RIGHT MAKES MIGHT: ALIGNING VERIFIED HIDDEN STATES EMPOWERS RL REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning from Verifiable Rewards (RLVR) has become the dominant approach for improving mathematical reasoning in large language models, yet current methods reduce each correct rollout to a single reward bit, ignoring the geometric structure shared among their hidden states. Investigating this structure, we find that at the *anchor token* (the position immediately before the answer marker), correct rollouts converge naturally because they must produce the same answer (cosine similarity ≈ 0.84), yet each retains residual variance from its unique reasoning path. Encouraging full alignment at this point pushes the model to extract a unified "correct decision" representation, reducing sensitivity to which reasoning path was taken.

Based on this observation, we propose Hidden-Align, an auxiliary loss function that aligns the last-layer hidden states of correct rollouts at the anchor token during RL training, without appreciable training overhead or additional inference cost. On eight mathematical reasoning benchmarks, Hidden-Align improves average pass@1 over the DAPO baseline by 3.8, 6.2, and 5.4 percentage points on Qwen3-1.7B, 4B, and 14B respectively, with consistent pass@k gains across all three scales. These benefits generalize across RL algorithms and model families, extending beyond mathematics to spatial reasoning. Systematic ablations support the design choices.

1 INTRODUCTION

Reinforcement Learning from Verifiable Rewards (RLVR) has become the dominant approach for mathematical reasoning in large language models (Guo et al., 2025a; Shao et al., 2024; Yu et al., 2026; Yue et al., 2025; He et al., 2025). Methods such as DAPO and GRPO generate groups of rollouts per prompt, score each with a binary correctness reward, and update the policy via group-relative advantages. This pipeline achieves strong gains on competition-level math benchmarks, yet it reduces every correct rollout to a single reward bit. When multiple rollouts in a group solve the same problem correctly, the geometric relationships among their internal representations are discarded. While regularization techniques such as KL penalties (Ouyang et al., 2022) exist, they operate on the output distribution. While recent work has used hidden states for reward prediction (Guo et al., 2025b) and reward model regularization (Yang et al., 2024), no existing method aligns correct rollouts' hidden states as a training signal during RL.

We investigate whether this structure can be exploited as an auxiliary training signal by analyzing the hidden states of RL-trained models during reasoning. We begin by probing the pairwise cosine similarity of correct rollouts' hidden states across token positions (Belinkov, 2022; Zou et al., 2023) (Figure 1). The results show a clear positional structure: in the reasoning chain, hidden states are dispersed as each rollout follows a different path, reflecting the diverse reasoning strategies cultivated by RL training. Near the answer tokens, they converge sharply (> 0.9) because all correct rollouts produce the same answer. Past the answer, similarity drops again. Between these regions sits the *anchor token*, the position immediately before the answer marker `\boxed{`. Here, hidden states are compact (cosine similarity ≈ 0.84) yet not fully converged, occupying a unique sweet spot.

We also notice that the last layer's curve stands out from shallower layers, exhibiting far stronger position-dependent variation in Figure 1(a). Repeating the analysis on the last layer across three model scales of Qwen3 (Yang et al., 2025), we find the same pattern consistently in Figure 1(b). Inspired by the effectiveness of representation alignment in other domains (Yu et al., 2024), we

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.

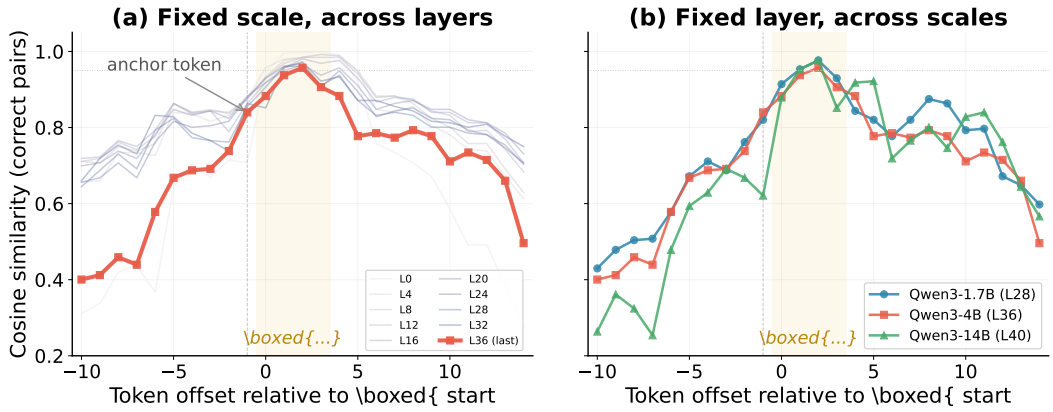


Figure 1: Pairwise cosine similarity of correct rollouts' hidden states, plotted against token position relative to \boxed{. **(a)** Across all layers of Qwen3-4B, the last layer (L36, red) exhibits uniquely strong position-dependent variation, with similarity rising toward the anchor token (offset $= -1$); shallower layers show weaker variation. **(b)** At the last layer across three model scales (1.7B, 4B, 14B), the same pattern holds consistently: the anchor-token hidden states form a compact but not fully converged cluster ($\cos \approx 0.84$).

hypothesize that actively aligning these hidden states, encouraging their cosine similarity toward full agreement, would push the model to distill the common structure shared by diverse correct reasoning paths into a robust decision representation.

We propose Hidden-Align, an auxiliary loss that maximizes the pairwise cosine similarity of correct rollouts' last-layer hidden states at the anchor token during RL training. Since the hidden states are already produced during training, this adds no appreciable training overhead or additional inference cost (Appendix B). We systematically ablate each design choice (loss type, anchor position, layer depth, and loss weight), supporting the proposed design choices.

In summary, we make the following contributions:

- We identify and quantify a consistent geometric phenomenon in RL-trained reasoning models: at the anchor token, correct rollouts' last-layer hidden states converge naturally but incompletely, and this pattern holds consistently across model scales.
- We propose Hidden-Align, an auxiliary loss for RL training that aligns correct rollouts' hidden states at the anchor token (§3).
- On eight math benchmarks across Qwen3-1.7B, 4B, and 14B, Hidden-Align improves both pass@1 and pass@k over the DAPO baseline, with no appreciable training overhead or additional inference cost. These benefits generalize across RL algorithms and model families, extending beyond mathematics to spatial reasoning. Systematic ablations on loss type, position, layer, and weight support the proposed design choices (§4).

2 Related Work

2.1 Reinforcement Learning for LLM Reasoning

RLVR has become the standard approach for improving mathematical reasoning in LLMs. DeepSeek-R1 (Guo et al., 2025a) demonstrated that pure RL training can induce emergent chain-of-thought reasoning, leading to broad adoption of group-relative optimization methods. Among these, GRPO (Shao et al., 2024) computes advantages within each rollout group, eliminating the need for a separate value model, and DAPO (Yu et al., 2026) further refines it with clip-higher, dynamic sampling, and token-level loss rebalancing. A growing family of variants further improves the policy optimization objective: Dr. GRPO (Liu et al., 2025) corrects normalization biases, VAPO (Yue et al., 2025) stabilizes value estimation, and others address advantage shaping (Zheng et al., 2025; Zhao et al., 2025; Shrivastava et al., 2025; Simoni et al., 2025), negative-sample utilization (Nan et al., 2025; Ding et al., 2025), KL penalty design (Hong et al., 2025), intrinsic exploration (Sun et al., 2025),

reward granularity (Xie et al., 2026), error diversity (Liu et al., 2026), and rank bias (He et al., 2025). All these methods operate at the token-probability level, optimizing the policy gradient, advantage function, or sampling strategy. Hidden-Align is orthogonal: it introduces a training signal in the hidden representation space, and can be combined with any of the above algorithms. We empirically demonstrate this compatibility with both DAPO and GSPO.

2.2 Representation Alignment as a Training Signal

Aligning intermediate representations with a reference target is effective in several domains. REPA (Yu et al., 2024) aligns diffusion transformer hidden states to DINOv2 features via cosine similarity, achieving a 17.5× training speedup and state-of-the-art generation quality. Earlier, contrastive and relational distillation methods (Tian et al., 2019; Park et al., 2019) established that aligning representational structure, rather than matching output distributions, transfers knowledge more effectively. In the speech domain, Wang et al. (2026a) use a representation alignment reward during RL training to close the reasoning gap between text and speech modalities, demonstrating that hidden-state similarity can serve directly as a reward signal. These works establish that representation alignment is a broadly effective training principle; we introduce correct-rollout representation alignment to RL-based reasoning for the first time, aligning hidden states at the anchor token rather than distilling from an external model.

2.3 Hidden States in Reinforcement Learning

Several recent works use LLM hidden states within the RL pipeline, though none perform alignment among correct rollouts. Guo et al. (2025b) show that a simple linear probe on hidden states can predict reward nearly as well as a full reward model, confirming that hidden states encode correctness information. Yang et al. (2024) regularize the reward model's hidden states to prevent over-optimization, improving generalization. Sun et al. (2026) reveal step-specific geometry in reasoning trajectories but do not intervene on these representations. CRAFT (Luo et al., 2026) uses contrastive learning to separate safe from unsafe reasoning trajectories, targeting safety rather than reasoning performance. Representation-based intrinsic rewards in traditional RL (Raileanu & Rocktäschel, 2020; Gupta et al., 2022; Wang et al., 2024) are exploration-oriented, whereas Hidden-Align is attraction-oriented: it rewards similarity among correct rollouts at a specific position, consolidating diverse reasoning paths into a unified decision representation.

