OISD: On-Policy Internal Self-Distillation of Language Models

Anonymous ACL submission

Abstract

Recent reinforcement learning (RL) post-training approaches primarily optimize the final output policy using sparse outcome-level rewards, while largely overlooking predictive signals encoded in intermediate representations. In this paper, we introduce a new paradigm called on-policy internal self-distillation and propose the OISD framework, which improves reasoning by transferring on-policy predictive signals from the final layer to intermediate representations. During rollout and Group Relative Policy Optimization (GRPO) optimization, the final layer acts as both the policy and a detached internal teacher for selected intermediate layers, which are guided to align with it through two complementary mechanisms: logit alignment, which transfers high-level reasoning behaviors (*how to think*), and attention alignment, which enforces consistent attention patterns (*where to look*) from the final layer to the selected intermediate layer, both without requiring external privileged information. Our OISD, together with GRPO, employs signed advantage-weighted Jensen–Shannon alignment to distill informative intermediate representations while preserving policy consistency under a unified acting policy. Experimental results demonstrate the effectiveness of OISD, with substantial and consistent improvements over strong reasoning RL baselines across four mathematical reasoning tasks.

1 Introduction

Large language models (LLMs) have achieved remarkable advances in mathematical reasoning, coding, and complex instruction-following (Yao et al., 2022; Wei et al., 2022). A key driver of this progress is reasoning-oriented post-training, in which models are further optimized after pretraining and supervised fine-tuning to generate more accurate, coherent, and reliable reasoning trajectories (Shao et al., 2024; Guo et al., 2025).

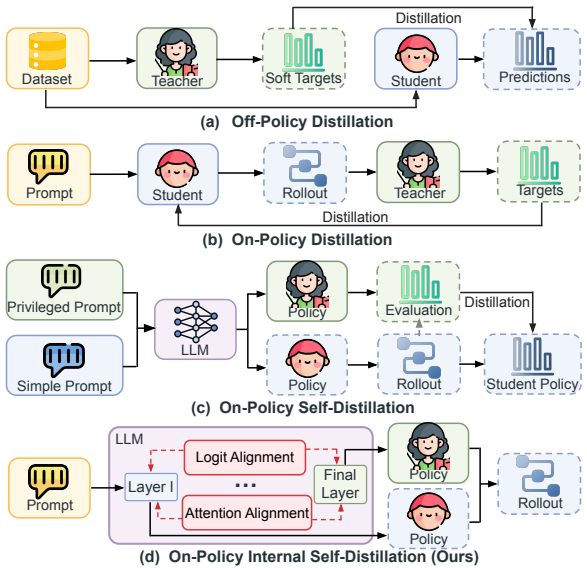

Figure 1: Comparison of four distillation paradigms: (a) Off-Policy Distillation, (b) On-Policy Distillation, (c) On-Policy Self-Distillation, and (d) On-Policy Internal Self-Distillation (Ours). The comparison highlights differences in teacher–student roles, rollout generation, and the location where distillation is applied.

Recent reinforcement learning with verifiable rewards (RLVR), particularly Group Relative Policy Optimization (GRPO)-style training, has significantly improved reasoning-oriented post-training without requiring dense human supervision (Shao et al., 2024; Guo et al., 2025; Zheng et al., 2025a; Yu et al., 2026). However, these approaches remain heavily final-output-centric, optimizing only the final acting policy with sparse outcome rewards while largely overlooking intermediate predictions, reasoning processes, and internal representations.

Knowledge distillation (Hinton et al., 2015; Bengio et al., 2015) provides dense token-level supervision along the reasoning trajectory, complementing sparse outcome rewards. In classical off-policy distillation, the student learns from fixed teacher-generated traces, creating a distribution

mismatch because the training trajectories come from the teacher rather than the student's own generations (Agarwal et al., 2024; Xu et al., 2025). On-policy distillation reduces this gap by supervising student-generated rollouts directly (Agarwal et al., 2024), but most methods still rely on external teacher models. Recent on-policy self-distillation methods remove the external teacher by augmenting the same model with privileged traces, feedback, or auxiliary prompts (Zhao et al., 2026). However, supervision still depends on externally injected privileged information rather than the model's own internal computations.

Existing off-policy and on-policy distillation methods primarily focus on improving model accuracy and reasoning capability, with limited exploration of leveraging the model's own internal mechanisms for distillation. However, prior studies (Yang et al., 2025b; Zhu et al., 2025) show that reasoning representations evolve systematically across layers: earlier layers preserve broader candidate predictions, intermediate layers integrate contextual and task-relevant information, and deeper layers progressively converge toward the final next-token decision. This hierarchical organization suggests that different layers serve distinct functional roles in the reasoning process.

Meanwhile, the logit lens framework (nostalgebraist, 2020) provides initial insights by projecting intermediate representations into token space using the unembedding matrix. This reveals that the residual stream in Transformer architectures contains rich predictive information that evolves progressively across layers and modules (Dai et al., 2022; Gupta et al., 2025). All these insights motivate leveraging the model's own internal computations as an intrinsic source of supervision for distillation.

In this paper, we propose OISD, an on-policy internal self-distillation framework for LLM reasoning. During training, OISD keeps the final layer as the sole acting policy for rollout and GRPO optimization, while using it as a detached internal teacher. Following the logit lens framework (Belrose et al., 2023), a selected intermediate layer is projected into the token space to form an auxiliary internal student on the same on-policy rollout. The student learns from two complementary signals: logit alignment, which transfers high-level reasoning behaviors (*how to think*), and attention alignment, which enforces consistent attention patterns (*where to look*) from the final layer to the selected intermediate layer, both without requiring external privileged information. Such a formulation encourages the model to learn stronger intermediate representations. During inference, the teacher model is used as the final model. OISD introduces dense internal supervision without requiring an external teacher, privileged prompts, or multiple acting policies, motivating the term *on-policy internal self-distillation*. A comparison between our OISD and other methods is shown in Figure 1.

1.1 Contributions

Our contributions are summarized as follows:

- We introduce a new paradigm called *on-policy internal self-distillation* for reasoning RL that leverages the model's own internal computations as supervision while preserving a single acting policy for rollout and optimization.
- We propose OISD, an on-policy internal self-distillation framework that uses the final layer as a detached internal teacher and an intermediate layer as an auxiliary internal student and transfers on-policy predictive signals from the final layer to intermediate representations.
- We develop two complementary internal alignment objectives: logit alignment for *how to think* by transferring predictive beliefs, and attention alignment for *where to look* by transferring evidence-routing behavior.
- We demonstrate substantial and consistent improvements over strong baselines across four mathematical reasoning benchmarks.

We hope this research initiates the investigation of on-policy internal self-distillation and encourages future research in this direction.

2 Related Work

Knowledge Distillation for Reasoning. Recent Reinforcement Learning from Human Feedback (RLHF) methods based on PPO provide a standard framework for preference optimization (Schulman et al., 2017; Ouyang et al., 2022). Later, reasoning-oriented RL methods, such as GRPO, DeepSeek-R1, DAPO, GSPO, Qwen3, and RAIF, leverage verifiable rewards and group-based policy optimization to improve long context reasoning (Shao et al., 2024; Guo et al., 2025; Zheng et al., 2025a; Yang et al., 2025a; Qin et al., 2026; Yu et al., 2026). However, supervision remains largely sparse, outcome-level, and focused on the final acting policy.

To provide denser reasoning supervision, prior work (Hinton et al., 2015; Bengio et al., 2015) often distills teacher distributions or reasoning traces into the student. Yet off-policy distillation introduces a training–inference mismatch, since the student is trained on teacher-generated trajectories but must reason from its own internal states during inference.

On-policy distillation reduces this mismatch by generating rollouts from the student and using the teacher's per-token log-probabilities as dense supervision on states actually visited by the student (Agarwal et al., 2024; Xu et al., 2025). More recent self-distillation methods reduce reliance on external teachers by enabling the model to supervise itself using privileged information or richer feedback (Agarwal et al., 2024; Hübotter et al., 2026; Shenfeld et al., 2026). For example, OPSD (Zhao et al., 2026) leverages privileged solutions, SDPO (Hübotter et al., 2026) exploits environment feedback, OPCD (Ye et al., 2026) distills context-conditioned behavior into model weights, and CODI (Shen et al., 2025) compresses explicit reasoning into continuous representations.

Our method places the teacher entirely within the model itself: the detached final layer supervises an intermediate layer on the same on-policy rollout while remaining the sole acting policy.

