoning subspace projection. In *The Fourteenth International Conference on Learning Representations*, 2026.

Hugging Face H4. AIME 2024. Hugging Face Dataset, 2024. URL https://huggingface.co/datasets/HuggingFaceH4/aime_2024.

Xinyan Jiang, Ninghao Liu, Di Wang, and Lijie Hu. Beyond scalars: Evaluating and understanding LLM reasoning via geometric progress and stability. In *Forty-third International Conference on Machine Learning*, 2026.

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

Runze Liu, Jiakang Wang, Yuling Shi, Zhihui Xie, Chenxin An, Kaiyan Zhang, Jian Zhao, Xiaodong Gu, Lei Lin, Wenping Hu, Xiu Li, Fuzheng Zhang, Guorui Zhou, and Kun Gai. Attention as a compass: Efficient exploration for process-supervised RL in reasoning models. In *The Fourteenth International Conference on Learning Representations*, 2026.

Chiyu Ma, Shuo Yang, Kexin Huang, Jinda Lu, Haoming Meng, Shangshang Wang, Bolin Ding, Soroush Vosoughi, Guoyin Wang, and Jingren Zhou. Fipo: Eliciting deep reasoning with future-kl influenced policy optimization. *arXiv preprint arXiv:2603.19835*, 2026.

math-ai. AMC23. Hugging Face Dataset, 2023. URL https://huggingface.co/datasets/math-ai/amc23.

MathArena. AIME 2025. Hugging Face Dataset, 2025. URL https://huggingface.co/datasets/MathArena/aime_2025.

MathArena. AIME 2026. Hugging Face Dataset, 2026. URL https://huggingface.co/datasets/MathArena/aime_2026.

OpenAI. OpenAI o1 System Card. https://openai.com/index/openai-o1-system-card/, 2024.

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. In *The Thirty-sixth Annual Conference on Neural Information Processing Systems*, 2022.

Hiroaki Sakoe and Seibi Chiba. *Dynamic programming algorithm optimization for spoken word recognition*, pp. 159–165. Morgan Kaufmann Publishers Inc., 1990.

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, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Hongze Tan, Zihan Wang, Jianfei Pan, Jinghao Lin, Hao Wang, Yifan Wu, Tao Chen, Zhihang Zheng, Zhihao Tang, and Haihua Yang. GTPO and GRPO-s: Token and sequence-level reward shaping with policy entropy. In *Forty-third International Conference on Machine Learning*, 2026.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *The Thirty-sixth Annual Conference on Neural Information Processing Systems*, 2022.

Guofu Xie, Yunsheng Shi, Hongtao Tian, Ting Yao, and Xiao Zhang. Capo: Towards enhancing llm reasoning through generative credit assignment. *arXiv preprint arXiv:2508.02298*, 2025.

Weichu Xie, Haozhe Zhao, Wenpu Liu, Yongfu Zhu, Liang Chen, Minghao Ye, Zirong Chen, Yuqi Xu, Shuai Dong, Ziyue Wang, Xinbo Xu, Kean Shi, Ruoyu Wu, Xiaoying Zhang, Wenqi Shao, Baobao Chang, Nan Duan, and Jiaqi Wang. Step-wise rubric rewards for llm reasoning. *arXiv preprint arXiv:2605.17291*, 2026.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. Reasoning models know when they're right: Probing hidden states for self-verification. In *Second Conference on Language Modeling*, 2025a.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025b.

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

Jiaru Zou, Ling Yang, Jingwen Gu, Jiahao Qiu, Ke Shen, Jingrui He, and Mengdi Wang. Reasonflux-PRM: Trajectory-aware PRMs for long chain-of-thought reasoning in LLMs. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

CONTENTS

- **Appendix A**: Additional Details on DAPO
- **Appendix B**: Additional Analyses of Divergence
  - **Appendix B.1**: Response-Level Diagnostics on Qwen3-8B-Base
  - **Appendix B.2**: Intuitions and Diagnostics on Llama3.1-8B-Instruct
- **Appendix C**: Additional Experiment Setups