3 Methodology

Our method adds a single auxiliary loss to standard RLVR training. Given a group of rollouts generated by the policy, we extract the last-layer hidden state at the *anchor token* from each correct rollout and maximize their pairwise cosine similarity. Figure 2 illustrates the combined objective $\mathcal{L} = \mathcal{L}_{\text{DAPO}} + \lambda \cdot \mathcal{L}_{\cos}$ and the overall training pipeline. We use DAPO (Yu et al., 2026) as our primary RL algorithm. Below, we describe the base algorithm (§3.1), the alignment loss (§3.2), the gradient derivation (§3.3), the rationale for our design choices (§3.4), and integration with the RL training pipeline (§3.5).

3.1 Preliminary: Group-Relative Policy Optimization

We build upon DAPO (Yu et al., 2026), a representative group-relative RLVR method. For each training prompt q, the policy π_θ generates a group of n responses $\{y_1, \ldots, y_n\}$. Each response y_i ($i = 1, \ldots, n$) receives a binary reward $r_i \in \{0, 1\}$ from a verifier. The group-relative advantage is:

$$A_i = \frac{r_i - \mu_r}{\sigma_r + \epsilon} \tag{1}$$

where μ_r and σ_r are the mean and standard deviation of rewards within the group, and ϵ is a small constant for numerical stability. The policy is updated via a token-averaged clipped surrogate loss:

$$\mathcal{L}_{\text{DAPO}} = -\mathbb{E}\left[\min\left(\rho_{i,t}A_i,\ \text{clip}(\rho_{i,t}, 1{-}\epsilon_l, 1{+}\epsilon_h)A_i\right)\right] \tag{2}$$

where $\rho_{i,t} = \pi_\theta(y_{i,t} \mid q, y_{i,<t}) / \pi_{\text{old}}(y_{i,t} \mid q, y_{i,<t})$ is the importance ratio at token t, conditioned on the prompt and preceding tokens. The expectation averages over valid response tokens, π_{old} is the policy before the current update, and ϵ_l, ϵ_h are the lower and upper clip bounds defined in DAPO.

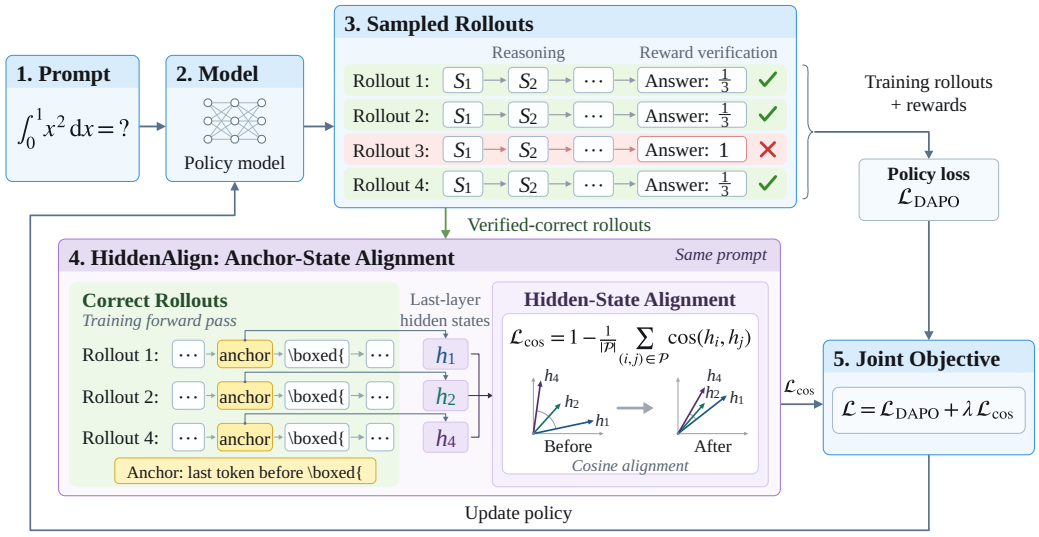


Figure 2: Overview of Hidden-Align. **Steps 1–3**: A prompt is fed to the LLM, which generates multiple rollouts; correct sequences are identified by reward verification. **Step 4**: From correct sequences, we extract last-layer hidden states at the anchor token (immediately before `\boxed{`) and align their directions using a pairwise cosine loss. Here, $\mathcal{P}$ denotes all distinct pairs of correct rollouts for the same prompt. **Step 5**: The combined objective $\mathcal{L} = \mathcal{L}_{\text{DAPO}} + \lambda \cdot \mathcal{L}_{\cos}$ is used for gradient update.

3.2 Hidden-Align Loss

Anchor Token Definition. For each response y_i in the group, we select the anchor token according to the task's output format. In our main mathematical experiments, we locate the first occurrence of the `\boxed{` token sequence in the generated output. The *anchor token* is defined as the token immediately preceding this marker, i.e., the point where the model has arrived at its answer but has not yet begun writing it. We extract the hidden state $\mathbf{h}_i$ at this position from the last transformer layer, i.e., the final representation before the LM head projects to vocabulary logits. As shown in Figure 1, the anchor token occupies a middle ground in cosine similarity: higher than the reasoning chain ($\sim$0.4) because correct rollouts are converging toward the same answer, but lower than the answer tokens themselves (> 0.9), where similarity is too high for any auxiliary loss to provide meaningful gradients.

Alignment Loss. For each prompt group, let $\mathcal{C}$ be the set of correct responses with a valid anchor token, and let $\mathcal{P} = \{(i,j) : i,j \in \mathcal{C}, i < j\}$ be all correct-answer pairs. We compute the average pairwise cosine similarity among correct rollouts' hidden states:

$$\mathcal{L}_{\cos} = 1 - \frac{1}{|\mathcal{P}|} \sum_{(i,j)\in\mathcal{P}} \cos(\mathbf{h}_i, \mathbf{h}_j) \tag{3}$$

where $\cos(\mathbf{h}_i, \mathbf{h}_j) = \hat{\mathbf{h}}_i^\top \hat{\mathbf{h}}_j$ with $\hat{\mathbf{h}}_i = \mathbf{h}_i / \|\mathbf{h}_i\|$. The L2 normalization affects the gradient computation, which we derive in §3.3. The loss is only computed when $|\mathcal{C}| \geq 2$; groups with fewer than two correct responses are skipped.

The total training loss combines the standard policy gradient with the alignment term:

$$\mathcal{L} = \mathcal{L}_{\text{DAPO}} + \lambda \cdot \mathcal{L}_{\cos}, \quad \lambda = 0.001 \tag{4}$$

Design Choices. We consider five candidate loss types defined on the anchor token hidden states within each prompt group:

- **Alignment**: *pull together* correct rollouts (maximize pairwise cosine similarity).

- **Separation**: *push apart* correct and incorrect rollouts (minimize cross-group cosine similarity).
- **Negative**: *push apart* incorrect rollouts from each other (minimize pairwise cosine similarity among incorrect).
- **Diversity**: *push apart* all rollouts regardless of correctness (minimize global pairwise cosine similarity).
- **Cluster**: *pull together* all rollouts regardless of correctness (maximize global pairwise cosine similarity).

We adopt alignment as our primary loss. Separation is likely redundant with the DAPO advantage; diversity and negative losses oppose the natural convergence; cluster conflates correct and incorrect rollouts. We also evaluate combinations of these losses. The small alignment weight $\lambda = 0.001$ balances the different scales of the alignment and RL losses; a stronger signal would interfere with the primary RL objective. We verify all these choices in §4.3.

3.3 Gradient Derivation

Gradient of the Alignment Loss. The gradient of $\mathcal{L}_{\cos}$ (Eq. 3) with respect to the raw hidden state $\mathbf{h}_i$ of a correct sample i in group g is:

$$\frac{\partial \mathcal{L}_{\cos}}{\partial \mathbf{h}_i} = -\frac{1}{|\mathcal{P}|} \sum_{\substack{j \in \mathcal{C}_g \\ j \neq i}} \frac{1}{\|\mathbf{h}_i\|} \left(\hat{\mathbf{h}}_j - s_{ij}\, \hat{\mathbf{h}}_i \right) \tag{5}$$

where $\mathcal{C}_g$ is the set of correct responses in group g and $s_{ij} = \hat{\mathbf{h}}_i^\top \hat{\mathbf{h}}_j$. The full derivation is provided in Appendix B.1. Intuitively, the gradient pushes $\hat{\mathbf{h}}_i$ toward the other correct hidden states $\hat{\mathbf{h}}_j$ in its group, with a projection term $s_{ij}\hat{\mathbf{h}}_i$ that removes the component along $\hat{\mathbf{h}}_i$.

3.4 Why It Works

Correct rollouts already cluster more tightly than correct–incorrect pairs at the anchor token, suggesting that the alignment loss reinforces an existing trend rather than imposing an artificial constraint. However, this clustering remains incomplete: correct rollouts arrive at the same answer via different reasoning paths, and their hidden states encode a mixture of path-specific details and a shared "correct decision" signal (see Appendix A for the full distribution).

Hidden-Align concentrates alignment at the answer boundary, strengthening the shared decision structure of correct rollouts while allowing their intermediate reasoning trajectories to differ. We encourage the model to factor out path-specific variance at the point of decision, distilling the common structure that leads to a correct answer.