Internal Signals for Reasoning. Early research efforts, such as deeply-supervised nets (DSN) (Lee et al., 2015), set a broader optimization precedent for attaching auxiliary losses to intermediate representations. More recently, Layerwise readouts show that intermediate Transformer representations already contain meaningful predictive structure before the final layer is reached. Logit-lens and tuned-lens analyses decode hidden states into vocabulary distributions, making prediction trajectories across layers directly observable (nostalgebraist, 2020; Belrose et al., 2023).

Later, DoLa uses contrasts between layers during decoding, while LayerSkip trains intermediate exits for efficient inference and self-speculative decoding (Chuang et al., 2024; Elhoushi et al., 2024). In particular, BuPO pioneered the investigation in optimizing fine-grained internal layer policies to guide the overall policy of the language model during the initial training stages (Tan et al., 2025). However, the authors observed that excessive alignment can lead to performance collapse, suggesting that intermediate policy optimization should be applied only for limited training iterations. Other approaches use layerwise signals for analysis, decoding, early exit, or inference efficiency rather than as auxiliary supervision (Kapadia et al., 2026).

Reasoning supervision can also arise from the reasoning process itself, not only from final answers. CODI compresses explicit reasoning into continuous internal representations through self-distillation (Shen et al., 2025), while step-level verification methods study supervision over intermediate reasoning steps (Lightman et al., 2024; Zheng et al., 2025b; Song et al., 2025). Attention provides another process-level signal by revealing how information is routed during token prediction. For example, MoLSAKI transfers stepwise attention from a teacher to a student during reasoning distillation (Chen et al., 2025), and RAL directly optimizes internal attention distributions through policy-gradient training (Li et al., 2026).

Our OISD develops two complementary internal alignment objectives in an on-policy setting: logit alignment supervises *how to think*, attention alignment supervises *where to look*, and both are distilled from the final layer to an intermediate layer on the same rollout.

3 On-Policy Internal Self-Distillation

In this section, we present OISD, an on-policy internal self-distillation framework that uses the final layer as a detached internal teacher and an intermediate layer as an auxiliary internal student during reasoning RL. We further introduce two complementary internal alignment objectives for supervising intermediate reasoning representations.

3.1 Transformer Depth as a Vertical Reasoning Axis

CoT reasoning exposes intermediate computation through additional reasoning tokens, expanding reasoning *horizontally* along the temporal dimension (Wei et al., 2022). More recent latent reasoning methods further extend this paradigm by evolving hidden states or recurrent activations across reasoning steps rather than fully expressing reasoning in natural language (Hao et al., 2024; Shen et al., 2025). Following (Zhu et al., 2025), these approaches primarily expand reasoning capacity through hidden-state evolution over time.

OISD instead explores a complementary *vertical* reasoning paradigm within a standard decoder-only Transformer consisting of L stacked layers. Rather than extending reasoning through additional tokens, recurrent activations, or latent rollout steps,

OISD leverages the intermediate computations already formed across model depth. The final layer supervises intermediate layers during the same forward pass, allowing reasoning signals to propagate *vertically* through the network without introducing latent tokens or modifying the decoding procedure.

To illustrate the vertical computation within autoregressive generation, following (Zhang et al., 2025; Tan et al., 2025), we formalize one decoding step. Let $\mathbf{x} = [x_1, \ldots, x_M]$ denote the original input prompt, and let $\mathbf{y}_{1:t-1} = [y_1, \ldots, y_{t-1}]$ denote the response prefix generated before step t, with $\mathbf{y}_{1:0} = \emptyset$ for the first decoding step. At generation step t, the model receives the full decoding context

$$\mathbf{c}_t = [\mathbf{x}; \mathbf{y}_{1:t-1}], \qquad t = 1, \ldots, T. \tag{1}$$

The model maps this context into initial hidden states $\mathbf{H}_t^0 \in \mathbb{R}^{|\mathbf{c}_t| \times d_{\text{model}}}$ using the embedding matrix $\mathbf{E} \in \mathbb{R}^{N \times d_{\text{model}}}$. Each Transformer layer then updates the residual stream over all context positions through multi-head self-attention (MHSA) and feed-forward network (FFN) transformations:

$$\begin{aligned} \mathbf{A}_t^l &= \mathrm{MHSA}\Big(\mathrm{LN}(\mathbf{H}_t^{l-1})\Big), \\ \bar{\mathbf{H}}_t^l &= \mathbf{H}_t^{l-1} + \mathbf{A}_t^l, \\ \mathbf{F}_t^l &= \mathrm{FFN}\Big(\mathrm{LN}(\bar{\mathbf{H}}_t^l)\Big), \\ \mathbf{H}_t^l &= \bar{\mathbf{H}}_t^l + \mathbf{F}_t^l, \end{aligned} \tag{2}$$

for $l = 1, \ldots, L$, where $\mathrm{LN}(\cdot)$ denotes layer normalization (Ba et al., 2016).

This formulation separates two axes of computation. Along the autoregressive sequence axis, the context grows from $\mathbf{c}_t = [\mathbf{x}; \mathbf{y}_{1:t-1}]$ to $\mathbf{c}_{t+1} = [\mathbf{x}; \mathbf{y}_{1:t}]$ after sampling y_t. Along the layer axis, the model performs vertical refinement, progressively transforming representations from shallow to deep layers. After L layers, the next-token policy is obtained from the final hidden state:

$$\mathbf{p}_t = \mathrm{Softmax}\left(\mathrm{LN}(\mathbf{h}_t^L)\mathbf{E}_{\mathrm{u}}^\top\right), \quad y_t \sim \mathbf{p}_t, \tag{3}$$

where $\mathbf{h}_t^L \in \mathbb{R}^{d_{\text{model}}}$ is the final-layer hidden state at the last position of context $\mathbf{c}_t$, $\mathbf{E}_{\mathrm{u}} \in \mathbb{R}^{N \times d_{\text{model}}}$ is the unembedding matrix, and $\mathbf{p}_t \in \mathbb{R}^N$ denotes the distribution over the next generated token y_t. The hidden states at layer l can be recursively expanded as

$$\mathbf{H}_t^l = \mathbf{H}_t^0 + \sum_{i=1}^{l} \mathbf{A}_t^i + \sum_{i=1}^{l} \mathbf{F}_t^i, \tag{4}$$

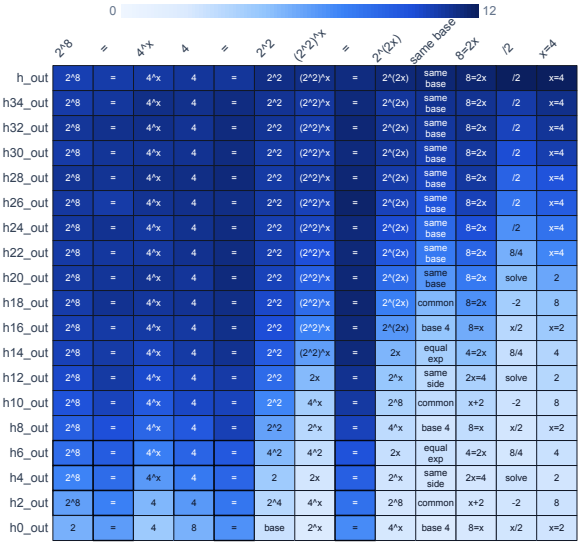

Figure 2: Layerwise logit-lens visualization of a reasoning trajectory of Qwen3-4B. Columns denote generated reasoning tokens and rows denote Transformer layers. Each cell shows the top token from the projected hidden state $\mathbf{p}_t^l$, with darker color indicating higher logits. The visualization reveals progressive refinement, where local symbols emerge in early layers and multi-step algebraic reasoning stabilizes in deeper layers.

showing that each layer accumulates transformations from all preceding layers.

For each intermediate hidden state at layer i, we can then project it back into token space using the same unembedding matrix:

$$\mathbf{p}_t^l = \mathrm{Softmax}\left(\mathrm{LN}(\mathbf{h}_t^l)\mathbf{E}_{\mathrm{u}}^\top\right), \tag{5}$$

where $\mathbf{p}_t^l$ is the layer-l next-token distribution induced by the hidden state $\mathbf{h}_t^l$. This projection leads to a provisional policy $y_t^l \sim \mathbf{p}_t^l$. Figure 2 visualizes the logit lens across layers.

