TEACH YOURSELF WHERE TO LOOK: ON-POLICY ATTENTION SELF-DISTILLATION FOR REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy self-distillation trains reasoning models on their own trajectories using dense token distribution guidance from a privileged teacher with access to a verified solution. This supervision transfers what the teacher predicts without directly transferring where it attends within the preceding context. We introduce On-Policy Attention Self-Distillation (OPASD), which complements token-level supervision with solution-conditioned attention distillation. Because the privileged teacher can attend to verified-solution tokens unavailable to the student, OPASD projects teacher attention onto student-visible positions and renormalizes the resulting distribution before alignment. Across three model sizes and four competition-level mathematics benchmarks, OPASD consistently outperforms token-only OPSD, improving average accuracy by 4.98 to 8.40 percentage points. OPASD also avoids the response-length inflation and performance degradation observed with token-only distillation, reducing generated rollout tokens by 73.9% and estimated model compute by 72.6% while training 1.53× faster. These results show that solution-conditioned attention provides a complementary supervision signal that makes on-policy self-distillation more accurate, stable, and compute-efficient.

1 INTRODUCTION

Improving language models' reasoning through post-training requires supervision that is informative and aligned with their own generated trajectories. Reinforcement learning with verifiable rewards (RLVR) learns from self-generated responses but typically provides a response-level reward with limited guidance for individual token decisions (Shao et al., 2024; Guo et al., 2025a; Yu et al., 2026). On-policy distillation (OPD) provides finer-grained supervision by having a teacher evaluate each prefix of a student-generated trajectory and provide a full next-token distribution (Agarwal et al., 2024). On-policy self-distillation (OPSD) further removes the need for a separately trained teacher by initializing both student and teacher from the same model (Zhao et al., 2026). The student generates a reasoning trajectory from the problem alone, while the teacher evaluates that trajectory with access to a verified solution. OPSD thus provides dense, solution-conditioned token-level supervision along the student's own reasoning process.

Despite this advantage, OPSD transfers privileged guidance only through next-token distributions. Such token-level supervision guides what the student predicts but does not directly guide where it allocates attention across the preceding context (Zhao et al., 2026; Yang et al., 2026b). This distinction matters in multi-step reasoning, where later steps often depend on constraints and intermediate results established in earlier steps (Chen et al., 2025; Guo et al., 2025b). The privileged teacher's attention distribution offers a distinct solution-conditioned signal that OPSD leaves unused. This raises the question: *How can on-policy self-distillation teach the student not only what to predict but also where to look?*

Unlike prior attention-alignment work on multimodal grounding or internal self-distillation (Li et al., 2026a; Liu et al., 2026), extending OPSD with attention-level supervision introduces a challenge absent from next-token distribution alignment. The teacher and student next-token distributions are directly comparable because both are defined over the same vocabulary. But attention distributions are defined over positions in their respective conditioning contexts. The student conditions on the problem and its generated prefix, while the privileged teacher additionally sees the verified solution. The teacher can therefore assign attention to solution tokens absent from the student's context. Direct

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

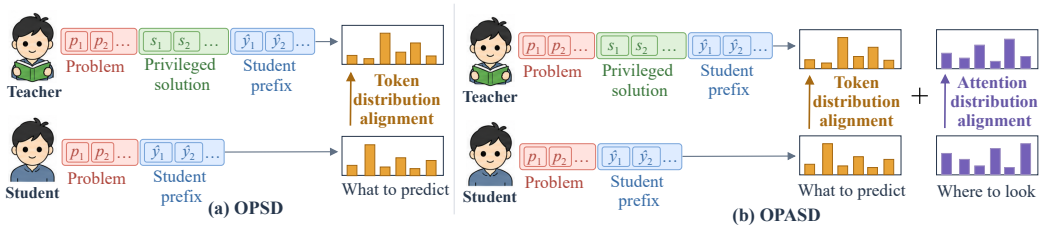

Figure 1: Comparison of OPSD and OPASD. (a) OPSD provides token-level supervision through teacher-student next-token distribution alignment. (b) OPASD provides both token- and attention-level supervision through joint next-token and projected attention distribution alignment.

alignment would require the student to match probability mass on positions it cannot observe, whereas simply discarding those positions leaves the teacher's remaining attention unnormalized. This creates a support mismatch between the teacher's solution-conditioned attention distribution and the student's attention distribution. An effective distillation target must preserve the teacher's solution-conditioned guidance while remaining fully defined over positions visible to the student.

We address this challenge with On-Policy Attention Self-Distillation (OPASD), which complements token-level OPSD with solution-conditioned attention supervision along the same student-generated trajectory, as illustrated in Figure 1. At each reasoning step, OPASD projects the privileged teacher's attention onto the student-visible support by removing solution-only positions and renormalizing the remaining mass. The resulting target remains conditioned on the verified solution but is defined entirely over positions visible to the student.