This offers an explanation for the improvements in both pass@1 and pass@k shown in Figure 3. A more consistent decision representation makes the model more likely to produce a correct answer in a single attempt (pass@1). Moreover, the process of distilling common structure across diverse paths deepens the model's understanding of what constitutes a correct solution, improving its ability to find correct answers even through novel reasoning paths (pass@k).

3.5 Integration with RL Training Pipeline

Challenge: Micro-Batch Training. Computing Eq. 5 requires access to all correct embeddings $\hat{\mathbf{h}}_j$ from the same group (i.e., rollouts generated from the same prompt). In standard RL training, each optimization step processes a large mini-batch of rollouts, which is further divided into micro-batches to fit GPU memory (see Appendix B for details). This structure is essential for stable gradient estimation and training efficiency. However, samples from the same group may be split across different micro-batches, complicating the computation of the alignment loss across a complete group. Gathering all group members into memory simultaneously would cause out-of-memory failures under typical training configurations, while computing the loss on incomplete groups would yield approximate gradient estimates.

Two-Phase Solution. We decompose the backward pass into two phases:

Table 1: Main results on mathematical reasoning benchmarks. Detailed evaluation settings are provided in Appendix B.

Method	Size	AIME24	AIME25	AIME26	AMC23	AMC24	HMMT	Minerva	Olympiad	Avg
Qwen3-4B (base)		24.79	21.98	17.29	69.22	45.35	11.25	56.25	54.15	37.53
+ DAPO	4B	48.12	43.44	40.31	88.28	58.75	19.79	53.68	63.35	51.97
+ DAPO + **Hidden-Align**		**56.67**	**49.90**	**50.52**	**90.62**	**63.40**	**27.81**	**59.19**	**67.36**	**58.18**
+ GSPO		45.63	44.79	39.48	88.52	57.29	21.35	57.35	64.39	52.35
+ GSPO + **Hidden-Align**		**47.29**	**46.67**	**43.65**	88.13	**58.47**	20.52	**58.82**	63.95	**53.44**
Qwen3-1.7B (base)		14.27	10.42	8.33	46.56	27.99	4.69	36.76	41.54	23.82
+ DAPO	1.7B	29.06	25.83	19.90	68.12	47.43	14.06	44.85	49.70	37.37
+ DAPO + **Hidden-Align**		**31.87**	**27.50**	**24.27**	**73.83**	**47.57**	**16.98**	**52.21**	**55.34**	**41.20**
Qwen3-14B (base)		29.48	25.52	17.60	72.97	49.58	11.25	61.03	58.61	40.76
+ DAPO	14B	60.94	51.98	49.79	92.27	64.10	31.35	58.82	72.85	60.26
+ DAPO + **Hidden-Align**		**68.12**	**61.67**	**60.94**	**93.98**	**65.62**	**34.58**	**64.71**	**75.82**	**65.68**

Phase 1 (Gather): We forward-pass all micro-batches sequentially with gradient computation disabled, which avoids storing the computation graph and significantly reduces memory usage. From each forward pass, we collect the anchor embedding, group identifier, and correctness label for every sample. These embeddings are L2-normalized and detached from the computation graph to serve as fixed references in Phase 2.

Phase 2 (Backward): For each micro-batch chunk, we extract the correct samples' hidden states during the policy-update forward pass to obtain gradient-attached embeddings $\hat{\mathbf{h}}_i$. For each anchor i, we compute the loss against the detached references from the same group:

$$\mathcal{L}_{\cos}^{(i)} = -\frac{1}{|\mathcal{P}|} \sum_{\substack{j \in \mathcal{C}_g \\ j \neq i}} \hat{\mathbf{h}}_i^\top \bar{\hat{\mathbf{h}}}_j \tag{6}$$

where $\bar{\hat{\mathbf{h}}}_j$ are detached (gradients flow only through $\hat{\mathbf{h}}_i$, not through $\bar{\hat{\mathbf{h}}}_j$). The total loss $\mathcal{L}_{\cos} = \sum_i \mathcal{L}_{\cos}^{(i)}$ yields the exact same gradients as Eq. 5: although each term only backpropagates through one anchor, every pair (i, j) is visited twice (once with i as the anchor and once with j), so both sides receive the correct gradient.

This decomposition achieves exact gradient computation with bounded memory usage, and requires no modification to the standard mini-batch/micro-batch training structure.

Group-Preserving Batch Reordering. In data-parallel training, rollouts are distributed across GPUs. If samples from the same group are assigned to different GPUs, the alignment loss cannot access the complete set of correct pairs within that group. We address this by reordering the training batch so that all rollouts sharing the same prompt are co-located on a single GPU. To prevent memory imbalance, we additionally ensure that each GPU receives a similar total workload during reordering.

4 Experiments

4.1 Experimental Setup

Models. We use Qwen3-4B as the primary model for main results and all ablation experiments, and Qwen3-1.7B and Qwen3-14B for scale verification. We apply Hidden-Align on top of DAPO training.

Training Data. We use DAPO-Math-17K (Yu et al., 2026), a standard RLVR training set of 17,398 mathematical problems covering competition math, algebra, geometry, number theory, and combinatorics. All problems have answers enclosed in `\boxed{}`, enabling automated reward verification.

Benchmarks. We evaluate on eight mathematical reasoning benchmarks spanning competition math and general math: AIME 2024/2025/2026 (Mathematical Association of America, 2024a), AMC

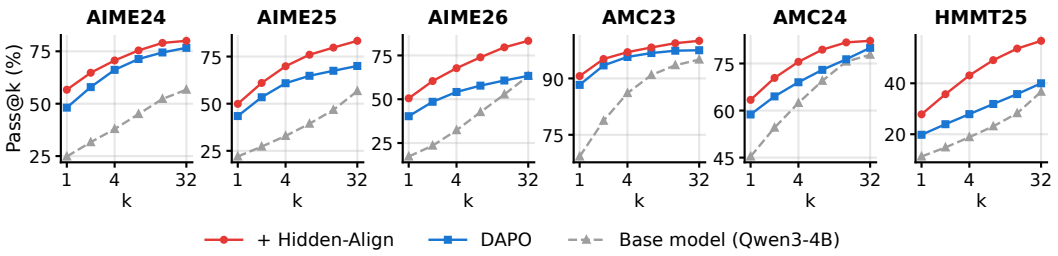


Figure 3: Pass@k curves on six competition benchmarks for Qwen3-4B (temperature 0.2, n=32). Hidden-Align (red) outperforms DAPO (blue) and the base model (gray dashed) across all tested values of k.

2023/2024 (Mathematical Association of America, 2024b), HMMT Feb 2025 (Harvard-MIT Mathematics Tournament, 2025) (competition-level, 30–45 questions each); Minerva Math (Lewkowycz et al., 2022) (general math), and OlympiadBench (He et al., 2024) (olympiad-level).

Metrics. We report accuracy on all benchmarks following standard RLVR evaluation protocols. Detailed evaluation settings (sampling parameters, number of trials) are provided in Appendix B.

4.2 Main Results

Table 1 presents our main results. With DAPO, Hidden-Align improves average accuracy by **6.22** percentage points (95% CI: [4.48, 7.96]; Appendix C.2) on Qwen3-4B, with consistent gains on all eight benchmarks at each of the three model scales. On Qwen3-4B, large gains include AIME 2024 (+8.54pp), AIME 2026 (+10.21pp), and HMMT (+8.02pp). On Minerva, Hidden-Align also reverses the accuracy drops observed under DAPO on Qwen3-4B and Qwen3-14B, exceeding the corresponding base-model scores. With GSPO (Zheng et al., 2025), Hidden-Align also improves average accuracy by **1.09** percentage points on Qwen3-4B.

Generalization. On Llama-3.2-3B-Instruct (Meta, 2024), Hidden-Align improves average accuracy over four mathematical reasoning benchmarks by **1.50** percentage points over DAPO (Table 2; full results in Appendix D.1). On Qwen3-VL-30B-A3B-Instruct (Bai et al., 2025), Hidden-Align improves the average score over four spatial reasoning benchmarks by **3.04** percentage points over DAPO (Table 3), extending its benefits beyond mathematical reasoning with an anchor immediately before `</think>` (Appendix D.2). In mathematical reasoning, Hidden-Align also improves average accuracy with a natural-language answer marker (Appendix D.3).

Robustness across Training Seeds. Across three training seeds on Qwen3-4B, Hidden-Align improves average accuracy from $\mathbf{52.33 \pm 1.00}$ to $\mathbf{55.80 \pm 2.08}$ (mean $\pm$ sample standard deviation), a gain of **3.47** percentage points. It outperforms DAPO on all three seeds. Per-seed results are provided in Appendix C.1.

Pass@k Analysis. Beyond pass@1, we evaluate Pass@k to measure the model's coverage of correct solutions across multiple attempts. We sample $n = 32$ responses per prompt at temperature 0.2 and compute Pass@k using the unbiased estimator from (Chen et al., 2021). As shown in Figure 3, Hidden-Align achieves higher Pass@k across all tested values of k on all six competition benchmarks. The gap is especially pronounced on harder benchmarks (AIME 2025/2026, HMMT). Results for the other model scales are provided in Appendix K.

4.3 Ablation Studies

We ablate each design choice of Hidden-Align to verify its necessity. All ablations are conducted on Qwen3-4B with DAPO training.