3.2 Guided Shallow Students, Stronger Deep Teachers

Rather than treating intermediate states as transient features, OISD attaches auxiliary losses to a fixed layer $l < L$, treating $\mathbf{h}_t^l$ as a shallow student and $\mathbf{h}_t^L$ as the deep teacher within the same forward pass. The teacher supervises the student through its token distribution and attention pattern, capturing both the belief reached by the model and the context used to form it. Because the student and teacher are induced from the same model at different depths, OISD uses the deep layer to guide better shallow representations; improving these early representations then strengthens the later computation and final-layer teacher.

3.2.1 How to Think: Reward-Conditioned Logit Alignment

The readout $y_t^l \sim \mathbf{p}_t^l$ reveals the provisional next-token belief encoded at layer l, which we interpret as the shallow student's intermediate thinking process before deeper vertical refinement. OISD transfers token-level beliefs from the deep teacher to the shadow student by comparing their logit-lens distributions. With temperature τ, the student and teacher distributions are

$$\begin{aligned} \mathbf{p}_t^{l,\tau} &= \mathrm{Softmax}\left(\mathrm{LN}(\mathbf{h}_t^l)\mathbf{E}_{\mathrm{u}}^\top/\tau\right), \\ \mathbf{q}_t^{L,\tau} &= \mathrm{Softmax}\left(\mathrm{LN}(\mathbf{h}_t^L)\mathbf{E}_{\mathrm{u}}^\top/\tau\right). \end{aligned} \tag{6}$$

Their discrepancy can be measured by a bounded divergence measure. Formally, our logit-alignment loss at layer l is defined as

$$\mathcal{L}_{\text{think}}^{(l)} = \mathbb{E}_t\left[A_t\,\mathrm{JS}\left(\mathbf{p}_t^{l,\tau}, \mathrm{sg}(\mathbf{q}_t^{L,\tau})\right)\right], \tag{7}$$

where A_t is the assigned sequence-level GRPO advantage, JS denotes the Jensen–Shannon divergence (Menéndez et al., 1997), and $\mathrm{sg}(\cdot)$ denotes stop-gradient, which treats the teacher distribution as a fixed target. We weight the divergence by A_t: high-advantage tokens pull the shallow student toward the deep teacher, while low-advantage tokens push it away from the teacher to avoid reinforcing poor continuations. We use the clipped advantage $A_t = \mathrm{clip}(A_t, -c, c)$, with a constant c.

3.2.2 Where to Look: Reward-Conditioned Attention Alignment

At each generation step, an LLM attends to selected parts of the prompt and previously generated tokens, such as the problem statement, earlier intermediate results, or nearby reasoning steps. OISD aligns attention patterns, using final-layer attention to guide the shallow student toward the evidence-routing behavior before producing the next token.

For layer l and head h, let $\mathbf{Q}_h^l, \mathbf{K}_h^l \in \mathbb{R}^{|\mathbf{c}_t| \times d_h}$ denote the query and key matrices computed from the current context $\mathbf{c}_t = [\mathbf{x}; \mathbf{y}_{1:t-1}]$. Let $\mathbf{q}_{t,h}^l$ be the query vector at the last position of $\mathbf{c}_t$, used to predict y_t, and let $\mathbf{k}_{j,h}^l$ be the key vector at context position j. The causal key set $\mathcal{C}_t = \{1, \ldots, |\mathbf{c}_t|\}$ indexes all prompt tokens and previously generated tokens available before y_t is produced. The attention weight between position t and any preceding position j ($j < t$) at layer l is computed as

$$a_{t,h}^l(j) = \frac{\exp\left((\mathbf{q}_{t,h}^l)^\top \mathbf{k}_{j,h}^l/\sqrt{d_h}\right)}{\sum_{j' \in \mathcal{C}_t} \exp\left((\mathbf{q}_{t,h}^l)^\top \mathbf{k}_{j',h}^l/\sqrt{d_h}\right)}. \tag{8}$$

To reduce memory cost on long responses, we compute the loss only for sampled decoding steps t and sampled causal keys $\tilde{\mathcal{C}}_t \subseteq \mathcal{C}_t$. The sampled key set combines a recent local window with strided global positions, so it covers both nearby reasoning tokens and distant prompt evidence. We renormalize attention over this sampled set:

$$\tilde{a}_{t,h}^l(j) = \frac{a_{t,h}^l(j)}{\sum_{k \in \tilde{\mathcal{C}}_t} a_{t,h}^l(k)}, \qquad j \in \tilde{\mathcal{C}}_t. \tag{9}$$

Our attention-alignment loss at layer l is defined as

$$\mathcal{L}_{\text{attn}}^{(l)} = \mathbb{E}_t\left[A_t \frac{1}{H}\sum_{h=1}^{H} \mathrm{JS}\left(\tilde{a}_{t,h}^l, \mathrm{sg}\left(\tilde{a}_{t,h}^L\right)\right)\right] \tag{10}$$

where H denotes the number of heads in transformers. Positive advantages encourage the shadow layer to follow the final layer's evidence use, while negative advantages reduce agreement with attention patterns with low-reward continuations.

3.3 The OISD Training Objective

We adopt GRPO (Shao et al., 2024) as the base training signal. For each GRPO rollout batch $\mathcal{B} = \{(\mathbf{x}_i, \mathbf{y}_i, A_{1:T_i}^i)\}_{i=1}^G$, OISD computes both the final-layer RL loss and the internal student-teacher losses on the same sampled trajectories:

$$\begin{aligned} \mathcal{L}_{\text{OISD}}(\mathcal{B}) = \mathcal{L}_{\text{GRPO}}(\mathcal{B}) &+ \lambda_{\text{think}} \frac{1}{G}\sum_{g=1}^{G} \mathcal{L}_{\text{think}}^{(l,g)} \\ &+ \lambda_{\text{attn}} \frac{1}{G}\sum_{g=1}^{G} \mathcal{L}_{\text{attn}}^{(l,g)}. \end{aligned} \tag{11}$$

Here, $\mathcal{L}_{\text{GRPO}}(\mathcal{B})$ is the final-layer loss of GRPO. The auxiliary losses $\mathcal{L}_{\text{think}}^{(l,g)}$ and $\mathcal{L}_{\text{attn}}^{(l,g)}$ are computed on the same g-th GRPO rollout. The coefficients λ_{think} and λ_{attn} control the strengths of belief and attention transfer. We set $\lambda_{\text{think}} = 1$ and $\lambda_{\text{attn}} = 0.1$. For layer selection l, we follow (Tan et al., 2025) and choose the last FFN layer with a positive exploration signal. Details are provided in the experiment section and the appendix.

4 Experiment

4.1 Experimental Setup

Models and Baselines. We evaluate OISD on Qwen3-4B, Qwen3-8B, OctoThinker-3B-Long-Base, and OctoThinker-8B-Long-Base. Qwen3-4B (Yang et al., 2025a) is trained on `deepmath-5k`

Table 1: Avg@K results on AMC23, MATH500, AIME24 and AIME25.

Methods	AMC23 (Avg@16)	MATH500 (Avg@16)	AIME24 (Avg@32)	AIME25 (Avg@32)	Avg.
Qwen3-4B					
Vanilla	67.66	80.29	23.20	18.60	47.44
PPO	77.03	83.64	32.60	27.60	55.22
Reinforce++	63.44	80.63	17.40	18.65	45.03
RLOO	77.66	82.73	30.83	24.79	54.00
GRPO	76.88	82.41	32.19	28.85	55.08
BuPO	81.09	84.90	36.88	31.15	58.51
OISD (ours)	**84.38**	**90.99**	**47.29**	**36.35**	**64.75**
Qwen3-8B					
Vanilla	67.34	80.46	26.98	19.17	48.49
GRPO	85.94	88.05	49.48	33.54	64.23
BuPO	89.22	87.76	**54.06**	34.38	66.36
OISD (ours)	**90.62**	**89.38**	52.40	**34.48**	**66.72**
OctoThinker-3B-Long-Base					
Vanilla	1.24	5.26	0.21	0.00	1.68
GRPO	27.50	46.07	0.63	0.10	18.58
BuPO	27.50	**49.79**	0.63	0.42	19.59
OISD (ours)	**27.81**	49.41	**1.98**	0.42	**19.91**
OctoThinker-8B-Long-Base					
Vanilla	4.53	9.84	0.52	0.10	3.75
GRPO	34.84	56.89	2.50	2.19	24.11
BuPO	37.66	62.05	4.69	6.77	27.79
OISD (ours)	**39.38**	**66.19**	**7.29**	**7.50**	**30.09**