Across Qwen3 models from 1.7B to 8B and four competition-level mathematics benchmarks, OPASD consistently outperforms token-only OPSD, improving average accuracy by 4.98 to 8.40 percentage points. Importantly, the improvement is accompanied by substantially lower training cost rather than additional computational burden. OPASD reduces generated rollout tokens by 73.9% and estimated model compute by 72.6%, while training 1.53× faster despite the additional attention objective. OPASD also exhibits more stable training and validation performance, shorter reasoning trajectories, and smaller shifts in epistemic verbalization. Together, these results suggest that solution-conditioned attention provides a complementary supervision signal that improves not only reasoning accuracy but also the stability and efficiency of on-policy self-distillation.

In summary, we make three contributions:

- We introduce OPASD, extending privileged on-policy self-distillation with solution-conditioned attention supervision and a student-support projection that resolves the attention-support mismatch between privileged teachers and students.
- We show consistent improvements over token-only OPSD across three model scales and four competition-level mathematics benchmarks, with average accuracy gains of 4.98 to 8.40 points.
- We show that OPASD substantially improves the efficiency and stability of on-policy training, reducing generated rollout tokens by 73.9% and estimated compute by 72.6%, while training 1.53× faster.

2 Related Work

On-Policy Distillation. Conventional knowledge distillation for autoregressive language models trains students on fixed reference or teacher-generated sequences, creating a mismatch between the prefixes used for distillation and those generated by the student at inference time (Hinton et al., 2015; Kim & Rush, 2016; Bengio et al., 2015). OPD mitigates this mismatch by sampling trajectories from the student and querying the teacher at the resulting student-generated prefixes, providing dense supervision along the student's own rollout (Agarwal et al., 2024; Gu et al., 2024; Yang et al., 2026c; Li et al., 2026d; Fu et al., 2026; Hou et al., 2026; Ko et al., 2026; Song & Zheng, 2026). OPSD extends this principle to self-distillation by using the same model as both student and privileged

teacher under different conditioning contexts. The student generates the reasoning trajectory from the problem alone, while the privileged teacher additionally conditions on the verified solution and provides token-level supervision along that same trajectory (Zhao et al., 2026). Related work explores self-distillation with additional context, demonstrations, environment feedback, and other forms of guidance (Snell et al., 2022; Qi et al., 2025; Hübotter et al., 2026; Shenfeld et al., 2026; Ye et al., 2026; Penaloza et al., 2026; Stein et al., 2026; Sang et al., 2026; Ding, 2026; Li et al., 2026b; Yang et al., 2026a; He et al., 2026). Recent studies also examined when on-policy distillation is effective and explored different ways to improve the supervision it provides (Li et al., 2026d; Kim et al., 2026a; Harne et al., 2026; Yang et al., 2026c; Stein et al., 2026; Yang et al., 2026d; Jang et al., 2026; Jin et al., 2026; Jia et al., 2026; Xu et al., 2026; Zhu et al., 2026; Li et al., 2026c).

Representation Distillation. Intermediate-representation distillation complements output matching by aligning teacher and student representations. Early methods match features at selected layers or relations between layers (Romero et al., 2015; Yim et al., 2017), while later work extends intermediate supervision to Transformer language models through hidden-state, attention, and relation-based objectives (Sun et al., 2019; Sanh et al., 2019; Jiao et al., 2020; Sun et al., 2020; Wang et al., 2020; 2021; 2023). These methods generally train on fixed inputs, whereas recent work aligns teacher and student hidden states at selected layers and response positions along student-generated trajectories (Yang et al., 2026b). This brings representation-level supervision to states visited by the current policy. Unlike hidden-state alignment, OPASD explicitly aligns attention distributions over the preceding context, with the additional challenge that its privileged teacher observes solution tokens unavailable to the student.

Attention Distillation. Attention has long served as an intermediate target for knowledge distillation, from transferring attention maps in convolutional networks to aligning attention matrices and self-attention relations in Transformers (Zagoruyko & Komodakis, 2016; Jiao et al., 2020; Sun et al., 2020; Wang et al., 2020; 2021; 2023; Zhao et al., 2024; Jin et al., 2024). More recent work has applied attention supervision to language reasoning, where teacher attention is used to guide students toward task-relevant information during multi-step reasoning (Chen et al., 2025; Guo et al., 2025b). Attention-level supervision has also been explored in multimodal post-training, including on-policy attention distillation (Kim et al., 2026b; Li et al., 2026a). Recent work has further considered internal attention self-distillation, aligning intermediate-layer attention with a detached final-layer teacher over the same reasoning context (Liu et al., 2026). These studies establish attention as a useful source of supervision but do not address attention alignment in privileged on-policy self-distillation, where a solution-conditioned teacher evaluates student-generated reasoning over a different context. OPASD resolves this mismatch by projecting solution-conditioned teacher attention onto student-visible positions before aligning it with student attention alongside the token-level objective.

3 On-Policy Attention Self-Distillation

Building on OPSD (Zhao et al., 2026), we extend on-policy self-distillation beyond token-level supervision with an attention-level objective. Along each student-generated reasoning trajectory, the same privileged teacher provides supervision over both next-token and attention distributions. Because the teacher additionally observes the reference solution, its attention distribution is defined over positions that are unavailable to the student and therefore cannot be aligned directly with the student's attention distribution. We resolve this support mismatch by projecting the teacher attention onto the student-visible support and combining the resulting attention supervision with the original token-level objective.