Table 2: Generalization to Llama-3.2-3B-Instruct.

Method	AIME24	AMC24	Math500	Minerva	Avg.
Base	6.77	8.47	40.60	15.07	17.73
DAPO	7.92	13.68	43.20	23.16	21.99
+ HA	**10.31**	13.33	**49.00**	21.32	**23.49**

Table 3: Spatial reasoning with Qwen3-VL-30B-A3B-Instruct.

Method	EmbSp.	RoboSp.	RefSp.	TreeBench	Avg.
Base	77.90	60.00	52.00	**43.70**	58.40
DAPO	**83.79**	62.86	50.54	40.25	59.36
DAPO + HA	83.41	**65.71**	**57.76**	42.72	**62.40**

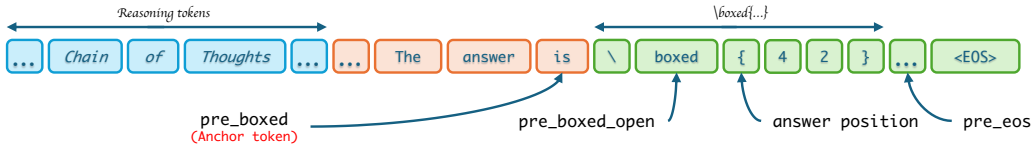


Figure 4: Token positions evaluated in the position ablation. The anchor token (pre_boxed) is immediately before the \boxed{ marker.

Table 4: Loss type ablation (all at anchor token, last layer, $\lambda = 0.001$). Alignment alone performs best.

Loss Type	AIME24	AIME26	AMC24	HMMT	Avg
DAPO (baseline)	48.12	40.31	58.75	19.79	41.74
Alignment (ours)	**56.67**	**50.52**	**63.40**	**27.81**	**49.60**
Separation	46.35	41.88	57.36	19.17	41.19
Diversity	49.69	45.21	61.94	24.17	45.25
Negative	49.06	45.83	58.40	24.38	44.42
Cluster	46.46	40.73	60.14	23.12	42.61
Align. + Sep.	45.62	41.25	59.93	22.40	42.30
Align. + Neg.	53.33	49.17	60.69	24.90	47.02
Align. + Div.	44.48	38.02	59.24	17.92	39.92

Table 5: Position ablation (alignment loss, last layer, $\lambda = 0.001$).

Anchor Position	AIME24	AIME26	AMC24	HMMT	Avg
DAPO (baseline)	48.12	40.31	58.75	19.79	41.74
pre_boxed (ours)	**56.67**	**50.52**	**63.40**	**27.81**	**49.60**
pre_eos	43.65	41.04	57.92	21.35	40.99
pre_boxed_open	48.02	43.33	61.32	23.02	43.92
answer position	48.12	42.71	58.82	22.19	42.96

Loss Type. We compare the five loss types defined in §3.2. Since correct rollouts already cluster at the anchor token, a pull-together signal (alignment) reinforces this existing trend. Table 4 confirms that alignment alone performs best, while combinations with other losses dilute the gain.

Anchor Position. We apply the alignment loss at different token positions relative to \boxed{. Figure 4 illustrates the four candidate positions. As discussed in §3.2, the anchor token occupies a unique sweet spot in cosine similarity; other positions are either too converged (answer tokens) or too dispersed (reasoning chain). Table 5 confirms that only the anchor token yields substantial improvement.

Layer Depth and Loss Weight. We further examine the transformer layer and alignment loss weight. The last layer yields the largest gain, while intermediate layers provide smaller improvements. A small alignment weight balances the different scales of the alignment and RL losses: $\lambda = 0.001$ performs best, whereas $\lambda = 0.01$ causes training divergence. Detailed results are provided in Appendix E.

4.4 Analysis

Alignment Loss and Reward. Figure 5 compares the alignment loss $\mathcal{L}_{\cos}$ and accuracy reward between Hidden-Align and the DAPO baseline over the course of training. In (a), the alignment loss for Hidden-Align starts high and gradually decreases, indicating that correct rollouts' hidden states are being actively aligned at the anchor token. For comparison, we also monitor the same metric on the DAPO baseline (which does not optimize it): the baseline shows a similar but weaker downward trend in early training, suggesting that RL training alone induces partial alignment. Moreover, around step 400–500, the baseline's alignment loss spikes sharply, indicating that the natural clustering breaks down in later training stages. In contrast, Hidden-Align maintains stable alignment throughout.

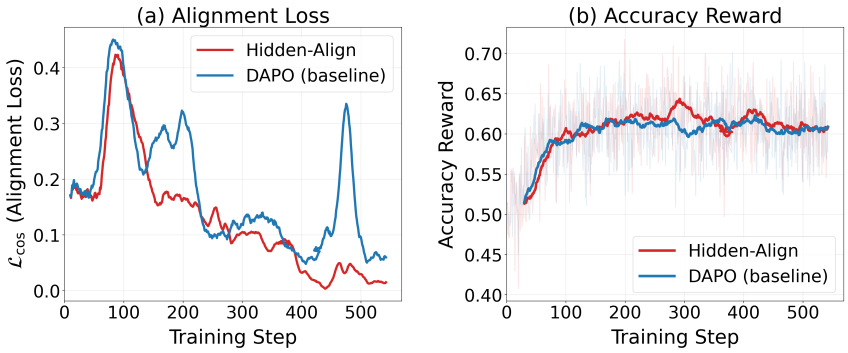


Figure 5: (a) Alignment loss $\mathcal{L}_{\cos}$ and (b) accuracy reward over training steps for Qwen3-4B. Hidden-Align (red) shows active alignment loss reduction; the baseline's alignment loss (monitored but not optimized) spikes in later stages. In (b), faint lines show raw per-step values and bold lines show smoothed trends. Hidden-Align generally achieves higher reward.

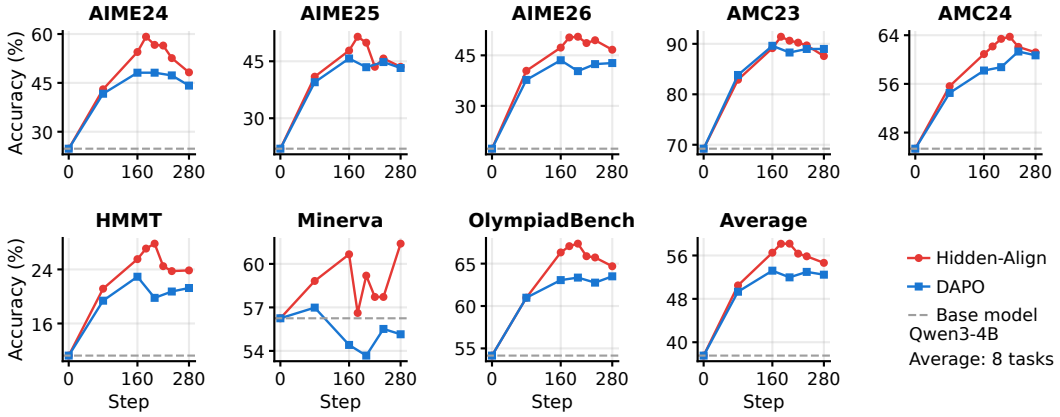


Figure 6: Per-benchmark accuracy over training steps for Qwen3-4B. Hidden-Align (red) outperforms DAPO (blue) on most benchmarks throughout training, with the largest gains on competition-level benchmarks (AIME, HMMT). The gray dashed line indicates the base model (before RL training). Average denotes the unweighted mean over the eight benchmarks in Table 1. Both methods peak around step 160–200 and gradually decline afterward.

In (b), Hidden-Align generally achieves higher accuracy reward than the baseline, particularly during steps 200–300. The overall trend confirms that aligning correct rollouts' hidden states provides a beneficial training signal.

Accuracy over Training Steps. Figure 6 shows per-benchmark accuracy at multiple training checkpoints. Hidden-Align consistently outperforms DAPO on competition-level benchmarks (AIME 2024/2025/2026, HMMT), with the gap emerging as early as step 80 and widening through step 160–200. On easier benchmarks (AMC 2023/2024), both methods perform similarly, as they are closer to the ceiling. Both methods peak around step 160–200 and gradually decline afterward.

5 Conclusion

We present Hidden-Align, a lightweight auxiliary loss that aligns correct rollouts' hidden states at the anchor token during RL training. It improves mathematical reasoning across three model scales, with consistent pass@k gains and no appreciable training overhead or additional inference cost. These benefits generalize across RL algorithms and model families, extending beyond mathematics to spatial reasoning. Systematic ablations support the design choices.

Ethical Considerations

This work improves mathematical reasoning in LLMs through a representation-level auxiliary loss applied during RL training. It does not involve human subjects, private data, or content generation in sensitive domains. The training data (DAPO-Math-17K) consists of publicly available mathematical problems. We do not foresee specific ethical risks beyond those common to general-purpose LLM reasoning research.

AI Use Statement

We used generative AI tools to assist with writing and language polishing. The authors reviewed and revised the AI-assisted text and take full responsibility for the final manuscript.

Reproducibility Statement

Training and evaluation details are provided in Appendix B. The supplementary material includes code, configurations, and analysis data for the mathematical reasoning experiments.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025. URL https://arxiv.org/abs/2511.21631.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Yuyang Ding, Chi Zhang, Juntao Li, Haibin Lin, and Min Zhang. Fapo: flawed-aware policy optimization for efficient and reliable reasoning. *arXiv preprint arXiv:2510.22543*, 2025.