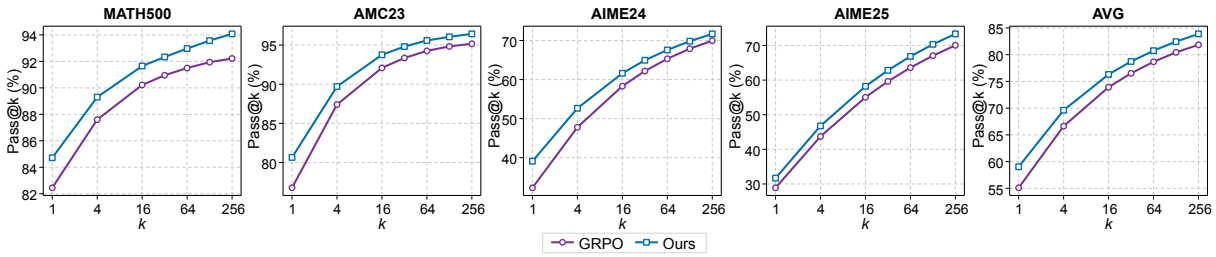

Figure 3: Pass@k comparison between GRPO and OISD using Qwen3-4B.

under the same setting as others. Qwen3-8B tests scaling within the Qwen3 family, while the OctoThinker models evaluate transfer to a different model family. We compare OISD against the corresponding Vanilla, PPO, Reinforce++, RLOO, GRPO, and BuPO (Tan et al., 2025). GRPO serves as the final-layer-only RL baseline, while BuPO represents direct internal-policy optimization. For layer selection l for both $\mathcal{L}^{l}_{\text{think}}$ and $\mathcal{L}^{l}_{\text{attn}}$, we follow (Tan et al., 2025), setting $l = 6$ for Qwen3-4B and Qwen3-8B, $l = 27$ for Llama-OctoThinker-3B-Base, and $l = 31$ for Llama-OctoThinker-8B-Base.

Evaluation benchmarks and metrics. The evaluation benchmarks include AMC23 (MAA, 2023), MATH500 (Lightman et al., 2024), AIME24, and AIME25 (MAA, 2024, 2025). We report Avg@16 for AMC23 and MATH500, Avg@32 for AIME24 and AIME25, and the mean across all four benchmarks. Avg@K denotes average accuracy over K sampled responses. We also report Pass@K, which measures whether at least one of the K sampled responses is correct under a fixed-sample evaluation protocol: $\text{Pass@K} := \mathbb{E}_{x\sim D}\left[1 - \frac{\binom{n-c}{K}}{\binom{n}{K}}\right]$, where c is the number of correct completions among n generated responses. To reduce evaluation variance, we use $n = 300$ following (Tan et al., 2025).

4.2 Main Results

Avg@K Performance. Table 1 reports the main Avg@k results on AMC23, MATH500, AIME24, and AIME25. Using the Qwen3-4B backbone,

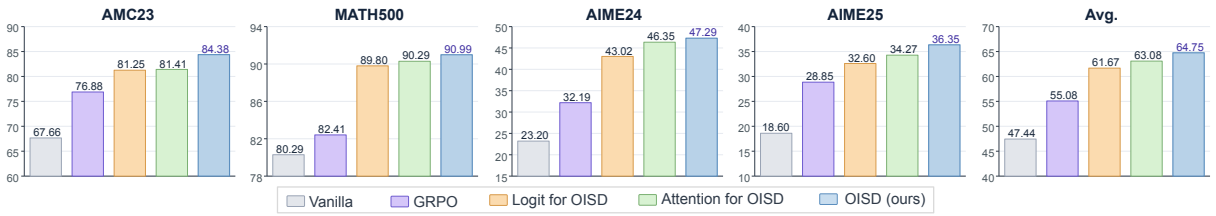

Figure 4: Effect of logit and attention alignment on Qwen3-4B. We compare Vanilla, GRPO, OISD with logit alignment only, OISD with attention alignment only, and the full OISD on AMC23, MATH500, AIME24, AIME25 with the overall average.

OISD achieves the best average score of 64.75, outperforming GRPO by 9.67 points and BuPO by 6.24 points. The gains are especially strong on AIME24 and AIME25, with improvements of 14.31 and 9.21 points over GRPO, respectively. AMC23 and MATH500 also improve from 76.88 and 82.41 to 84.38 and 90.99. At the larger Qwen3-8B scale, OISD attains the best overall average of 66.72 and remains competitive on the more challenging AIME benchmarks.

Pass@K Performance. Figure 3 demonstrates that increasing K improves the pass rate for both GRPO and OISD. OISD consistently achieves higher average Pass@K across all evaluated settings, suggesting that OISD improves not only average accuracy but also the overall quality of the sampled solution set.

4.3 Ablation Studies

Effect of Logit and Attention Alignment. We further ablate the effects of $\mathcal{L}_{\text{think}}^{(l)}$ and $\mathcal{L}_{\text{attn}}^{(l)}$ by training three variants: OISD with logit alignment only, OISD with attention alignment only, and the full OISD. Figure 4 shows that all are effective. Logit alignment improves the macro-average from 55.08 to 61.67, with the largest gain on AIME24 (32.19 $\rightarrow$ 43.02), while attention alignment performs slightly better overall, reaching 63.08 on average and 46.35 on AIME24. Combining both yields the best performance, achieving the macro-average of 64.75. These results suggest that predictive logit alignment and evidence-routing attention alignment provide complementary supervision.

Comparison between OISD and OPSD. We also compare OISD with the representative OPSD on-policy self-distillation framework (Zhao et al., 2026). OPSD follows the training setting of OISD. Unlike OISD, OPSD has access to privileged ground-truth solutions during training, whereas our method requires no external solution supervision.

Table 2 compares OISD with OPSD on Qwen3-4B. While OPSD improves reasoning through context-based teacher–student views and already surpasses GRPO, OISD instead performs fully internal on-policy supervision by directly distilling the model's own final-layer logits and attention from the same rollout. Under the same Avg@K evaluation protocol, OPSD achieves an average score of 56.63, whereas OISD reaches 64.75. OISD consistently outperforms OPSD.

Table 2: *Avg@K* results comparison between GRPO, OPSD (Zhao et al., 2026), and OISD on Qwen3-4B. AMC23 and MATH500 report Avg@16, while AIME24 and AIME25 report Avg@32.

Method	AMC23	MATH500	AIME24	AIME25	Avg.
GRPO	76.88	82.41	32.19	28.85	55.08
OPSD	77.19	86.64	37.08	25.62	56.63
OISD (ours)	**84.38**	**90.99**	**47.29**	**36.35**	**64.75**

4.4 Analysis

Training Dynamics. Figure 5 summarizes the training dynamics of OISD with the Qwen3-4B backbone. We track reward, layer-6 entropy, and response length to analyze the effect of OISD. The reward steadily improves during training, while the entropy increases during the early stage of training, suggesting broader policy exploration and a larger reasoning search space induced by OISD. The response length grows from roughly 1300 to 1600 tokens before stabilizing with fluctuations.

Qualitative Attention Visualization. We explore how attention alignment shapes intermediate-layer reasoning in Figure 6. In Figure 6(a), we use a fixed reasoning example and mark the words that provide critical evidence for the final answer. For layer 6 and the final layer, we average causal attention from selected late-reasoning query positions over heads, merge subwords into word-level scores, and normalize within each layer. Darker color highlights indicate stronger attention to the marked evidence words. The final layer places stronger attention on

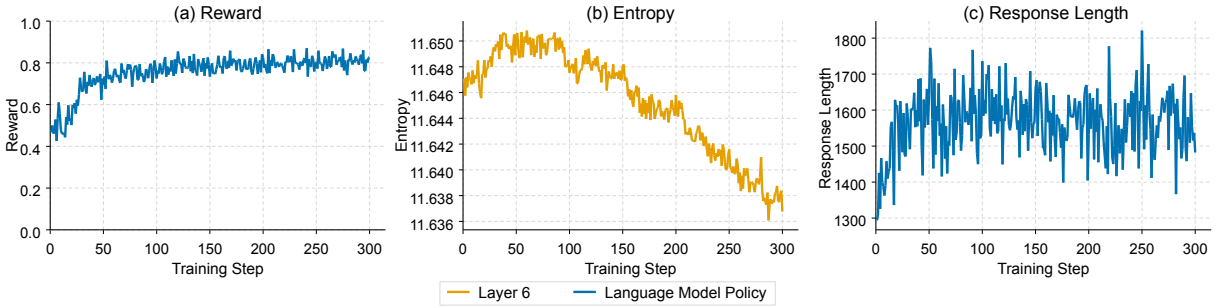

Figure 5: Training dynamics of OISD over the full RL run on Qwen3-4B. (a) Average rollout reward of the final-layer acting policy increases steadily across training. (b) Response-token entropy of the middle-layer readout (layer 6). (c) The response length of generated trajectories grows as the policy develops longer reasoning chains.