3.1 On-Policy Self-Distillation

We consider a reasoning dataset $\mathcal{S} = \{(x_i, y_i^\star)\}_{i=1}^N$, where x_i denotes a problem and $y_i^\star$ its verified reference solution. Following OPSD, one language model defines a privileged teacher and a student under different conditioning contexts; the teacher uses the initial parameters θ_0, kept frozen throughout training. The teacher conditions on both the problem and its reference solution, while the student conditions only on the problem, matching the inference-time condition,

$$p_T(\cdot \mid x, y^\star) \triangleq p_{\theta_0}(\cdot \mid x, y^\star), \qquad p_S(\cdot \mid x) \triangleq p_\theta(\cdot \mid x). \tag{1}$$

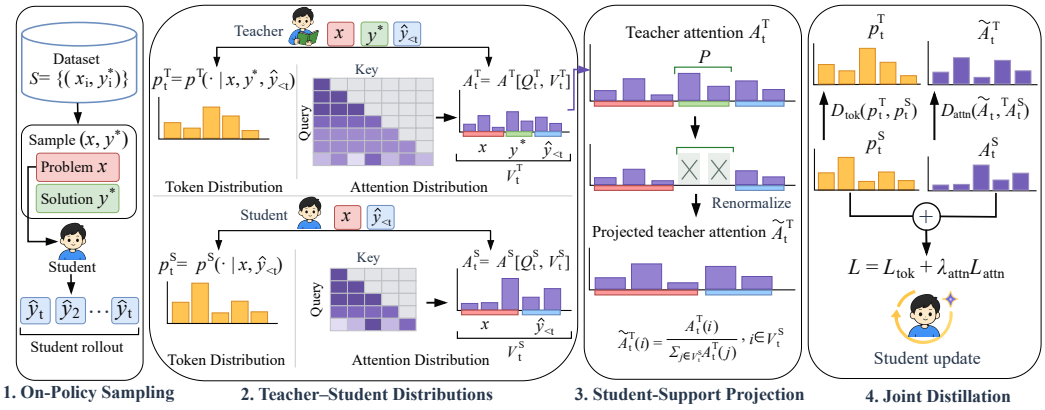

Figure 2: Overview of On-Policy Attention Self-Distillation (OPASD). (1) Given a problem x and its verified solution $y^\star$, the student samples an on-policy reasoning trajectory $\hat{y}$ from x alone. (2) The student and privileged teacher evaluate the same trajectory under their respective conditioning contexts, producing next-token distributions p_t^S, p_t^T and attention distributions A_t^S, A_t^T. (3) Because the teacher can attend to reference-solution positions unavailable to the student, its attention is restricted to the student-visible support and renormalized to obtain $\widetilde{A}_t^T$. (4) OPASD jointly aligns the token and projected attention distributions, with gradients propagated only through the student.

For each problem x, the student samples an on-policy reasoning trajectory

$$\hat{y} = (\hat{y}_1, \ldots, \hat{y}_T) \sim p_S(\cdot \mid x). \tag{2}$$

The privileged teacher evaluates this same trajectory rather than generating its own. At reasoning step t, the two policies produce

$$p_t^S = p_S(\cdot \mid x, \hat{y}_{<t}), \qquad p_t^T = p_T(\cdot \mid x, y^\star, \hat{y}_{<t}). \tag{3}$$

The token-level distillation objective is

$$\mathcal{L}_{\text{tok}} = \frac{1}{T} \sum_{t=1}^{T} D_{\text{tok}} \left(p_t^T, p_t^S \right), \tag{4}$$

where D_{tok} denotes the token-distribution divergence. Gradients are propagated only through the student, while the privileged teacher provides a detached target. We retain this objective and additionally use the privileged teacher for attention-level supervision.

3.2 Support Mismatch in Privileged Attention

The token distributions p_t^S and p_t^T are directly comparable because both are defined over the same vocabulary. Attention distributions are instead defined over context positions, and the teacher context additionally contains reference-solution positions that are absent from the student's context. We average the attention distributions across the N_h heads of the final transformer layer,

$$A^S = \frac{1}{N_h} \sum_{h=1}^{N_h} A^S_{\text{last},h}, \qquad A^T = \frac{1}{N_h} \sum_{h=1}^{N_h} A^T_{\text{last},h}. \tag{5}$$

For each generated token $\hat{y}_t$, let Q_t^S and Q_t^T denote its corresponding query positions in the student and teacher sequences, and let V_t^S and V_t^T denote their visible supports. Let $\mathcal{P}$ denote the reference-solution positions available only to the teacher, such that $V_t^T = V_t^S \cup \mathcal{P}$ and $V_t^S \cap \mathcal{P} = \varnothing$. The corresponding attention distributions are $A_t^S = A^S[Q_t^S, V_t^S]$ and $A_t^T = A^T[Q_t^T, V_t^T]$.

The privileged teacher can therefore assign attention mass to positions in $\mathcal{P}$ that are unavailable to the student. Consequently, A_t^S and A_t^T are defined over different supports and cannot be aligned directly. We resolve this mismatch by projecting the teacher attention onto the student-visible support V_t^S.