Mengfei Du, Binhao Wu, Zejun Li, Xuanjing Huang, and Zhongyu Wei. EmbSpatial-Bench: Benchmarking spatial understanding for embodied tasks with large vision-language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 346–355, 2024. doi: 10.18653/v1/2024.acl-short.33. URL https://aclanthology.org/2024.acl-short.33/.

B. Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1):1–26, 1979. doi: 10.1214/aos/1176344552.

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*, 2025a.

Jizhou Guo, Zhaomin Wu, and S Yu Philip. Reward inside the model: A lightweight hidden-state reward model for llm's best-of-n sampling. In *2nd AI for Math Workshop@ ICML 2025*, 2025b.

Tarun Gupta, Peter Karkus, Tong Che, Danfei Xu, and Marco Pavone. Foundation models for semantic novelty in reinforcement learning. *arXiv preprint arXiv:2211.04878*, 2022.

Harvard-MIT Mathematics Tournament. HMMT february competition. https://www.hmmt.org/, 2025.

Andre Wang He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting grpo beyond distribution sharpening. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25559–25571, 2025.

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.

Minjie Hong, Zirun Guo, Yan Xia, Zehan Wang, Ziang Zhang, Tao Jin, and Zhou Zhao. Apo: Enhancing reasoning ability of mllms via asymmetric policy optimization. *arXiv preprint arXiv:2506.21655*, 2025.

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.

Shijie Lian, Changti Wu, Laurence Tianruo Yang, Hang Yuan, Bin Yu, Lei Zhang, and Kai Chen. Euclid's gift: Enhancing spatial perception and reasoning in vision-language models via geometric surrogate tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Findings*, pp. 9824–9835, June 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Wenpu Liu, Yuqi Xu, Weichu Xie, Yongfu Zhu, Shuai Dong, Ziyue Wang, Wenqi Shao, Xiaoying Zhang, Tong Yang, Nan Duan, et al. Leveraging error diversity in group rollouts for reinforcement learning. *arXiv preprint arXiv:2605.17333*, 2026.

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.

Haozheng Luo, Yimin Wang, Jiahao Yu, Binghui Wang, and Yan Chen. Contrastive reasoning alignment: Reinforcement learning from hidden representations. *arXiv preprint arXiv:2603.17305*, 2026.

Mathematical Association of America. AIME problems and solutions. https://maa.org/, 2024a.

Mathematical Association of America. AMC 10/12 problems and solutions. https://maa.org/, 2024b.

Meta. Llama 3.2 model card. https://github.com/meta-llama/llama-models/blob/main/models/llama3_2/MODEL_CARD.md, 2024.

Gongrui Nan, Siye Chen, Jing Huang, Mengyu Lu, Dexun Wang, Chunmei Xie, Weiqi Xiong, Xianzhou Zeng, Qixuan Zhou, Yadong Li, et al. Ngrpo: Negative-enhanced group relative policy optimization. *arXiv preprint arXiv:2509.18851*, 2025.

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

Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3967–3976, 2019.

Roberta Raileanu and Tim Rocktäschel. Ride: Rewarding impact-driven exploration for procedurally-generated environments. *arXiv preprint arXiv:2002.12292*, 2020.

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.

Vaishnavi Shrivastava, Ahmed Awadallah, Vidhisha Balachandran, Shivam Garg, Harkirat Behl, and Dimitris Papailiopoulos. Sample more to think less: Group filtered policy optimization for concise reasoning. *arXiv preprint arXiv:2508.09726*, 2025.

Marco Simoni, Aleksandar Fontana, Giulio Rossolini, Andrea Saracino, and Paolo Mori. Gtpo: Stabilizing group relative policy optimization via gradient and entropy control. *arXiv preprint arXiv:2508.03772*, 2025.

Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. RoboSpatial: Teaching spatial understanding to 2D and 3D vision-language models for robotics. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15768–15780. IEEE, 2025. URL https://arxiv.org/abs/2411.16537.

Lihao Sun, Hang Dong, Bo Qiao, Qingwei Lin, Dongmei Zhang, and Saravan Rajmohan. Llm reasoning as trajectories: Step-specific representation geometry and correctness signals. *arXiv preprint arXiv:2604.05655*, 2026.

Yan Sun, Jia Guo, Stanley Kok, Zihao Wang, Zujie Wen, and Zhiqiang Zhang. Efficient reinforcement learning for large language models with intrinsic exploration. *arXiv preprint arXiv:2511.00794*, 2025.

Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019.

Boyuan Wang, Yun Qu, Yuhang Jiang, Jianzhun Shao, Chang Liu, Wenming Yang, and Xiangyang Ji. Llm-empowered state representation for reinforcement learning. *arXiv preprint arXiv:2407.13237*, 2024.

Chaoren Wang, Heng Lu, Xueyao Zhang, Shujie Liu, Yan Lu, Jinyu Li, and Zhizheng Wu. Closing the modality reasoning gap for speech large language models. *arXiv preprint arXiv:2601.05543*, 2026a.

Haochen Wang, Xiangtai Li, Zilong Huang, Anran Wang, Jiacong Wang, Tao Zhang, Jiani Zheng, Sule Bai, Zijian Kang, Jiashi Feng, Zhuochen Wang, and Zhaoxiang Zhang. Traceable evidence enhanced visual grounded reasoning: Evaluation and method. In *International Conference on Learning Representations*, volume 2026, pp. 148769–148794, 2026b. URL https://openreview.net/forum?id=zwMpimSZEq.

Weichu Xie, Haozhe Zhao, Wenpu Liu, Yongfu Zhu, Liang Chen, Minghao Ye, Zirong Chen, Yuqi Xu, Shuai Dong, Ziyue Wang, et al. 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.

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. Regularizing hidden states enables learning generalizable reward model for llms. *Advances in Neural Information Processing Systems*, 37:62279–62309, 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. *Advances in Neural Information Processing Systems*, 38:113222–113244, 2026.

Sihyun Yu, Sangkyung Kwak, Huiwon Jang, Jongheon Jeong, Jonathan Huang, Jinwoo Shin, and Saining Xie. Representation alignment for generation: Training diffusion transformers is easier than you think. *arXiv preprint arXiv:2410.06940*, 2024.

Wentao Yuan, Jiafei Duan, Valts Blukis, Wilbert Pumacay, Ranjay Krishna, Adithyavairavan Murali, Arsalan Mousavian, and Dieter Fox. RoboPoint: A vision-language model for spatial affordance prediction for robotics. *arXiv preprint arXiv:2406.10721*, 2024.

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*, 2025.

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, et al. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*, 2025.

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.

Enshen Zhou, Jingkun An, Cheng Chi, Yi Han, Shanyu Rong, Chi Zhang, Pengwei Wang, Zhongyuan Wang, Tiejun Huang, Lu Sheng, and Shanghang Zhang. RoboRefer: Towards spatial referring with reasoning in vision-language models for robotics. *Advances in Neural Information Processing Systems*, 38:28404–28481, 2026. doi: 10.52202/085713-0957.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

APPENDIX

A Cosine Similarity Distributions

To verify that correct rollouts cluster more tightly than other pairs at the anchor token, we compute pairwise cosine similarities on 1,000 prompts from the DAPO-Math-17K training set, using the base Qwen3-4B model with 10 rollouts per prompt.

Figure 7 shows the distribution of pairwise cosine similarity at the last layer (Layer 36) for three token positions. Position (a) is the *anchor token*, the token immediately before the `\boxed{` marker, where the model has finished reasoning but has not yet begun writing the answer. Here, correct–correct pairs (blue) are clearly shifted toward higher similarity compared to correct–wrong (orange) and wrong–wrong (red) pairs, confirming that correct rollouts naturally cluster more tightly at this position. Position (b) is the token at the start of the answer region, where the model is about to generate the answer digits. The correct–correct distribution concentrates near 1.0, since correct rollouts produce the same answer and thus their hidden states become nearly identical. This leaves no room for an auxiliary loss to provide meaningful gradient signal. Position (c) is the last token of the response (pre-EOS), where the model is about to terminate generation. While some clustering is visible, this position carries no semantic significance for reasoning; any similarity here reflects formatting patterns rather than decision quality.

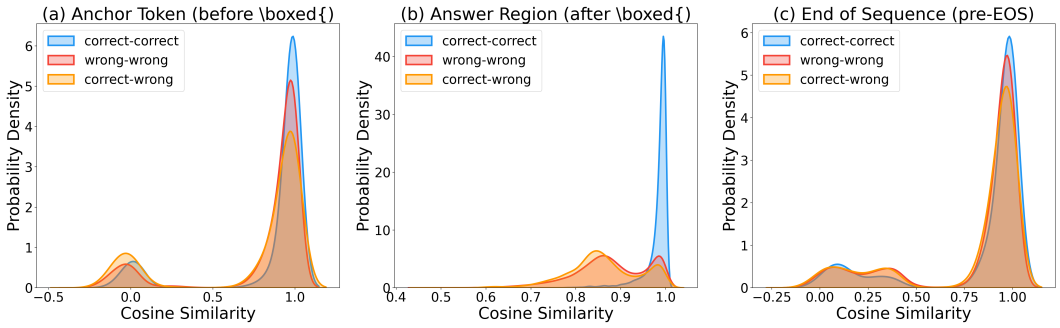


Figure 7: Pairwise cosine similarity distributions at three token positions (Qwen3-4B, Layer 36). (a) At the anchor token, correct–correct pairs cluster more tightly than other pairs. (b) In the answer region, all correct-correct pairs converge near 1.0. (c) At the end of sequence, distributions overlap with no useful structure.