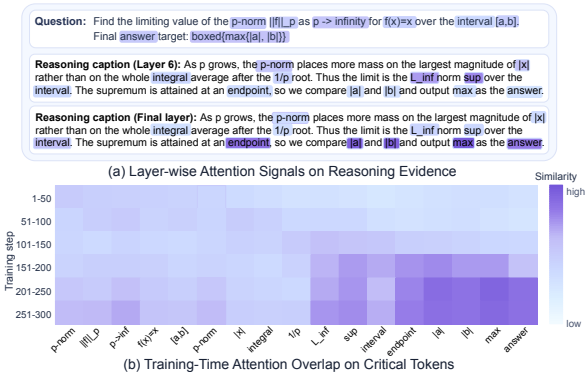

Figure 6: Attention visualization during OISD training. (a) Attention highlights from the intermediate and final layers on the same reasoning trajectory, where highlighted tokens support the final answer. (b) Similarity between intermediate- and final-layer attention on critical tokens across training blocks, with darker colors indicating stronger alignment (higher similarity).

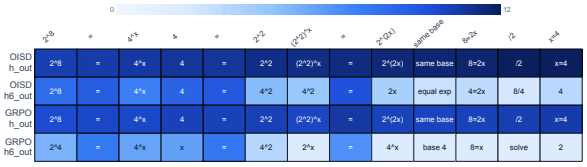

Figure 7: Logit-lens comparison between OISD and GRPO on a reasoning trace. Each row shows the top token decoded from the final or layer-6 hidden state. Bold outlines indicate agreement between layer-6 and final-layer predictions. OISD exhibits earlier alignment on reasoning-critical tokens, whereas GRPO produces more local or incomplete intermediate predictions.

answer-supporting evidence, including the limiting norm, supremum, endpoint comparison, and final maximum operation, while layer 6 shows a broader but partially overlapping pattern.

Figure 6(b) tracks this alignment during training. We compute attention Jensen–Shannon divergence between layer 6 and the final layer and convert it into a normalized agreement score, where higher values indicate stronger overlap. Columns correspond to the critical-token groups in Figure 6(a). Darker colors in later stages indicate that attention alignment gradually moves the intermediate layer toward the final layer's evidence-routing pattern.

Qualitative Logit-Lens Visualization. Figure 7 presents a logit-lens comparison between GRPO and OISD on a representative reasoning trace using the Qwen3-4B backbone. Each row shows the top decoded token from either the final layer or the intermediate layer-6 hidden state, while bold outlines denote agreement between layer-6 and final-layer predictions. OISD exhibits substantially earlier alignment between intermediate and final predictions on reasoning-critical tokens, indicating that the intermediate layer develops more globally consistent reasoning behaviors during generation. In contrast, GRPO often produces more local, fragmented, or incomplete intermediate predictions.

5 Conclusion

We introduce *on-policy internal self-distillation*, a new paradigm for reasoning RL that leverages the model's own internal computation as supervision while preserving a single acting policy for rollout and optimization. Building on this idea, we propose OISD, which distills the model's final-layer computation into an intermediate layer through two complementary signals: logit alignment for *how to think* and attention alignment for *where to look*. OISD preserves policy consistency and provides persistent internal supervision for learning improved intermediate representations. Experiments on Qwen3-4B demonstrate substantial gains over strong reasoning RL baselines, highlighting on-policy internal self-distillation as a promising direction for reasoning-oriented LLM post-training.

Limitations

Limited computational resources constrained our exploration of broader model scales and architectures. Nevertheless, this work provides a promising perspective on on-policy internal self-distillation for LLM reasoning. Our experiments mainly focus on mathematical reasoning with verifiable answers, and future work should investigate whether the proposed framework generalizes to more complex reasoning tasks requiring longer reasoning chains or different forms of supervision. In addition, we do not conduct an in-depth study of layer selection. Extending internal supervision to multiple layers could be an interesting direction for future research.

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, volume 2024, pages 21246–21263.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. 2023. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28.

Yao Chen, Jiawei Sheng, Wenyuan Zhang, and Tingwen Liu. 2025. Improving reasoning capabilities in small models through mixture-of-layers distillation with stepwise attention on key information. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4952–4971.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R Glass, and Pengcheng He. 2024. Dola: Decoding by contrasting layers improves factuality in large language models. In *International Conference on Learning Representations*, volume 2024, pages 54158–54183.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Knowledge neurons in pretrained transformers. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8493–8502.

Mostafa Elhoushi, Akshat Shrivastava, Diana Liskovich, Basil Hosmer, Bram Wasti, Liangzhen Lai, Anas Mahmoud, Bilge Acun, Saurabh Agarwal, Ahmed Roman, and 1 others. 2024. Layerskip: Enabling early exit inference and self-speculative decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12622–12642.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Akshat Gupta, Jay Yeung, Gopala Anumanchipalli, and Anna Ivanova. 2025. How do llms use their depth? *arXiv preprint arXiv:2510.18871*.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and 1 others. 2026. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*.

Shashank Kapadia, Deep Naryan Mishra, Sujal Reddy Alugubelli, Haoan Wang, Saipraveen Vabbilisetty, Rishi Bhatia, and Anupriya Sharma. 2026. Leap: Layer-wise exit-aware pretraining for efficient transformer inference. *arXiv preprint arXiv:2605.01058*.

Chen-Yu Lee, Saining Xie, Patrick Gallagher, Zhengyou Zhang, and Zhuowen Tu. 2015. Deeply-supervised nets. In *Artificial intelligence and statistics*, pages 562–570. Pmlr.

Bangzheng Li, Jianmo Ni, Chen Qu, Ian Miao, Liu Yang, Xingyu Fu, Muhao Chen, and Derek Zhiyuan Cheng. 2026. Reinforced attention learning. *arXiv preprint arXiv:2602.04884*.

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

MAA. 2023. American mathematics contest 12 (amc 12).

MAA. 2024. American invitational mathematics examination (aime).

MAA. 2025. American invitational mathematics examination (aime).

María Luisa Menéndez, Julio Angel Pardo, Leandro Pardo, and María del C Pardo. 1997. The jensen-shannon divergence. *Journal of the Franklin Institute*, 334(2):307–318.

nostalgebraist. 2020. Interpreting gpt: The logit lens. https://www.lesswrong.com/posts/AcKRB8wDpdaN6v6ru/interpreting-gpt-the-logit-lens. Accessed: 2026-05-14.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

Yulei Qin, Gang Li, Zongyi Li, Zihan Xu, Yuchen Shi, Zhekai Lin, Xiao Cui, Ke Li, and Xing Sun. 2026. Incentivizing reasoning for advanced instruction-following of large language models. *Advances in Neural Information Processing Systems*, 38:108337–108401.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. 2025. Codi: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 677–693.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. 2026. Self-distillation enables continual learning. *arXiv preprint arXiv:2601.19897*.

Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. 2025. Prmbench: A fine-grained and challenging benchmark for process-level reward models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25299–25346.

Yuqiao Tan, Minzheng Wang, Shizhu He, Huanxuan Liao, Chengfeng Zhao, Qiunan Lu, Tian Liang, Jun Zhao, and Kang Liu. 2025. Bottom-up policy optimization: Your language model policy secretly contains internal policies. *arXiv preprint arXiv:2512.19673*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Wenda Xu, Rujun Han, Zifeng Wang, Long Le, Dhruv Madeka, Lei Li, William Wang, Rishabh Agarwal, Chen-Yu Lee, and Tomas Pfister. 2025. Speculative knowledge distillation: Bridging the teacher-student gap through interleaved sampling. In *International Conference on Learning Representations*, volume 2025, pages 64616–64646.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yiran Yang, Haifeng Sun, Jingyu Wang, Qi Qi, Zirui Zhuang, Huazheng Wang, Pengfei Ren, Jing Wang, and Jianxin Liao. 2025b. Unveiling internal reasoning modes in llms: A deep dive into latent reasoning vs. factual shortcuts with attribute rate ratio. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 2186–2206.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Tianzhu Ye, Li Dong, Xun Wu, Shaohan Huang, and Furu Wei. 2026. On-policy context distillation for language models. *arXiv preprint arXiv:2602.12275*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2026. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244.