Algorithm 1 On-Policy Attention Self-Distillation (OPASD)

Require: Reasoning dataset $\mathcal{S} = \{(x_i, y_i^\star)\}_{i=1}^N$; language model p_θ; token divergence D_{tok}; attention divergence D_{attn}; attention-loss coefficient λ_{attn}

Define $p_S(\cdot \mid x)$ from p_θ and $p_T(\cdot \mid x, y^\star)$ from the frozen initialization p_{θ_0}
while not converged **do**
  Sample a minibatch $\mathcal{B} \subset \mathcal{S}$
  for all $(x, y^\star) \in \mathcal{B}$ **do**
    Sample an on-policy trajectory $\hat{y} \sim p_S(\cdot \mid x)$
    Evaluate both policies on $\hat{\mathbf{y}}$ to obtain $\{p_t^S, p_t^T, A_t^S, A_t^T\}_{t=1}^T$; detach teacher
    Compute token-level distillation: $\mathcal{L}_{\text{tok}}(x, y^\star) \leftarrow \frac{1}{T}\sum_{t=1}^T D_{\text{tok}}\big(p_t^T, p_t^S\big)$
    Project teacher attention: $\widetilde{A}_t^T(i) \leftarrow \frac{A_t^T(i)}{\sum_{j \in V_t^S} A_t^T(j)}, \quad i \in V_t^S$
    Compute attention-level distillation: $\mathcal{L}_{\text{attn}}(x, y^\star) \leftarrow \frac{1}{T}\sum_{t=1}^T D_{\text{attn}}\Big(\widetilde{A}_t^T, A_t^S\Big)$
  end for
  Compute the minibatch loss: $\mathcal{L}_{\mathcal{B}} \leftarrow \frac{1}{|\mathcal{B}|}\sum_{(x,y^\star)\in\mathcal{B}} \left[\mathcal{L}_{\text{tok}}(x, y^\star) + \lambda_{\text{attn}}\mathcal{L}_{\text{attn}}(x, y^\star)\right]$
  Update the student parameters using $\nabla_\theta \mathcal{L}_{\mathcal{B}}$
end while

3.3 Student-Support Projection

To obtain a teacher attention target on the same support as the student, we restrict A_t^T to the student-visible positions V_t^S and renormalize the remaining mass (Theorem O.2). For each position $i \in V_t^S$, the projected teacher attention is

$$\widetilde{A}_t^T(i) = \frac{A_t^T(i)}{\sum_{j \in V_t^S} A_t^T(j)}. \tag{6}$$

The resulting distribution $\widetilde{A}_t^T$ is defined over the same support as A_t^S. Although attention assigned to the reference-solution positions is removed, the target remains solution-conditioned because A_t^T is computed under the privileged context $(x, y^\star, \hat{y}_{<t})$.

The projection also preserves the teacher's relative attention over the student-visible positions. Since every position in V_t^S is rescaled by the same normalization factor, for any $i, j \in V_t^S$ with $A_t^T(j) > 0$,

$$\frac{\widetilde{A}_t^T(i)}{\widetilde{A}_t^T(j)} = \frac{A_t^T(i)}{A_t^T(j)}. \tag{7}$$

3.4 Joint Distillation

After projection, the student attention A_t^S and the projected teacher attention $\widetilde{A}_t^T$ are defined over the same support. We distill the projected teacher attention using the generalized Jensen-Shannon divergence (JSD). Let $M_t^{(\beta)} = \beta\widetilde{A}_t^T + (1-\beta)A_t^S$, where $\beta \in (0, 1)$. The per-step attention objective is

$$\ell_{\text{attn}}(t) = D_{\text{attn}}\left(\widetilde{A}_t^T, A_t^S\right) \triangleq \beta D_{\text{KL}}\left(\widetilde{A}_t^T \,\|\, M_t^{(\beta)}\right) + (1-\beta)D_{\text{KL}}\left(A_t^S \,\|\, M_t^{(\beta)}\right). \tag{8}$$

After rescaling, $\beta \to 0$ and $\beta \to 1$ recover the forward and reverse KL, which we use as the endpoint variants.

$$\mathcal{L}_{\text{attn}} = \frac{1}{T}\sum_{t=1}^T \ell_{\text{attn}}(t). \tag{9}$$

The training objective combines the token-level OPSD loss with the proposed attention-level loss,

$$\mathcal{L} = \mathcal{L}_{\text{tok}} + \lambda_{\text{attn}}\mathcal{L}_{\text{attn}}. \tag{10}$$

Here λ_{attn} controls the strength of attention supervision. Only the student branch receives gradients; the privileged teacher and reference solution are used only during training. Algorithm 1 summarizes the complete training procedure.

Table 1: Comparison of on-policy self-distillation methods across model scales. Avg@12 accuracy (%) of the instruction-tuned Base model, token-only OPSD, and OPASD on four competition-level mathematical reasoning benchmarks. OPASD achieves the highest performance across all three Qwen3 model sizes.