B Training Details

Mini-Batch and Micro-Batch in RL Training. In RLVR training, each optimization step begins with a *rollout phase*: the current policy generates a large batch of responses (the mini-batch) across many prompts. This mini-batch must be large enough to provide stable gradient estimates, since the group-relative advantage A_i is computed within each prompt group and benefits from seeing many groups per update. However, the subsequent *training phase*—which runs forward and backward passes through the full model—cannot fit the entire mini-batch into GPU memory at once. The mini-batch is therefore split into smaller *micro-batches* that are processed sequentially, with gradients accumulated before each optimizer step. This mini-batch/micro-batch structure is standard in RL training and cannot be easily modified without affecting training stability or efficiency.

RL Configuration. For the main mathematical experiments, we use the EasyR1 implementation of DAPO with token-level importance ratios and token-averaged policy loss. Online filtering retains prompt groups whose mean reward is between 0.01 and 0.99. We use symmetric clipping with $\epsilon_{\text{low}} = \epsilon_{\text{high}} = 0.2$, disable the KL penalty, and perform one policy update per rollout batch. The reference embeddings are recomputed for each rollout batch and held fixed during micro-batch gradient accumulation. Model parameters are updated only after all micro-batches have been

Table 6: Training hyperparameters for the main mathematical experiments.

Parameter	Value
Base model	Qwen3-4B
	Qwen3-1.7B
	Qwen3-14B
Training data	DAPO-Math-17K
RL algorithm	DAPO
Learning rate	1×10^{-6}
Optimizer	AdamW (β_1=0.9, β_2=0.999)
Weight decay	1×10^{-2}
LR warmup	0
Clip ratio (high/low)	0.2 / 0.2
KL penalty	disabled
Online filtering	enabled
Rollout n	10
Rollout batch size	64
Global batch size	64
Micro batch size	1
Max prompt length	2048
Max response length	8192
Tensor parallel size	2
GPUs	NVIDIA H200; count varies by run (4 for timing)
Training epochs	2
Save frequency	every 20 steps
Hidden-Align-specific	
λ (alignment)	0.001
λ_{sep} / λ_{div} / λ_{neg}	0
Layer index	-1 (last layer)
Anchor position	pre_boxed_last_token
Warmup start	step 0
Warmup steps	60

processed. The reward is binary answer correctness, without a separate format bonus. Soft overlong penalties are inactive: the response limit is 8,192 tokens, below the configured penalty threshold of 50,000, with an overlong buffer of zero.

For GSPO, we use sequence-level importance ratios and sequence-averaged policy loss; the recorded runs retain the same clipping thresholds and online filtering. All experiments use NVIDIA H200 GPUs.

Alignment Activation Rate. A prompt group contributes to alignment when it contains at least two correct rollouts with valid anchor tokens. For the Qwen3-4B run used in the main results, such groups constitute 87.66% of the logged eligible training groups over steps 1–50, 89.92% over steps 1–200, and 90.71% over steps 61–200, after the alignment-weight warmup. These rates describe the groups used for training after online filtering and show that alignment signals are available for most groups, including early in training.

Training Cost. We estimate the time for 200 Qwen3-4B training steps from checkpoint timestamps for three DAPO runs and three Hidden-Align runs, each using four NVIDIA H200 GPUs. The timing cohort is recorded independently of the accuracy runs in Appendix C.1. Table 7 gives the individual estimates and their mean and sample standard deviation, computed before rounding. The estimated means are approximately 53.6 and 52.9 hours for DAPO and Hidden-Align, respectively. Their difference of 0.68 hours is smaller than the observed across-run standard deviations of 2.37 and 3.20 hours.

We reconstruct elapsed time from the modification times of rank-0 model checkpoints through step 200. The first available checkpoint is at step 20, so we extrapolate steps 0–20 using the observed rate over steps 20–40. For subsequent intervals, we divide the elapsed time by the number of steps and exclude intervals above 30 minutes per step as likely interruptions. We replace their durations using

the run's median valid interval rate among intervals starting at step 100 or later. This procedure is identical for all six runs.

Table 7: Estimated training time for 200 Qwen3-4B steps on four NVIDIA H200 GPUs. SD denotes the sample standard deviation.

Method	**Seed / summary**	**Time (h)**
DAPO	Seed 1	56.34
DAPO	Seed 2	52.21
DAPO	Seed 3	52.27
DAPO	Mean $\pm$ SD	53.61 ± 2.37
DAPO + Hidden-Align	Seed 1	50.64
DAPO + Hidden-Align	Seed 2	51.57
DAPO + Hidden-Align	Seed 3	56.59
DAPO + Hidden-Align	Mean $\pm$ SD	52.93 ± 3.20

A separate 30-step profile uses eight H200 GPUs per method, an upper clipping threshold of 0.28, and no alignment-weight warmup. Averaged over steps 10–30, throughput is 969.9 tokens/s/GPU for DAPO and 973.4 for Hidden-Align; mean step times are 193.6 and 177.5 seconds, respectively. Throughput counts non-padding prompt and response tokens divided by step duration and GPU count. Mean response lengths are 2,170 and 1,978 tokens, respectively, so step-time differences also reflect rollout length. These short-run measurements are separate from the four-GPU, 200-step estimates above.

Qwen3-1.7B Checkpoint Selection. For Qwen3-1.7B, we select the checkpoint with the best training-validation performance for each method, corresponding to step 500 for DAPO and step 520 for DAPO + Hidden-Align.

GSPO Evaluation. Both GSPO and GSPO + Hidden-Align are evaluated at training step 160 on the eight mathematical reasoning benchmarks.

Evaluation Settings. For the main mathematical benchmarks, we sample 32 responses per question at temperature 0.2 on AIME24/25/26, AMC23/24, and HMMT, and generate one response per question at temperature 0 on Minerva Math and OlympiadBench. Accuracy is the fraction of correct responses under these settings. We use a maximum generation length of 32,768 tokens with thinking mode disabled. For pass@k evaluation, we sample $n = 32$ responses per prompt with temperature $= 0.2$. Pass@k is computed using the unbiased estimator (Chen et al., 2021):

$$\text{pass@}k = 1 - \frac{\binom{n-c}{k}}{\binom{n}{k}}$$

where n is the total number of samples and c is the number of correct samples. All evaluations use tensor parallelism of 1 (greedy) or 2 (pass@32) with vLLM, and GPU memory utilization of 0.9.

B.1 Gradient Derivation

We derive the gradient of $\mathcal{L}_{\cos}$ (Eq. 3) with respect to the raw hidden state $\mathbf{h}_i$. Expanding the cosine similarity as $\cos(\mathbf{h}_i, \mathbf{h}_j) = \hat{\mathbf{h}}_i^\top \hat{\mathbf{h}}_j$ with $\hat{\mathbf{h}}_i = \mathbf{h}_i / \|\mathbf{h}_i\|$, we first compute the Jacobian of L2 normalization:

$$\frac{\partial \hat{\mathbf{h}}_i}{\partial \mathbf{h}_i} = \frac{1}{\|\mathbf{h}_i\|} \left(\mathbf{I} - \hat{\mathbf{h}}_i \hat{\mathbf{h}}_i^\top \right) \tag{7}$$

The gradient of a single cosine term $s_{ij} = \hat{\mathbf{h}}_i^\top \hat{\mathbf{h}}_j$ with respect to $\mathbf{h}_i$:

$$\frac{\partial s_{ij}}{\partial \mathbf{h}_i} = \frac{1}{\|\mathbf{h}_i\|} \left(\hat{\mathbf{h}}_j - s_{ij}\, \hat{\mathbf{h}}_i \right) \tag{8}$$

Summing these contributions over all correct pairs and applying the factor $-1/|\mathcal{P}|$ from Eq. 3 yields Eq. 5. Within each local training mini-batch, we average the alignment loss over all valid correct–correct pairs, with pairs formed only among rollouts of the same prompt.

C Statistical Reliability

C.1 Results across Training Seeds

Table 8 reports the per-seed results on Qwen3-4B, with both methods evaluated at training step 200 for all three seeds. The summary in Section 4.2 is computed from the unrounded per-run averages, with sample standard deviation across seeds.

Table 8: Per-seed accuracy (%) on Qwen3-4B. Avg is the unweighted mean over the eight benchmarks.