Honglin Zhang, Qianyue Hao, Fengli Xu, and Yong Li. 2025. Reinforcement learning fine-tuning enhances activation intensity and diversity in the internal circuitry of llms. *arXiv preprint arXiv:2509.21044*.

Siyan Zhao, Zhihui Xie, Mengchen Liu, Jing Huang, Guan Pang, Feiyu Chen, and Aditya Grover. 2026. Self-distilled reasoner: On-policy self-distillation for large language models. *arXiv preprint arXiv:2601.18734*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025a. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025b. Processbench: Identifying process errors in mathematical reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1009–1024.

Ruijie Zhu, Tianhao Peng, Tianhao Cheng, Xingwei Qu, Jinfa Huang, Dawei Zhu, Hao Wang, Kaiwen Xue,

Xuanliang Zhang, Yong Shan, Tianle Cai, Taylor Kergan, Assel Kembay, Andrew Smith, Chenghua Lin, Binh Nguyen, Yuqi Pan, Yuhong Chou, Zefan Cai, and 14 others. 2025. A survey on latent reasoning. *arXiv preprint arXiv:2507.06203*.

A Appendix

A.1 Training Dynamics

As shown in Figure 8, Qwen3-4B OISD steadily improves as training progresses. Performance increases across most benchmarks from early to later checkpoints, leading to a higher overall average. In the late stage, the results stabilize within a narrow range.

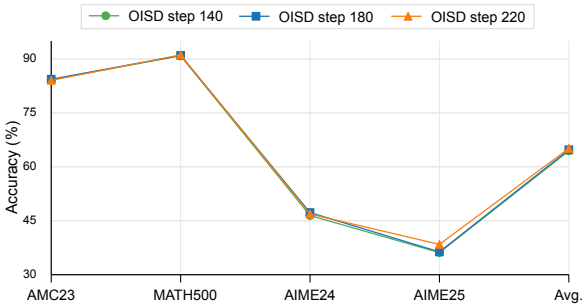

Figure 8: Checkpoint progression for Qwen3-4B OISD with logit and attention alignment. Results are reported with Avg@16 on AMC23 and MATH500, Avg@32 on AIME24 and AIME25, and the four-benchmark average.

A.2 Training Details

All experiments are conducted on four NVIDIA A100 or H200 GPUs. Since reinforcement fine-tuning for long-form reasoning is computationally intensive, we utilize FlashAttention, FSDP-2 sharding, gradient checkpointing, dynamic batching by token length, and vLLM rollouts with chunked prefill to improve training throughput and memory efficiency. The implementation is built with PyTorch 2.6.0+cu124, Transformers 4.56.0, vLLM 0.8.5, Ray 2.47.1, and PEFT 0.18.1.

To ensure consistency and comparability across experiments, we adopt the same on-policy training pipeline for GRPO, OPSD and OISD. We follow the (Tan et al., 2025) for all shared settings, including the `deepmath-5k` training subset, a prompt batch size of 128, and 8 rollouts per prompt. We use AdamW with a learning rate of 1×10^{-6}, the GRPO advantage estimator, token-mean loss aggregation, and a symmetric policy clipping window of $\epsilon = 0.2$.

For OISD, the rollout policy is always read from the final Transformer layer, while the shadow student is fixed at layer $l = 6$. We set the two internal loss weights to λ_{think}=1.0 and λ_{attn}=0.1.

A.3 Gradients of the Think and Attention Losses

A.3.1 Gradients of the Think loss

Recall that, with temperature τ, the layer-l shadow student and the final-layer teacher distributions are

$$\begin{aligned}
\mathbf{z}_t^l &= \mathrm{LN}(\mathbf{h}_t^l)\mathbf{E}_{\mathrm{u}}^\top, \\
\mathbf{z}_t^L &= \mathrm{LN}(\mathbf{h}_t^L)\mathbf{E}_{\mathrm{u}}^\top, \\
\mathbf{p}_t^{l,\tau} &= \mathrm{Softmax}(\mathbf{z}_t^l/\tau), \\
\mathbf{q}_t^{L,\tau} &= \mathrm{sg}\left[\mathrm{Softmax}(\mathbf{z}_t^L/\tau)\right],
\end{aligned}$$

where $\mathrm{sg}(\cdot)$ denotes stop-gradient on the detached final-layer teacher. Define the mixture distribution

$$\mathbf{m}_t^{l,L,\tau} = \frac{1}{2}\left(\mathbf{p}_t^{l,\tau} + \mathbf{q}_t^{L,\tau}\right).$$

The token-level Jensen–Shannon divergence is

$$\begin{aligned}
\mathrm{JS}\left(\mathbf{p}_t^{l,\tau}, \mathbf{q}_t^{L,\tau}\right) &= \frac{1}{2}\mathrm{KL}\left(\mathbf{p}_t^{l,\tau} \middle\| \mathbf{m}_t^{l,L,\tau}\right) \\
&\quad + \frac{1}{2}\mathrm{KL}\left(\mathbf{q}_t^{L,\tau} \middle\| \mathbf{m}_t^{l,L,\tau}\right).
\end{aligned}$$

Expanding over vocabulary index i gives

$$\mathrm{JS} = \frac{1}{2}\sum_i p_i \log\frac{p_i}{m_i} + \frac{1}{2}\sum_i q_i \log\frac{q_i}{m_i},$$

where $p_i = \mathbf{p}_{t,i}^{l,\tau}$, $q_i = \mathbf{q}_{t,i}^{L,\tau}$, and $m_i = \mathbf{m}_{t,i}^{l,L,\tau}$.

Since the teacher distribution is detached via the stop-gradient and

$$m_i = \frac{1}{2}(p_i + q_i), \qquad \frac{\partial m_i}{\partial p_i} = \frac{1}{2},$$

the derivative with respect to p_i becomes

$$\begin{aligned}
\frac{\partial\,\mathrm{JS}}{\partial p_i} &= \frac{1}{2}\left(\log\frac{p_i}{m_i} + 1 - \frac{p_i}{2m_i}\right) - \frac{q_i}{4m_i} \\
&= \frac{1}{2}\log\frac{p_i}{m_i} + \frac{1}{2} - \frac{p_i + q_i}{4m_i}.
\end{aligned}$$

Using $m_i = \frac{1}{2}(p_i + q_i)$, the last two terms cancel, yielding

$$\boxed{\frac{\partial\,\mathrm{JS}}{\partial p_i} = \frac{1}{2}\log\frac{p_i}{m_i}}.$$

Define

$$\mathbf{g}_t = \frac{1}{2}\log\frac{\mathbf{p}_t^{l,\tau}}{\mathbf{m}_t^{l,L,\tau}}.$$

Using the chain rule, we have

$$\frac{\partial \mathcal{L}_{\text{think}}^{(l)}}{\partial \mathbf{z}_t^l} = A_t \left(\frac{\partial \mathbf{p}_t^{l,\tau}}{\partial \mathbf{z}_t^l} \right)^\top \frac{\partial \, \text{JS}}{\partial \mathbf{p}_t^{l,\tau}}.$$

From the previous derivation,

$$\frac{\partial \, \text{JS}}{\partial \mathbf{p}_t^{l,\tau}} = \mathbf{g}_t, \qquad \mathbf{g}_t = \frac{1}{2} \log \frac{\mathbf{p}_t^{l,\tau}}{\mathbf{m}_t^{l,L,\tau}}.$$

The softmax Jacobian is

$$\frac{\partial \mathbf{p}_t^{l,\tau}}{\partial \mathbf{z}_t^l} = \frac{1}{\tau} \left[\text{diag}(\mathbf{p}_t^{l,\tau}) - \mathbf{p}_t^{l,\tau} (\mathbf{p}_t^{l,\tau})^\top \right].$$