Method	AIME24	AIME25	AIME26	HMMT25	Average
Qwen3-8B					
Base	60.83	48.89	51.39	29.17	47.57
OPSD	59.16	51.97	53.33	33.33	49.45
OPASD	**67.50**	**57.22**	**61.11**	**35.83**	**55.42**
Qwen3-4B					
Base	58.33	47.50	51.11	30.27	46.80
OPSD	51.11	43.88	46.11	28.33	42.36
OPASD	**63.33**	**50.27**	**55.56**	**33.89**	**50.76**
Qwen3-1.7B					
Base	33.61	30.28	31.94	19.17	28.75
OPSD	36.70	28.33	32.78	18.33	29.04
OPASD	**43.05**	**35.27**	**34.44**	**23.33**	**34.02**

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models and datasets. We experiment with instruction-tuned models from the Qwen3 (Team et al., 2025) family: Qwen3-1.7B, Qwen3-4B, Qwen3-8B. For training, we use the mathematical reasoning subset of OpenThoughts (Guha et al., 2026), sampling up to 30K problem-solution pairs. We evaluate on competition-level mathematics benchmarks including AIME 2024, AIME 2025, AIME 2026, and HMMT 2025.

Baselines. We compare against two baselines: (1) BASE, the original instruction-tuned checkpoint without post-training; and (2) OPSD (Zhao et al., 2026), the token-only on-policy self-distillation method. OPSD and our method are trained using the same dataset, model initialization, and training budget.

Implementation Details. We keep the privileged teacher fixed at the initial checkpoint throughout training. For token-level supervision, we use full-vocabulary logit distillation. For attention-level supervision, we align the student's final-layer attention with the projected teacher attention over the student-visible support. All experiments are conducted on H100 or H200 GPUs using LoRA (Hu et al., 2021). Complete training and evaluation configurations are provided in Appendix M.

4.2 COMPARISON OF ON-POLICY SELF-DISTILLATION METHODS

Table 1 compares OPASD with the instruction-tuned base model and token-only OPSD across three Qwen3 model sizes. OPASD achieves the highest average performance at every scale and outperforms OPSD on all four benchmarks. The average score rises from 29.04 to 34.02 on Qwen3-1.7B, from 42.36 to 50.76 on Qwen3-4B, and from 49.45 to 55.42 on Qwen3-8B, giving gains of 4.98, 8.40, and 5.97 points, respectively. OPASD also surpasses the base model at every scale. These consistent gains show that attention-level supervision complements the token-level objective across model scales and benchmark difficulty levels.

We further evaluate OPASD across sampling budgets, multiple training seeds, and test-time generation budgets in Appendices B, F, and G, respectively. These analyses show that OPASD improves both Pass@1 and Pass@12, remains stable across training seeds, and consistently outperforms OPSD across generation budgets.

4.3 ABLATION STUDIES AND ANALYSIS

We examine three design choices in OPASD: (1) the divergence used for attention distillation (Section 4.3.1), (2) the weight of attention supervision (Section 4.3.2), and (3) the depth of attention

distillation (Section 4.3.3). Additional experiments evaluate how teacher-only attention is handled (Appendix H), the token-loss clipping strategy (Appendix I), and the individual contributions of token- and attention-level supervision (Appendix J). We then present a qualitative comparison of OPSD and OPASD reasoning trajectories (Section 4.3.4), analyze their performance and behavioral dynamics (Section 4.3.5), and compare their efficiency (Section 4.3.6).

4.3.1 Attention Divergence Objective

We compare forward KL (FKL) ($\widetilde{A}_t^T \parallel A_t^S$), reverse KL (RKL) ($A_t^S \parallel \widetilde{A}_t^T$), and JSD ($\beta = 0.5$) for attention distillation on Qwen3-1.7B, while keeping all other settings fixed. As shown in Table 2, JSD achieves the highest average accuracy of 32.85, compared with 32.36 for reverse KL and 31.73 for forward KL. JSD achieves this advantage through a symmetric and bounded objective that balances the mode-covering behavior of forward KL with the mode-seeking behavior of reverse KL. This balance preserves the broader context emphasized by the teacher while retaining focus on its most relevant positions.

Table 2: Effect of the attention divergence objective. JSD achieves the highest average accuracy across the three objectives.

Objective	AIME24	AIME25	AIME26	HMMT25	Avg
FKL	39.72	**33.05**	33.05	**21.11**	31.73
RKL	40.83	32.22	**35.83**	20.56	32.36
JSD	**41.94**	32.50	**35.83**	**21.11**	**32.85**

4.3.2 Strength of Attention Supervision

We vary the attention-loss weight λ_{attn} on Qwen3-1.7B while keeping the token-loss coefficient fixed at 1. As shown in Table 3, the highest average score is 34.02 at $\lambda_{\text{attn}} = 0.5$, while weights of 1.0 and 2.0 yield 32.85 and 31.82, respectively. This suggests that attention supervision is most effective as a complementary signal, with larger weights placing excessive emphasis on matching the teacher's attention and reducing the influence of the token-level objective.

Table 3: Effect of attention loss weight. With the token-loss coefficient fixed at 1, $\lambda_{\text{attn}} = 0.5$ yields the highest average performance.