Seed	AIME24	AIME25	AIME26	AMC23	AMC24	HMMT	Minerva	Olympiad	Avg
DAPO									
Seed 1	48.12	43.44	40.31	88.28	58.75	19.79	53.68	63.35	51.97
Seed 2	49.06	43.75	43.54	88.44	61.39	20.73	54.78	66.02	53.46
Seed 3	44.17	44.06	41.15	89.06	56.53	20.83	54.78	61.87	51.56
DAPO + Hidden-Align									
Seed 1	56.67	49.90	50.52	90.62	63.40	27.81	59.19	67.36	58.18
Seed 2	49.17	45.73	46.46	90.16	60.49	23.12	58.46	65.13	54.84
Seed 3	52.08	44.69	44.69	88.44	58.54	22.60	57.72	66.17	54.37

C.2 Question-Level Confidence Interval

For the Qwen3-4B Seed 1 comparison in Table 1, we estimate uncertainty using a paired bootstrap over questions, stratified by benchmark (Efron, 1979). Both methods are evaluated at step 200, using the same recorded evaluation outcomes as in the main table. Within each benchmark, we resample questions with replacement, using the same sampled questions for both methods and retaining all responses to each question (32 for the six competition benchmarks and one for Minerva Math and OlympiadBench). We compute accuracy within each benchmark and average the eight benchmark scores with equal weights. Using 20,000 bootstrap resamples with random seed 20260926, the 2.5th and 97.5th percentiles of the accuracy difference give a 95% confidence interval of [4.48, 7.96] percentage points for the observed 6.22-point improvement. This interval measures question-level uncertainty for the fixed checkpoints and observed responses, separately from the training-seed variation in Appendix C.1.

D Additional Generalization Experiments

D.1 Llama Results

Table 9 reports the six-benchmark results for Llama-3.2-3B-Instruct, with both DAPO and DAPO + Hidden-Align evaluated at step 200. Its Avg is the unweighted mean of all six listed benchmarks. The Avg in the main-text Table 2 instead averages AIME24, AMC24, Math500 (Lightman et al., 2024), and Minerva Math with equal weights. For Math500, we use one greedy response per question (temperature 0) over all 500 test questions, following the same evaluation harness as the Qwen models. Answers are checked using rule-based extraction and mathematical equivalence tests, with an LLM-based extraction and equivalence fallback. Accuracy is the fraction of correct responses. AIME25, AIME26, and HMMT yield near-zero scores for both methods on this model and offer limited discrimination between them; they are excluded from the six-benchmark average.

Table 9: Llama-3.2-3B-Instruct accuracy (%). Avg is the unweighted mean of the six benchmarks. Bold values indicate improvements over DAPO.

Method	AIME24	AMC23	AMC24	Math500	Minerva	Olympiad	Avg
Base	6.77	23.12	8.47	40.60	15.07	11.87	17.65
DAPO	7.92	28.20	13.68	43.20	23.16	15.58	21.96
DAPO + Hidden-Align	**10.31**	**30.86**	13.33	**49.00**	21.32	**17.95**	**23.80**

D.2 Spatial Reasoning Setup

We evaluate Qwen3-VL-30B-A3B-Instruct on EmbSpatialBench (Du et al., 2024), RoboSpatial-Home (Song et al., 2025), RefSpatialBench (Zhou et al., 2026), and TreeBench (Wang et al., 2026b), abbreviated as EmbSp., RoboSp., RefSp., and TreeBench in Table 3.

Training data. The spatial training mixture draws on RoboPoint (Yuan et al., 2024), the RefSpatial training corpus (Zhou et al., 2026), TreeVGR-SFT-35K and TreeVGR-RL-37K (Wang et al., 2026b), EmbSpatial-SFT (Du et al., 2024), Euclid30K (Lian et al., 2026), and SpaceThinker[1]. It covers spatial grounding, pointing, multiple-choice questions, grounded question answering, mathematics, short-answer questions, and bounding-box tasks. The grounded-QA subset pairs textual answers with normalized bounding boxes for referenced objects. We use a validation set of 961 examples for training monitoring.

Training configuration. Training uses a rollout batch size of 256, a global batch size of 64, and eight rollouts per prompt, with a learning rate of 10^{-6}. The maximum prompt and response lengths are 2,048 and 4,096 tokens, respectively. We set the lower and upper clipping thresholds to 0.2 and 0.28, enable KL regularization, and disable online filtering. Tensor parallelism is set to 1, and the point-verification threshold is 0.1. We configure three training epochs and save checkpoints every 10 steps. Hidden-Align uses last-layer hidden states with alignment weight $\lambda = 0.001$, no alignment-weight warmup, and all other auxiliary loss weights set to zero. The anchor is the token immediately preceding the closing `</think>` tag.

Training rewards. Each sample's task type determines its verifier. Multiple-choice and short-answer tasks use answer matching, with gpt-oss-20b as a fallback when rule-based matching fails; spatial-grounding QA uses the same text-verification procedure. For pointing tasks with coordinate targets, a target point is correct if its nearest predicted point is within a Euclidean distance of 0.1 in normalized coordinates, and scores are averaged over target points. For region-mask targets, any predicted point inside the target mask yields a correct score. Grounded QA equally weights text correctness and mean bounding-box intersection over union (IoU), with predicted boxes extracted from the reasoning text; standalone bounding-box tasks use IoU. Mathematical answers use rule-based mathematical verification with an LLM answer-extraction fallback. Task accuracy and format scores receive weights of 0.9 and 0.1, respectively. Responses must begin with a valid, nonempty `<think>... </think>` block to receive a nonzero reward.

Evaluation. For the four evaluation benchmarks, inputs are image–question pairs, and final answers are coordinates, yes/no answers, or multiple-choice options. We use greedy decoding (temperature 0) with one response per question and a maximum of 32,768 generated tokens on all four benchmarks. Table 10 summarizes the scoring rules. For EmbSpatialBench and TreeBench, we first extract the option by rule and use gpt-oss-20b to match the response to an option only when rule-based extraction fails. RoboSpatialHome and RefSpatialBench use rule-based scoring without an LLM judge. Both DAPO and DAPO + Hidden-Align are evaluated at step 160, and Avg is the unweighted mean over the four spatial benchmarks.

D.3 Alternative Answer Format

We test Hidden-Align with a natural-language answer marker on Qwen3-4B. This experiment jointly changes the training-time answer marker, verifier, and anchor rule from the boxed-answer convention to `answer is`. We locate the last valid occurrence of the lowercase marker, requiring that it is not preceded by a letter and is followed by a space. The anchor is the token immediately before the token containing the beginning of this marker. Both DAPO and DAPO + Hidden-Align use the modified answer convention and are evaluated at step 160.

Table 11 reports the complete eight-benchmark results. Hidden-Align improves accuracy on four benchmarks, decreases it on three, and matches DAPO on Minerva Math. The unweighted eight-benchmark average increases from 47.45 to 47.98 (+0.53 percentage points). These results support applicability under another answer convention.

[1] `https://huggingface.co/datasets/remyxai/SpaceThinker`

Table 10: Evaluation protocols for spatial reasoning. Each benchmark score is averaged over its evaluation items.

Benchmark	Items	Scoring rule
EmbSpatialBench	3,640	Exact match between the extracted option and the correct option.
RoboSpatialHome	350	Exact match for yes/no questions. For pointing questions, a response is correct if either of the first two predicted points falls inside the target mask.
RefSpatialBench	277	Each item is scored by the fraction of predicted points inside the target mask. The score is averaged over all 277 items: Location (100), Placement (100), and Unseen (77).
TreeBench	405	Exact option match. Bounding-box IoU is an auxiliary metric and does not contribute to the reported accuracy.

Table 11: Accuracy (%) with the `answer is` marker on Qwen3-4B at step 160. Avg is the unweighted mean over the eight benchmarks.

Method	AIME24	AIME25	AIME26	AMC23	AMC24	HMMT	Minerva	Olympiad	Avg
DAPO	35.73	34.48	**34.69**	81.72	**55.35**	18.44	**58.09**	**61.13**	47.45
DAPO + Hidden-Align	**40.73**	**35.62**	33.44	**82.73**	53.06	**19.17**	**58.09**	60.98	**47.98**

E Additional Ablation Studies

All ablations are conducted on Qwen3-4B with DAPO training.

Layer Depth. We apply the alignment loss at different transformer layers. As shown in Figure 1(a), only the last layer exhibits strong position-dependent variation in cosine similarity; shallower layers show weaker variation, meaning the anchor token is less distinctive at those depths. Table 12 confirms that the last layer (Layer 36) produces the largest gain, while intermediate layers yield smaller gains.

Loss Weight. We vary λ from 0.0001 to 0.01. Too small a value provides insufficient signal; too large a value overpowers the RL objective and destabilizes training. Table 13 shows that $\lambda = 0.001$ is optimal, with $\lambda = 0.01$ causing training divergence.

Table 12: Layer ablation (alignment loss, anchor token, $\lambda = 0.001$).

Layer	AIME24	AIME26	AMC24	HMMT	Avg
DAPO (baseline)	48.12	40.31	58.75	19.79	41.74
Layer 16	45.00	43.85	62.15	24.38	43.85
Layer 24	47.40	44.27	61.88	21.98	43.88
Layer 28	44.69	44.58	62.99	20.62	43.22
Layer 32	49.48	43.44	60.07	23.75	44.19
Layer 36 (ours)	**56.67**	**50.52**	**63.40**	**27.81**	**49.60**

Table 13: Loss weight ablation (alignment loss, anchor token, last layer).

λ	AIME24	AIME26	AMC24	HMMT	Avg
DAPO (baseline)	48.12	40.31	58.75	19.79	41.74
0.0001	47.29	44.90	59.03	20.42	42.91
0.0005	50.21	43.75	59.31	21.77	43.76
0.001 (ours)	**56.67**	**50.52**	**63.40**	**27.81**	**49.60**
0.005	42.71	39.06	57.15	22.81	40.43
0.01[†]	35.83	35.83	52.71	19.06	35.86

[†] Training diverged after step 100; results reported at step 80.