A ADDITIONAL DETAILS ON DAPO

DAPO (Yu et al., 2025) modifies GRPO in four ways to improve training efficiency and stability. First, DAPO adopts a token-level policy gradient loss, averaging directly over all generated tokens rather than first averaging within each response. Consequently, each token receives equal weight in the objective. Second, it adopts dynamic sampling to retain only groups containing both correct and incorrect responses (*i.e.*, $\mathcal{G}^+ \neq \emptyset$ and $\mathcal{G}^- \neq \emptyset$), since groups with identical rewards yield no group-relative learning signal (*i.e.*, $\hat{A}_i = 0$). Third, it uses asymmetric clipping with the interval $[1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}]$, where $\epsilon_{\text{high}} > \epsilon_{\text{low}}$. Finally, it removes the KL regularization term in GRPO by setting $\beta = 0$. The resulting objective is

$$\mathcal{J}_{\text{DAPO}}(\theta) = \mathbb{E}_{\substack{q\sim\mathcal{D}, \\ \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}(\cdot|q)}, \\ \mathcal{G}^+\neq\emptyset,\, \mathcal{G}^-\neq\emptyset}} \left[\frac{1}{\sum_{i=1}^{G}|o_i|} \sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \min\Big(\rho_{i,t}(\theta)\hat{A}_i, \text{clip}\big(\rho_{i,t}(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}\big)\hat{A}_i\Big)\right]. \tag{10}$$

Algorithm 1: Geometry-based Credit Assignment from Hidden-state Trajectories (GAIT)

Input: Old policy $\pi_{\theta_{\text{old}}}$, rollout group $\mathcal{G}$, correct responses $\mathcal{G}^+ \subset \mathcal{G}$, sequence-level advantages $\{\hat{A}_i\}$
Output: Step-level advantages $\{A_{i,t}\}$

```
// Stage 1.1: Extracting projected hidden states.
```
foreach $o_i \in \mathcal{G}$ **do**
- $\{s_{i,t}\}_{t=1}^{T_i} \leftarrow \text{Split}(o_i, \backslash\text{n}\backslash\text{n})$
- $\{h_{i,t}\}_{t=1}^{T_i} \leftarrow \text{HiddenState}(\pi_{\theta_{\text{old}}}, o_i)$
- $\{z_{i,t}\}_{t=1}^{T_i} \leftarrow \text{HARP}(\{h_{i,t}\}_{t=1}^{T_i})$

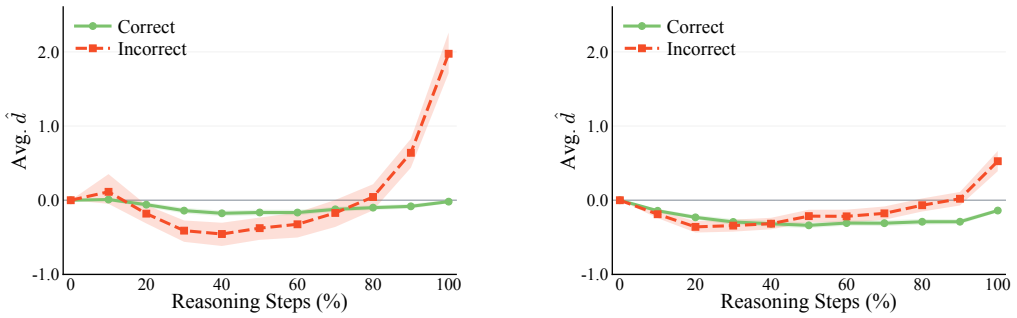
```
// Stage 1.2: Reasoning trajectory alignment.
```
foreach $o_i \in \mathcal{G}$ **do**
- $\mathcal{G}_i^+ \leftarrow \begin{cases} \mathcal{G}^+, & o_i \notin \mathcal{G}^+, \\ \mathcal{G}^+ \setminus \{o_i\}, & o_i \in \mathcal{G}^+ \end{cases}$
- **foreach** $o_j^+ \in \mathcal{G}_i^+$ **do**
  - $P^*(o_i, o_j^+) \leftarrow \text{DTW}(\{z_{i,t}\}_{t=1}^{T_i}, \{z_{j,k}^+\}_{k=1}^{T_j})$
  - Compute $\{a(s_{i,t}, o_j^+)\}_{t=1}^{T_i}$ using Equation (5)

```
// Stage 2: Divergence computation.
```
Compute $d_{i,t}$ for all i, t using Equation (6)
Compute $\hat{d}_{i,t}$ for all i, t using Equation (7)
Compute $\Delta\hat{d}_{i,t}$ for all i, t using Equation (8)

```
// Stage 3: Advantage reweighting.
```
Compute $A_{i,t}$ for all i, t using Equation (9)
Broadcast each $A_{i,t}$ to all tokens in $s_{i,t}$
return $\{A_{i,t}\}$