λ_{attn}	AIME24	AIME25	AIME26	HMMT25	Avg
0.25	**43.33**	34.16	35.56	21.94	33.75
0.5	43.05	**35.27**	34.44	**23.33**	**34.02**
1.0	41.94	32.50	**35.83**	21.11	32.85
2.0	39.44	32.28	34.44	21.11	31.82

4.3.3 Attention Distillation Depth

We ablate the number of final transformer layers used for attention distillation on Qwen3-1.7B. For multi-layer variants, we compute a separate loss from the head-averaged attention at each layer and average the losses. As shown in Table 4, using only the final layer achieves the highest average score of 34.02, compared with 32.54 and 31.04 for the final two and three layers. Giving earlier-layer attention equal weight shifts some supervision toward broader intermediate patterns. Distilling only the final layer works better because its attention acts on representations refined by the preceding layers and is more closely tied to the model's next-token prediction.

Table 4: **Effect of attention distillation depth.** Using only the final transformer layer achieves the highest average performance compared with using the final two or three layers.

Layers	AIME24	AIME25	AIME26	HMMT25	Avg
1	**43.05**	**35.27**	**34.44**	**23.33**	**34.02**
2	41.13	34.57	33.36	21.11	32.54
3	39.44	30.83	33.88	20.00	31.04

4.3.4 Qualitative Analysis

Figure 3 compares the base model, OPSD, and OPASD on the same number-theory problem. The base model commits to an incomplete solution, while OPSD explores additional possibilities but evaluates them incorrectly. OPASD carries this broader analysis through consistently and reaches the correct solution. OPASD's attention supervision provides additional guidance on earlier calculations,

Question. Let p be the least prime number for which there exists a positive integer n such that $n^4 + 1$ is divisible by p^2. Find the least positive integer m such that $m^4 + 1$ is divisible by p^2.

Base

Okay, so I need to find the least prime number p such that there's a positive integer n where $n^4 + 1$ is divisible by p^2. ... Since we constructed n = 155 to satisfy this condition, and it is the unique solution modulo 17^2, it is the smallest such positive integer. ... Final Answer: 155

OPSD

Okay, let's see. The problem is asking for the smallest prime p such that there exists some positive integer n with $n^4 + 1$ divisible by p^2. ... Other possible lifts: Root≡2: gives m=155. Root≡9: gives2899 (large). Root≡15:gives m=134. Root≡8:gives m=1164 (large). ... Therefore minimal m=134.

OPASD

Okay, so I need to find the smallest prime number p such that there's some positive integer n where $n^4 + 1$ is divisible by p squared. ... Let's consider $m = 8 + 17k$, and compute $m^4 + 1$ mod 289. ... $8k \equiv 14 \pmod{17}$. ... The inverse of 8 modulo 17 is 15 ... $k \equiv 14 \cdot 15 \equiv 210 \equiv 6 \pmod{17}$. ... Thus, the smallest such k is 6, and: $m = 8 + 17 \cdot 6 = 110$. ... Therefore, the smallest positive integer m such that $m^4 + 1 \equiv 0 \pmod{289}$ is: 110.

Figure 3: Qualitative comparison of reasoning across the base model, OPSD, and OPASD on an AIME24 number-theory problem. The base model overlooks valid alternatives, OPSD considers more candidates but evaluates one incorrectly, and OPASD identifies the correct minimum. Highlighted excerpts mark the decisive steps.

helping the student evaluate later candidates more consistently than token-level supervision alone. The probability and geometry examples in Appendix L illustrate a similar distinction.

4.3.5 Performance and Behavioral Dynamics

Figure 4 compares OPSD and OPASD in training and validation accuracy, response length, and epistemic-token usage.

Figure 4(a) shows that OPASD maintains rollout accuracy between 0.70 and 0.80, with response lengths remaining between 1.0K and 1.6K tokens. In contrast, OPSD's accuracy declines to 0.50–0.60 as its response length grows from 1.1K to 6.1K tokens. Figure 4(b) shows that this contrast extends to validation. Across the recorded checkpoints, OPASD's Avg@12 accuracy varies by only 2.22 points, whereas OPSD's accuracy spans 10.76 points and declines steadily after approximately 100 steps. OPASD also maintains consistently shorter validation responses. Because OPSD optimizes token-level teacher-student agreement without an explicit outcome signal, it does not directly distinguish

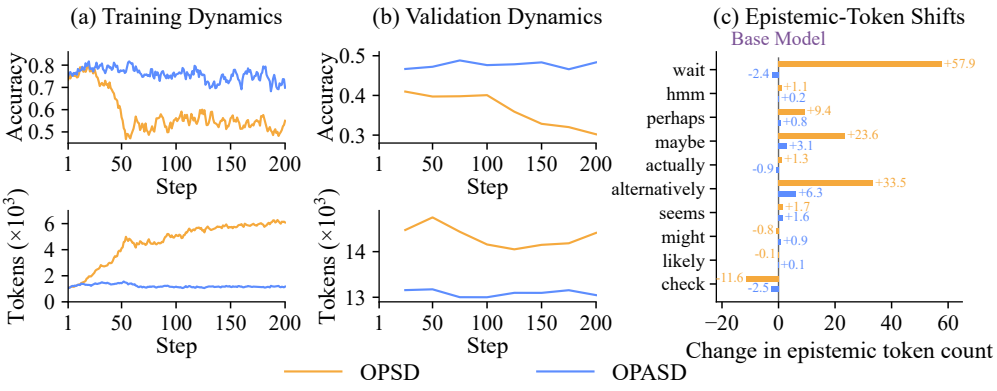

Figure 4: Training dynamics and epistemic-token shifts on Qwen3-4B. OPASD maintains stable training and validation performance with concise responses and limited epistemic-token shifts, whereas OPSD exhibits declining accuracy, response-length inflation, and substantially greater use of uncertainty and reconsideration markers.