F Case Study

Table 14 illustrates the "knowing but not doing" phenomenon that Hidden-Align addresses. On this geometry problem (AIME 2024 I, Problem 11), the DAPO baseline *can* find the correct answer (104) in some rollouts (11/32), but its most frequent incorrect output is 291, a plausible intermediate value. With Hidden-Align, the model's correct-answer rollout rate increases dramatically (26/32). These results show that Hidden-Align makes the model more likely to produce a correct solution it could already find.

Table 14: Case study on AIME 2024 I Problem 11. Hidden-Align increases the correct-answer sampling frequency from 11/32 to 26/32.

Problem	Rectangles $ABCD$ and $EFGH$ are drawn such that D, E, C, F are collinear. Also, A, D, H, G all lie on a circle. If $BC = 16$, $AB = 107$, $FG = 17$, and $EF = 184$, what is the length of CE?
Ground truth	$\boxed{104}$
DAPO baseline	11/32 correct; most frequent wrong answer: 291 (8/32)
Hidden-Align	26/32 correct

G Hidden States vs. Logits Alignment

To verify that aligning hidden states is not equivalent to aligning output logits, we run an additional experiment replacing the last-layer hidden states with the pre-softmax logits in the alignment loss, keeping all other settings identical. Table 15 shows that hidden-state alignment yields a larger average gain over the DAPO baseline than logit alignment (+7.86 vs. +0.89 percentage points). These results support the choice of hidden-state alignment.

Table 15: Hidden-state vs. logit alignment (Qwen3-4B, anchor token, $\lambda = 0.001$). Avg is the unweighted mean over the four benchmarks.

Method	**AIME24**	**AIME26**	**AMC24**	**HMMT**	**Avg**
DAPO (baseline)	48.12	40.31	58.75	19.79	41.74
Logit (pre-softmax)	46.67	45.42	58.54	19.90	42.63
Hidden-state (ours)	**56.67**	**50.52**	**63.40**	**27.81**	**49.60**

H Accuracy under Matched Answer Formats

We test whether the accuracy gain persists when the two methods produce answers with the same format status. We compare Qwen3-4B DAPO and DAPO + Hidden-Align at step 200 on the eight main benchmarks, matching responses by benchmark, question, and repeat index. A response is tagged as boxed if it contains an explicit balanced `\boxed{...}` marker. Non-boxed responses can still be correct under the answer extractor, so their format status is separate from correctness. Table 16 reports pooled accuracy over all 7,506 response pairs and the subsets where both methods produce boxed or non-boxed answers. Here, responses are weighted equally, whereas Table 1 weights benchmarks equally. The full set also includes pairs with different format status across methods.

Table 16: Accuracy (%) under matched answer formats on Qwen3-4B at step 200. Δ is the gain over DAPO in percentage points, computed before rounding.

Subset	**Pairs**	**DAPO**	**+ Hidden-Align**	Δ
All responses	7,506	53.36	59.46	+6.10
Both boxed	5,265	67.29	73.05	+5.75
Both non-boxed	644	5.59	6.37	+0.78

Hidden-Align retains a 5.75-point accuracy advantage within the subset where both methods produce explicit boxed answers.

I Confidence and Diversity of Incorrect Answers

We examine whether Hidden-Align increases confidence in incorrect answers or concentrates them on a single wrong answer. Both analyses use Qwen3-4B DAPO and DAPO + Hidden-Align at step 200.

Confidence in Incorrect Answers. For an incorrect response with a valid boxed answer span, confidence is the geometric mean of the teacher-forced probabilities assigned by its own model to the tokens inside that span, conditioned on the prompt and preceding generated tokens. We analyze the eight main benchmarks and compare both the full set of valid incorrect boxed answers for each method and the paired subset where both methods produce incorrect boxed answers for the same benchmark, question, and repeat index.

We estimate uncertainty with a paired, benchmark-stratified prompt-cluster bootstrap (Efron, 1979). Within each benchmark, we sample its original number of questions with replacement, retaining all responses for each sampled question and using the same question multiplicities for both methods. We recompute the pooled mean confidence difference for each replicate and report 95% percentile intervals from 10,000 resamples with random seed 20260925. Table 17 shows that both intervals include zero, providing no clear evidence of increased confidence in incorrect boxed answers.

Table 17: Mean confidence in incorrect boxed answers on Qwen3-4B at step 200. Counts are DAPO / Hidden-Align; Δ and its 95% confidence interval compare Hidden-Align with DAPO.

Subset	Counts	DAPO	+ Hidden-Align	Δ	95% CI
All wrong boxed	2,061 / 1,959	0.9800	0.9854	+0.0054	$[-0.0010, 0.0119]$
Both wrong boxed	1,180 / 1,180	0.9789	0.9844	+0.0055	$[-0.0039, 0.0154]$

Diversity of Incorrect Answers. We use the 205 questions from the six competition benchmarks, with 32 sampled responses per question for each method. For each question, we collect the extracted incorrect answers, lowercase them, and remove whitespace, `$`, `\left`, and `\right`. We retain questions with at least two incorrect responses whose normalized extracted answers are nonempty. A question exhibits *mode collapse* if all its retained incorrect answers are identical. We also compute the Shannon entropy of the empirical wrong-answer distribution using natural logarithms, then average over eligible questions.

Table 18: Diversity of incorrect answers on Qwen3-4B at step 200. Rates and mean entropies use questions with at least two nonempty incorrect answers; entropy is measured in nats.

Method	Eligible questions	Collapsed	Rate (%)	Entropy
DAPO	141	12	8.51	1.501
DAPO + Hidden-Align	134	11	8.21	1.525

Table 18 shows similar collapse rates and mean entropies, indicating comparable diversity among the extracted incorrect answers.

J LIMITATIONS

- Extending to other tasks requires identifying the corresponding anchor position for each answer format.
- Hidden-Align targets tasks with verifiable answers. Its effectiveness on open-ended creative generation, where correctness is not uniquely defined and diversity may itself be the goal, remains untested.
- Validation on even larger scales (70B+) is future work.

K PASS@k ACROSS MODEL SCALES

Table 19 extends the pass@k analysis to Qwen3-1.7B and Qwen3-14B using the same checkpoints as Table 1. We compute the unbiased estimator described in Appendix B from 32 responses per question. Hidden-Align improves the six-benchmark average at every tested value of k for both model scales, although individual benchmarks do not always improve.

Table 19: Pass@k (%) on six competition benchmarks for Qwen3-1.7B and Qwen3-14B. Avg is the unweighted mean over these six benchmarks. Bold values indicate improvements over DAPO.

Benchmark	**Method**	$k=1$	$k=2$	$k=4$	$k=8$	$k=16$	$k=32$
Qwen3-1.7B							
AIME24	DAPO	29.06	37.06	44.71	51.12	56.69	63.33
	+ Hidden-Align	**31.87**	**40.93**	**48.89**	**55.74**	**61.47**	**66.67**
AIME25	DAPO	25.83	32.10	37.80	42.97	48.19	53.33
	+ Hidden-Align	**27.50**	**32.55**	36.81	41.19	46.77	53.33
AIME26	DAPO	19.90	26.85	33.35	38.57	42.48	46.67
	+ Hidden-Align	**24.27**	**31.92**	**38.32**	**43.48**	**48.06**	**53.33**
AMC23	DAPO	68.12	78.60	85.50	89.71	92.79	95.00
	+ Hidden-Align	**73.83**	**82.70**	**88.20**	**91.52**	**93.77**	95.00
AMC24	DAPO	47.43	54.64	60.23	65.05	69.62	75.56
	+ Hidden-Align	**47.57**	54.28	60.15	**65.87**	**70.69**	73.33
HMMT	DAPO	14.06	17.53	20.12	21.62	23.33	26.67
	+ Hidden-Align	**16.98**	**20.29**	**22.54**	**25.11**	**28.82**	**33.33**
Avg	DAPO	34.07	41.13	46.95	51.51	55.52	60.09
	+ Hidden-Align	**37.00**	**43.78**	**49.15**	**53.82**	**58.27**	**62.50**
Qwen3-14B							
AIME24	DAPO	60.94	70.91	76.39	79.59	82.53	86.67
	+ Hidden-Align	**68.12**	**77.75**	**82.58**	**85.41**	**87.88**	**90.00**
AIME25	DAPO	51.98	61.61	69.62	75.99	81.48	86.67
	+ Hidden-Align	**61.67**	**71.32**	**77.23**	**81.06**	**84.09**	86.67
AIME26	DAPO	49.79	60.87	68.62	74.37	79.21	83.33
	+ Hidden-Align	**60.94**	**70.65**	**76.96**	**81.48**	**84.87**	**86.67**
AMC23	DAPO	92.27	95.96	97.19	97.48	97.50	97.50
	+ Hidden-Align	**93.98**	**96.73**	**97.76**	**98.58**	**99.40**	**100.00**
AMC24	DAPO	64.10	69.64	73.44	75.36	76.55	77.78
	+ Hidden-Align	**65.62**	**71.17**	**74.84**	**77.13**	**79.41**	**82.22**
HMMT	DAPO	31.35	39.49	46.39	52.03	56.74	60.00
	+ Hidden-Align	**34.58**	**42.62**	**49.99**	**56.67**	**63.27**	**70.00**
Avg	DAPO	58.40	66.41	71.94	75.80	79.00	81.99
	+ Hidden-Align	**64.15**	**71.71**	**76.56**	**80.05**	**83.16**	**85.93**