Applying the chain rule together with the softmax Jacobian gives

$$\begin{aligned} \frac{\partial \mathcal{L}_{\text{think}}^{(l)}}{\partial \mathbf{z}_t^l} &= A_t \left(\frac{\partial \mathbf{p}_t^{l,\tau}}{\partial \mathbf{z}_t^l} \right)^\top \frac{\partial \, \text{JS}}{\partial \mathbf{p}_t^{l,\tau}} \\ &= \frac{A_t}{\tau} \left[\text{diag}(\mathbf{p}_t^{l,\tau}) - \mathbf{p}_t^{l,\tau} (\mathbf{p}_t^{l,\tau})^\top \right] \mathbf{g}_t, \end{aligned}$$

where

$$\mathbf{g}_t = \frac{1}{2} \log \frac{\mathbf{p}_t^{l,\tau}}{\mathbf{m}_t^{l,L,\tau}}.$$

Expanding the matrix-vector product yields

$$\frac{\partial \mathcal{L}_{\text{think}}^{(l)}}{\partial \mathbf{z}_t^l} = \frac{A_t}{\tau} \left[\text{diag}(\mathbf{p}_t^{l,\tau}) \mathbf{g}_t - \mathbf{p}_t^{l,\tau} (\mathbf{p}_t^{l,\tau})^\top \mathbf{g}_t \right].$$

For the first term,

$$\text{diag}(\mathbf{p}_t^{l,\tau}) \mathbf{g}_t = \mathbf{p}_t^{l,\tau} \odot \mathbf{g}_t,$$

while for the second term,

$$(\mathbf{p}_t^{l,\tau})^\top \mathbf{g}_t = \left\langle \mathbf{p}_t^{l,\tau}, \mathbf{g}_t \right\rangle$$

is a scalar, giving

$$\mathbf{p}_t^{l,\tau} (\mathbf{p}_t^{l,\tau})^\top \mathbf{g}_t = \mathbf{p}_t^{l,\tau} \left\langle \mathbf{p}_t^{l,\tau}, \mathbf{g}_t \right\rangle.$$

Therefore,

$$\frac{\partial \mathcal{L}_{\text{think}}^{(l)}}{\partial \mathbf{z}_t^l} = \frac{A_t}{\tau} \left[\mathbf{p}_t^{l,\tau} \odot \mathbf{g}_t - \mathbf{p}_t^{l,\tau} \left\langle \mathbf{p}_t^{l,\tau}, \mathbf{g}_t \right\rangle \right].$$

Since the inner product term is scalar-valued,

$$\mathbf{p}_t^{l,\tau} \left\langle \mathbf{p}_t^{l,\tau}, \mathbf{g}_t \right\rangle = \mathbf{p}_t^{l,\tau} \odot \left(\left\langle \mathbf{p}_t^{l,\tau}, \mathbf{g}_t \right\rangle \mathbf{1} \right),$$

Thus,

$$\boxed{\frac{\partial \mathcal{L}_{\text{think}}^{(l)}}{\partial \mathbf{z}_t^l} = \frac{A_t}{\tau} \left[\mathbf{p}_t^{l,\tau} \odot \left(\mathbf{g}_t - \left\langle \mathbf{p}_t^{l,\tau}, \mathbf{g}_t \right\rangle \mathbf{1} \right) \right]}.$$

Since

$$\mathbf{z}_t^l = \text{LN}(\mathbf{h}_t^l) \mathbf{E}_{\text{u}}^\top,$$

the chain rule gives

$$\boxed{\frac{\partial \mathcal{L}_{\text{think}}^{(l)}}{\partial \mathbf{h}_t^l} = J_{\text{LN}}(\mathbf{h}_t^l)^\top \mathbf{E}_{\text{u}} \frac{\partial \mathcal{L}_{\text{think}}^{(l)}}{\partial \mathbf{z}_t^l}}$$

where $J_{\text{LN}}(\mathbf{h}_t^l)$ denotes the Jacobian of layer normalization. The clipped advantage

$$A_t = \text{clip}(A_t, -c, c)$$

acts as a signed scaling factor: positive-advantage tokens pull the shadow student toward the detached final-layer teacher, while negative-advantage tokens reverse the gradient direction to suppress poor reasoning trajectories.

A.3.2 Gradients of Attention Loss

Let

$$\mathbf{p}_{t,h}^l = \tilde{a}_{t,h}^l, \qquad \mathbf{p}_{t,h}^L = \text{sg}(\tilde{a}_{t,h}^L),$$

where $\mathbf{p}_{t,h}^l$ denotes the sampled attention distribution of the shadow layer and $\mathbf{p}_{t,h}^L$ denotes the detached final-layer teacher distribution over the sampled causal set $\tilde{\mathcal{C}}_t$.

For decoding step t and attention head h, the attention-alignment loss is

$$\mathcal{L}_{\text{attn}}^{(l)} = \frac{A_t}{H} \, \text{JS} \left(\mathbf{p}_{t,h}^l, \mathbf{p}_{t,h}^L \right).$$

Define the mixture distribution

$$\mathbf{m}_{t,h} = \frac{1}{2} \left(\mathbf{p}_{t,h}^l + \mathbf{p}_{t,h}^L \right).$$

Using the same JS-divergence derivation as before,

$$\frac{\partial \, \text{JS}(\mathbf{p}_{t,h}^l, \mathbf{p}_{t,h}^L)}{\partial \mathbf{p}_{t,h}^l(j)} = \frac{1}{2} \log \frac{\mathbf{p}_{t,h}^l(j)}{\mathbf{m}_{t,h}(j)}.$$

Therefore,

$$\frac{\partial \mathcal{L}_{\text{attn}}^{(l)}}{\partial \mathbf{p}_{t,h}^l(j)} = \frac{A_t}{2H} \log \frac{\mathbf{p}_{t,h}^l(j)}{\mathbf{m}_{t,h}(j)}.$$

Let the sampled attention logits at layer l be

$$s_{t,h}^l(j) = \frac{(\mathbf{q}_{t,h}^l)^\top \mathbf{k}_{j,h}^l}{\sqrt{d_h}}, \qquad j \in \tilde{\mathcal{C}}_t,$$

with

$$p_{t,h}^l(j) = \text{Softmax}(\mathbf{s}_{t,h}^l)_j.$$

Using the softmax Jacobian,

$$\frac{\partial p^l_{t,h}(j)}{\partial s^l_{t,h}(i)} = p^l_{t,h}(j)\left(\mathbf{1}_{i=j} - p^l_{t,h}(i)\right),$$

the gradient with respect to the sampled attention logits becomes

$$\frac{\partial \mathcal{L}^{(l)}_{\text{attn}}}{\partial s^l_{t,h}(i)} = A_t\, p^l_{t,h}(i)\left(g_{t,h}(i) - \sum_{j\in\tilde{\mathcal{C}}_t} p^l_{t,h}(j) g_{t,h}(j)\right)$$

where

$$g_{t,h}(i) = \frac{1}{2H}\log\frac{p^l_{t,h}(i)}{m_{t,h}(i)}.$$

Equivalently, in vector form,

$$\boxed{\frac{\partial \mathcal{L}^{(l)}_{\text{attn}}}{\partial \mathbf{s}^l_{t,h}} = A_t\, \mathbf{p}^l_{t,h} \odot \left(\mathbf{g}_{t,h} - \langle \mathbf{p}^l_{t,h}, \mathbf{g}_{t,h}\rangle \mathbf{1}\right)}$$

with

$$\mathbf{g}_{t,h} = \frac{1}{2H}\log\frac{\mathbf{p}^l_{t,h}}{\mathbf{m}_{t,h}}.$$

Since the sampled attention logits are defined as

$$s^l_{t,h}(j) = \frac{(\mathbf{q}^l_{t,h})^\top \mathbf{k}^l_{j,h}}{\sqrt{d_h}}, \qquad j \in \tilde{\mathcal{C}}_t,$$

their partial derivatives are

$$\frac{\partial s^l_{t,h}(j)}{\partial \mathbf{q}^l_{t,h}} = \frac{\mathbf{k}^l_{j,h}}{\sqrt{d_h}}, \qquad \frac{\partial s^l_{t,h}(j)}{\partial \mathbf{k}^l_{j,h}} = \frac{\mathbf{q}^l_{t,h}}{\sqrt{d_h}}.$$