continuation that improves the final answer from continuation that merely extends the reasoning process. OPASD adds attention supervision that guides the student toward context emphasized by the solution-conditioned teacher. This contextual signal helps preserve relevant information and limit repetitive continuation as the student progressively learns the teacher's attention distribution (Appendix E).

Figure 4(c) compares changes in the mean number of epistemic tokens (expressions of uncertainty, reconsideration, or self-verification) (Kim et al., 2026a) per response from the base model to the final OPSD and OPASD checkpoints across all four benchmarks. Across the ten markers, the mean count increases by 70.9% for OPSD but only 4.4% for OPASD. The difference remains after normalizing for response length and within both correct and incorrect groups, with OPSD using *wait*, *alternatively*, and *maybe* markers more often than OPASD (Appendix D). Together with OPSD's increasing response length and declining validation accuracy, this result suggests that OPSD produces more repeated uncertainty and reconsideration without improving its reasoning performance. This is because OPSD aligns the next-token distribution without directly guiding which earlier steps the student should revisit to resolve uncertainty. OPASD additionally aligns the student's attention distribution with that of the solution-conditioned teacher, guiding its use of earlier context during reconsideration.

4.3.6 Efficiency

Figure 5 compares OPSD and OPASD over 200 Qwen3-4B training steps on four H100 GPUs. Compared with OPSD, OPASD improves average Avg@12 across the four benchmarks from 42.36 to 50.76 and reduces wall-clock time from 8.04 to 5.27 hours, a $1.53\times$ speedup. OPASD generates 15.58 million response tokens compared with 59.73 million for OPSD, a 73.9% reduction, while also reducing estimated model compute by 72.6% and peak allocated GPU memory by 11.9%. Although attention supervision increases actor-update cost from 0.07 to 0.12 ms per processed token, OPASD avoids OPSD's response-length growth and substantially reduces the total number of tokens generated and processed. OPASD therefore Pareto-dominates OPSD in accuracy, wall-clock time, generated tokens, and peak memory, with lower estimated model compute as well. We further examine token-learning efficiency in Appendix C.

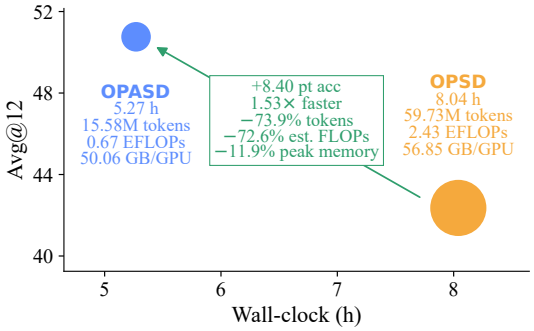

Figure 5: OPASD is Pareto-dominant over OPSD with higher Avg@12 and lower wall-clock time, estimated model compute, generated response tokens, and peak GPU memory. Estimated compute is reported in EFLOPs (10^{18} FLOPs). Bubble area is proportional to the total number of response tokens generated.

5 Conclusion

OPSD distills the privileged teacher's next-token distribution but leaves its solution-conditioned attention distribution unused. Transferring this signal requires resolving a support mismatch because the teacher can attend to verified-solution positions the student cannot observe. We introduce OPASD to address this mismatch by projecting and renormalizing teacher attention over student-visible positions before distilling it alongside token distributions. Across three model scales and four mathematics benchmarks, OPASD consistently improves accuracy over token-only OPSD. Further analyses show more stable training and validation accuracy, shorter responses, smaller shifts in epistemic-token usage, and lower training cost. Qualitative examples complement these aggregate findings by illustrating more consistent use of intermediate results. Together, these findings show that attention supervision complements token-level self-distillation.

Limitations. OPASD outperforms OPSD across three Qwen3 model sizes, but its evaluation with other model families would provide a comprehensive assessment. We keep the privileged teacher fixed throughout training. Whether attention supervision offers similar benefits under other teacher-update strategies remains to be studied.

AI USE STATEMENT

We used generative AI tools to obtain feedback on experimental design and to assist with the analysis of experimental findings. These tools also supported literature searches, analysis code, figure preparation, and the drafting and editing of parts of the manuscript. The authors reviewed all AI-assisted material and take responsibility for the methods, results, claims, code, and figures in this paper.