Table 3: **Response-level diagnostics using** $-\max\{\Delta\hat{d}_{i,t}\}_{t=1}^{T_i}$**.** Results are reported for Qwen3-8B-Base, Llama3.1-8B-Instruct, and their DAPO checkpoints. Selecting the highest-scoring response among 32 candidates (BoN@32) substantially outperforms random selection (Pass@1), while ROC-AUC indicates strong separation between correct and incorrect responses.

Model	Pass@1 (%)	BoN@32 (%)	ROC-AUC
Qwen3-8B-Base			
Base Model	38.80	61.35	0.766
DAPO Checkpoint	67.71	95.99	0.881
Llama3.1-8B-Instruct			
Base Model	33.01	84.85	0.886
DAPO Checkpoint	61.30	89.00	0.862



(a) Llama3.1-8B-Instruct. (b) DAPO checkpoint.

Figure 6: **Average $\hat{d}$ throughout reasoning for correct and incorrect responses from (a) Llama3.1-8B-Instruct and (b) its DAPO checkpoint.** For both models, incorrect trajectories increasingly diverge from successful trajectories, while correct trajectories remain relatively close. After DAPO training, the divergence of incorrect trajectories is substantially reduced.

B Additional Analyses of Divergence

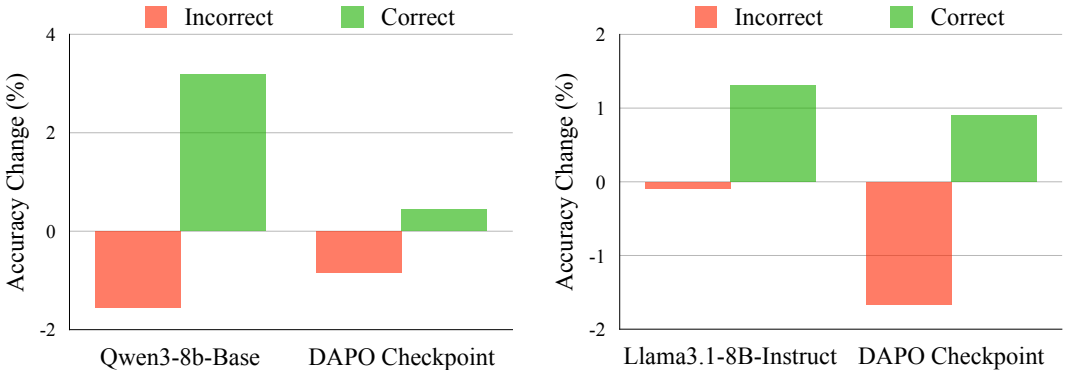
B.1 Response-Level Diagnostics on Qwen3-8B-Base

We assess whether temporal divergence $\Delta\hat{d}_{i,t}$ is informative about response correctness by assigning each response o_i the score $-\max\{\Delta\hat{d}_{i,t}\}_{t=1}^{T_i}$. Higher scores indicate smaller maximum stepwise increases in normalized divergence. For best-of-N (BoN), we select the highest-scoring response among N sampled candidates and report the resulting accuracy. We also report the area under the receiver operating characteristic curve (ROC-AUC) to measure how well these scores distinguish correct from incorrect responses. As shown in Table 3, while sampling 32 responses per prompt, BoN@32 substantially exceeds random-selection accuracy (Pass@1). ROC-AUC increases from 0.766 to 0.881 after DAPO, with both values well above the chance level of 0.5. These results show that temporal divergence is informative about response correctness, with stronger discrimination at the evaluated DAPO checkpoint.

B.2 Intuitions and Diagnostics on Llama3.1-8B-Instruct

We repeat the analyses in Section 4.2.1 and Appendix B.1 using Llama3.1-8B-Instruct and its DAPO checkpoint to examine whether the findings extend to another model family.