Applying the chain rule, the gradient with respect to the query vector becomes

$$\boxed{\frac{\partial \mathcal{L}^{(l)}_{\text{attn}}}{\partial \mathbf{q}^l_{t,h}} = \frac{1}{\sqrt{d_h}} \sum_{j\in\tilde{\mathcal{C}}_t} \frac{\partial \mathcal{L}^{(l)}_{\text{attn}}}{\partial s^l_{t,h}(j)} \mathbf{k}^l_{j,h}}$$

Similarly, for each sampled key position j,

$$\boxed{\frac{\partial \mathcal{L}^{(l)}_{\text{attn}}}{\partial \mathbf{k}^l_{j,h}} = \frac{1}{\sqrt{d_h}} \frac{\partial \mathcal{L}^{(l)}_{\text{attn}}}{\partial s^l_{t,h}(j)} \mathbf{q}^l_{t,h}}$$

The corresponding compact vector forms are

$$\boxed{\frac{\partial \mathcal{L}^{(l)}_{\text{attn}}}{\partial \mathbf{q}^l_{t,h}} = \frac{1}{\sqrt{d_h}} (\mathbf{K}^l_{\tilde{\mathcal{C}}_t,h})^\top \frac{\partial \mathcal{L}^{(l)}_{\text{attn}}}{\partial \mathbf{s}^l_{t,h}}}$$

and

$$\boxed{\frac{\partial \mathcal{L}^{(l)}_{\text{attn}}}{\partial \mathbf{K}^l_{\tilde{\mathcal{C}}_t,h}} = \frac{1}{\sqrt{d_h}} \frac{\partial \mathcal{L}^{(l)}_{\text{attn}}}{\partial \mathbf{s}^l_{t,h}} (\mathbf{q}^l_{t,h})^\top}$$

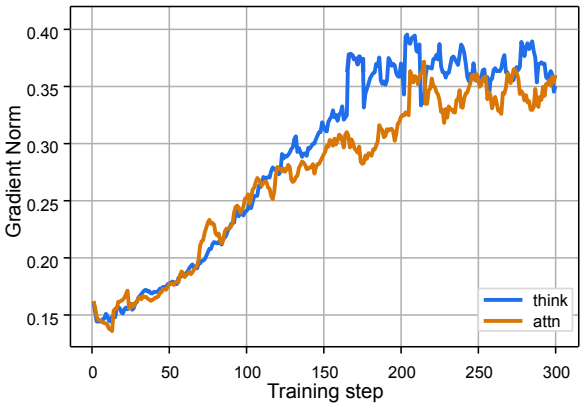

Figure 9: Gradient norm for the think and attention alignment losses.

A.3.3 Gradient Norm Visualization

Figure 9 shows the actor gradient norms induced by the two internal alignment losses. The think loss $\mathcal{L}_{\text{think}}$ transfers token-level beliefs from the final-layer teacher to the intermediate-layer student, while the attention loss $\mathcal{L}_{\text{attn}}$ transfers evidence-routing patterns over the sampled causal context.

As shown in Figure 9, both losses produce stable and progressively stronger gradient signals throughout training. The increasing gradient norm suggests that OISD gradually strengthens the interaction between intermediate and final-layer reasoning states. As RL training improves the final-layer policy, the resulting teacher–student mismatch provides increasingly informative internal supervision, leading to stronger latent-space supervision on intermediate representations.

A.4 Sample Prediction Comparison

For qualitative comparison, we inspect the reasoning traces of GRPO, OPSD, and OISD on the same AIME 2024 tetrahedron problem. The traces and per-problem statistics are taken from the corresponding AIME 2024 Avg@32 evaluation dump used in the main comparison.

Problem. *Let $ABCD$ be a tetrahedron such that $AB{=}CD{=}\sqrt{41}$, $AC{=}BD{=}\sqrt{80}$, and $BC{=}AD{=}\sqrt{89}$. There exists a point I inside the tetrahedron such that the distances from I to each of the faces of the tetrahedron are all equal. This distance can be written in the form $\frac{m\sqrt{n}}{p}$, where m, n, p are positive integers, m and p are relatively prime, and n is not divisible by the square of any prime. Find $m+n+p$. Ground truth:* 104.

Problem

Let ABCD be a tetrahedron such that AB=CD=sqrt(41), AC=BD=sqrt(80), and BC=AD=sqrt(89). There exists a point I inside the tetrahedron such that the distances from I to each face are all equal. This distance can be written as m sqrt(n)/p, where gcd(m,p)=1 and n is square-free. Findm+n+p.
Ground truth: 104.

OISD (ours) response

The rollout recognizes the disphenoid structure and embeds the tetrahedron in a rectangular box with x^2+y^2=41, x^2+z^2=80, and y^2+z^2=89, giving (x,y,z)=(4,5,8). It places A=(0,0,0), B=(4,5,0), C=(4,0,8), D=(0,5,8), and uses the scalar triple product AB dot (AC x AD)=-320 to obtain V=|-320|/6=160/3. A single face has ||AB x AC||=sqrt(3024)=12 sqrt(21), so the total surface area is 24 sqrt(21). Therefore r=3V/S=20 sqrt(21)/63, and m+n+p=20+21+63=104.
Final answer: 104.

GRPO response

The rollout starts from free coordinates A=(0,0,0), B=(sqrt(41),0,0), C=(p,q,r), and D=(s,t,u). It correctly derives p=16/sqrt(41), s=25/sqrt(41), q^2+r^2=t^2+u^2=3024/41, and qt+ru=1112/41. However, after reaching this underconstrained coordinate system, it searches over unsupported assumptions such as maybe C and D are symmetric or maybe CD is perpendicular to AB. The reasoning budget is exhausted before the system is reduced to a closed-form inradius.
Final answer: 0.

OPSD response

The rollout computes the scalar triple product route and obtains V=160/3, but it also applies a disphenoid formula route and obtains V=80 sqrt(2)/3. It notices the contradiction and says that one of the approaches must be wrong, yet it does not stabilize which evidence to trust. Instead, it switches back and forth between the two volume computations until the token budget is exhausted. Other incorrect samples terminate at intermediate quantities such as 3024, 6400, or 41.
Final answer: 1.

Figure 10: Reasoning comparison on the same AIME 2024 tetrahedron problem. The top box states the problem and ground truth; each column shows the first sampled rollout for one method together with its Avg@32 pass rate on this problem.

As shown in Figure 10, OISD's first rollout closes the reasoning chain by recognizing the disphenoid structure, computing $V{=}\frac{160}{3}$ and $r{=}\frac{20\sqrt{21}}{63}$, and returning 104. By contrast, the GRPO rollout derives several correct coordinate constraints but then searches over unsupported symmetry assumptions and terminates with 0. OPSD obtains the useful scalar-triple-product volume but also produces a conflicting formula-based volume, repeatedly revisits the disagreement, and finally emits the answer 1.

A.5 Additional Ablations

Loss weight λ_{think} in the think-only setting. The logit-alignment weight λ_{think} controls the strength of the internal teacher signal in the think-only setting. A smaller value weakens the pressure from the final-layer distribution, while a larger value makes the shadow layer follow the teacher more aggressively. We set $\lambda_{\text{think}} \in \{0.5, 1.0, 2.0\}$ with the attention branch disabled and report the comparison in Table 3.

Table 3: Sensitivity of the think-only (logit-only) OISD variant to λ_{think} on Qwen3-4B. AMC23 and MATH500 report Avg@16, AIME24 and AIME25 report Avg@32.

λ_{think}	AMC23	MATH500	AIME24	AIME25	Avg.
0.5	82.50	87.39	41.25	30.52	60.41
1.0	81.25	89.80	43.02	32.60	61.67
2.0	81.09	86.36	40.83	31.56	59.96

With $\lambda_{\text{think}}{=}0.5$, the auxiliary logit alignment already improves the internal-student variant, but the signal appears insufficient to fully align the intermediate layer with the final-layer teacher. Increasing the weight to 1.0 gives the best average performance, making the think loss provides useful internal supervision while remaining compatible with the on-policy GRPO update.

When the weight is increased to 2.0, the auxiliary objective becomes too strong and performance drops. This indicates that logit alignment

should guide the intermediate layer without competing with the on-policy update. We therefore use $\lambda_{\text{think}}=1.0$ as the default think-loss weight in the main experiments. We choose the attention-loss weight by the same criterion and set $\lambda_{\text{attn}}=0.1$.

A.6 Information About Use of AI Assistants

ChatGPT was used for language editing, manuscript polishing, and assistance with part of the equation derivations of JS. All outputs were reviewed and verified by the authors.