Intuition 1: In hidden state space, incorrect responses gradually diverge from correct ones as errors develop. As shown in Figure 6, correct responses maintain relatively low average normalized divergence, whereas incorrect responses exhibit increasing divergence in later reasoning stages. This pattern appears both before and after DAPO, consistent with our first intuition.



(a) Qwen3-8B-Base and its DAPO checkpoint. (b) Llama3.1-8B-Instruct and its DAPO checkpoint.

Figure 7: **Downstream continuation accuracy changes at steps with positive $\Delta\hat{d}_{i,t}$, estimated via MC sampling.** Across both Qwen3-8B-Base and Llama3.1-8B-Instruct, including their DAPO checkpoints, these divergent steps decrease downstream accuracy in incorrect responses but increase it in correct responses.

Table 4: **Pass@32 comparison between our method (GAIT), and baselines.** We report mean ± standard deviation over three RL runs with different random seeds. Best and second-best results are marked in **bold** and underlined, respectively. The green rows denote GAIT. Due to computational constraints, the other two models are evaluated only against DAPO and FIPO, the strongest baseline on Qwen3-8B-Base. Across model scales and families, GAIT achieves the best average Pass@32.

Method	AIME24	AIME25	AIME26	AMC23	BeyondAIME	Avg.
Qwen3-8B-Base						
GRPO	69.63±0.48	47.57±0.56	64.06±2.04	95.61±0.89	38.31±2.03	63.04±0.89
DAPO	**73.61**±2.70	50.99±0.90	64.45±2.06	96.87±1.02	41.16±1.57	65.42±0.51
PURE	50.49±3.31	35.52±2.62	40.38±0.89	91.25±1.56	27.14±0.60	48.96±1.15
ATTNRL	72.25±1.16	52.38±2.01	62.39±2.78	95.31±0.67	**46.65**±1.98	65.80±1.02
FIPO	71.76±0.39	45.69±4.09	65.91±2.67	96.21±1.08	40.82±0.43	64.08±0.83
GAIT	72.88±1.20	**54.18**±0.65	**66.81**±1.33	**97.01**±1.61	40.98±1.24	**66.37**±0.16
Δ vs. DAPO	**-0.73**	**+3.19**	**+2.36**	**+0.14**	**-0.18**	**+0.95**
Qwen3-4B-Base						
DAPO	48.43±2.91	44.21±2.39	38.55±2.45	93.42±0.23	29.33±2.99	50.79±0.57
FIPO	54.40±1.79	**46.66**±0.77	37.17±2.16	94.51±1.54	29.90±0.55	52.53±0.45
GAIT	**54.71**±2.19	45.91±3.46	**42.13**±1.35	**94.96**±0.06	**30.68**±1.69	**53.68**±0.82
Δ vs. DAPO	**+6.28**	**+1.70**	**+3.58**	**+1.54**	**+1.35**	**+2.89**
Llama3.1-8B-Instruct						
DAPO	33.99±1.11	18.68±0.62	17.14±1.63	75.65±3.46	9.97±0.96	31.09±0.77
FIPO	34.13±2.00	17.92±2.95	**17.94**±0.46	**77.83**±1.74	10.47±1.21	31.66±0.59
GAIT	**37.85**±2.52	**18.75**±1.49	16.82±2.30	75.04±2.32	**11.25**±0.69	**31.94**±0.87
Δ vs. DAPO	**+3.86**	**+0.07**	**-0.32**	**-0.61**	**+1.28**	**+0.85**

Intuition 2: In hidden state space, increasing divergence in correct responses may reflect useful alternative reasoning. For steps with $\Delta\hat{d}_{i,t} > 0$, we compare MC-estimated continuation accuracy from the prefixes $\{s_{i,1}, \ldots, s_{i,t-1}\}$ and $\{s_{i,1}, \ldots, s_{i,t}\}$, respectively. As shown in Figure 7b, continuation accuracy decreases on average after these steps in incorrect responses but increases in correct responses, both before and after DAPO. These results support our second intuition that divergent steps in correct responses can reflect useful alternative reasoning paths.

The response-level diagnostic also remains informative on Llama3.1-8B-Instruct. In Table 3, BoN@32 reaches 84.85% for Llama3.1-8B-Instruct and 89.00% for its DAPO checkpoint, substantially exceeding their respective Pass@1 accuracies. The corresponding ROC-AUC values are 0.886 and 0.862, both well above chance. Together, these findings support the two intuitions across both model families and motivate using temporal divergence $\Delta\hat{d}_{i,t}$ for step-level credit assignment.

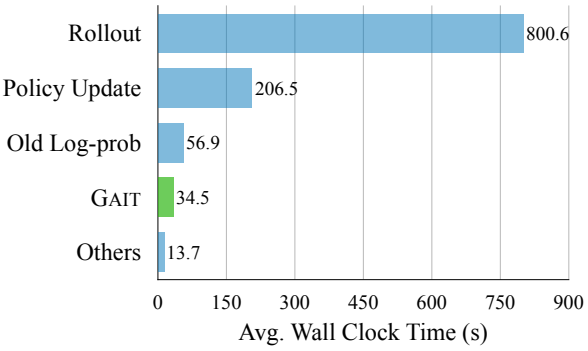


Figure 8: **Average runtime breakdown per RL step.** Using Qwen3-8B-Base as the base model, GAIT introduces negligible computational overhead, accounting for only 3.1% of the wall-clock time per RL step.

Table 5: **Ablation of HARP projection.** Without projection, credit assignment operates directly on the original hidden states. Using HARP in GAIT improves accuracy on all three AIME benchmarks.

HARP Projection	AIME24	AIME25	AIME26	Avg.
✗	31.98	26.35	25.73	28.02
✓	**32.43**	**27.50**	**26.56**	**28.83**

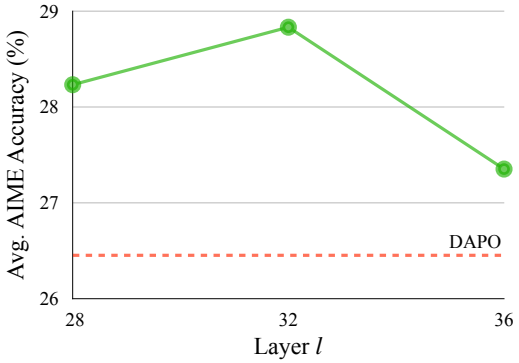


Figure 9: **Ablation of the layer l for extracting hidden states.** GAIT outperforms DAPO (red dashed line) at all tested layers, with the highest accuracy at $l = 32$.

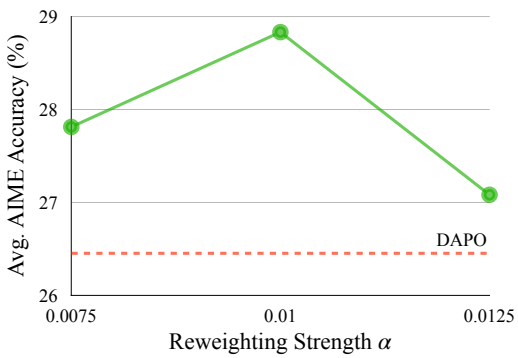


Figure 10: **Ablation of the advantage reweighting strength α in Equation (9).** GAIT outperforms DAPO (red dashed line) for all tested values, with the highest accuracy at $\alpha = 0.01$.

C ADDITIONAL EXPERIMENT SETUPS

Baseline configurations. Following JUSTRL (He et al., 2025), our GRPO configuration uses the "clip higher" strategy with an asymmetric clipping range of $[0.8, 1.28]$ and the same token-level policy gradient loss as DAPO. For DAPO, following the findings of JUSTRL, we remove the overlong penalty. This modification improves DAPO's performance in our experiments.

Model-specific hyperparameters. For Qwen3-4B-Base, we extract hidden states from layer $l = 32$, corresponding to approximately 90% of the model depth, and set $\alpha = 0.0075$. For Llama3.1-8B-Instruct, we use layer $l = 28$, also corresponding to approximately 90% of the model depth, and set $\alpha = 0.015$. All other hyperparameters and training settings are identical to those used for Qwen3-8B-Base, including $\eta_{\text{low}} = 0$, $\eta_{\text{high}} = 0.2$, and the requirement that GAIT is applied only to rollout groups with at least four correct responses ($|\mathcal{G}^+| \geq 4$). For the remaining groups, we use the original DAPO advantages.