ETHICS STATEMENT

OPASD studies mathematical reasoning using an existing training dataset and established benchmarks. It does not involve human subjects or collect new personal data. Although we evaluate our method on mathematical reasoning, it could be applied in other domains and would inherit risks associated with the underlying models and training data. Models trained with OPASD should be assessed for reliability and misuse before deployment.

REPRODUCIBILITY STATEMENT

Section 3 describes the OPASD training objective and how the privileged teacher's attention is projected onto positions visible to the student. Section 4 reports the models, training data, baselines, evaluation benchmarks, and ablation studies used to assess the method. We will publicly release the code and experiment configurations to enable reproduction of our results.

REFERENCES

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

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

Yao Chen, Jiawei Sheng, Wenyuan Zhang, and Tingwen Liu. 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*, pp. 4952–4971, 2025.

Ken Ding. Hdpo: Hybrid distillation policy optimization via privileged self-distillation. *arXiv preprint arXiv:2603.23871*, 2026.

Yuqian Fu, Haohuan Huang, Kaiwen Jiang, Jiacai Liu, Zhuo Jiang, Yuanheng Zhu, and Dongbin Zhao. Revisiting on-policy distillation: Empirical failure modes and simple fixes. *arXiv preprint arXiv:2603.25562*, 2026.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *International Conference on Learning Representations*, volume 2024, pp. 32694–32717, 2024.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, et al. Openthoughts: Data recipes for reasoning models. In *International Conference on Learning Representations*, volume 2026, pp. 108059–108130, 2026.

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.

Yiju Guo, Wenkai Yang, Zexu Sun, Ning Ding, Zhiyuan Liu, and Yankai Lin. Learning to Focus: Causal attention distillation via gradient-guided token pruning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 24921–24948, 2025b. URL https://arxiv.org/abs/2506.07851.

Sarthak Harne, Chinmay Karkar, Yash Pandya, Ahmed Awadallah, and Akshay Nambi. Privileged, but biased: How pi-conditioned teachers break self-distillation. *arXiv preprint arXiv:2608.04794*, 2026.

Yinghui He, Simran Kaur, Adithya Bhaskar, Yongjin Yang, Jiarui Liu, Narutatsu Ri, Liam Fowl, Abhishek Panigrahi, Danqi Chen, and Sanjeev Arora. Self-distillation zero: Self-revision turns binary rewards into dense supervision. *arXiv preprint arXiv:2604.12002*, 2026.

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

Wenjin Hou, Shangpin Peng, Weinong Wang, Zheng Ruan, Yue Zhang, Zhenglin Zhou, Mingqi Gao, Yifei Chen, Kaiqi Wang, Hongming Yang, et al. Uni-opd: Unifying on-policy distillation with a dual-perspective recipe. *arXiv preprint arXiv:2605.03677*, 2026.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

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

Ijun Jang, Jewon Yeom, Juan Yeo, Hyunggyu Lim, and Taesup Kim. Stable on-policy distillation through adaptive target reformulation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 42217–42227, 2026.

Nan Jia, Haojin Yang, Xing Ma, Jiesong Lian, Shuailiang Zhang, Weipeng Zhang, Ke Zeng, Xunliang Cai, and Zequn Sun. Asymmetric on-policy distillation: Bridging exploitation and imitation at the token level. *arXiv preprint arXiv:2605.06387*, 2026.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. Tinybert: Distilling bert for natural language understanding. In *Findings of the association for computational linguistics: EMNLP 2020*, pp. 4163–4174, 2020.

Heegon Jin, Seonil Son, Jemin Park, Youngseok Kim, Hyungjong Noh, and Yeonsoo Lee. Align-to-distill: Trainable attention alignment for knowledge distillation in neural machine translation. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 722–732, 2024.

Woogyeol Jin, Taywon Min, Yongjin Yang, Dennis Wei, Yi Zhou, Swanand Ravindra Kadhe, Nathalie Baracaldo, and Kimin Lee. Entropy-aware on-policy distillation of language models. *arXiv preprint arXiv:2603.07079*, 2026.

Jeonghye Kim, Xufang Luo, Minbeom Kim, Sangmook Lee, Dohyung Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Why does self-distillation (sometimes) degrade the reasoning capability of llms? *arXiv preprint arXiv:2603.24472*, 2026a.

Jiwan Kim, Kibum Kim, Sangwoo Seo, and Chanyoung Park. Compodistill: Attention distillation for compositional reasoning in multimodal llms. In *International Conference on Learning Representations*, volume 2026, pp. 71814–71844, 2026b.

Yoon Kim and Alexander M Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pp. 1317–1327, 2016.

Jongwoo Ko, Sara Abdali, Young Jin Kim, Tianyi Chen, and Pashmina Cameron. Scaling reasoning efficiently via relaxed on-policy distillation. *arXiv preprint arXiv:2603.11137*, 2026.

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

Gengsheng Li, Tianyu Yang, Junfeng Fang, Mingyang Song, Mao Zheng, Haiyun Guo, Dan Zhang, Jinqiao Wang, and Tat-Seng Chua. Unifying group-relative and self-distillation policy optimization via sample routing. *arXiv preprint arXiv:2604.02288*, 2026b.

Yang Li, Semih Yavuz, and Shafiq Joty. Rise: Recursive improvement via self-extrapolating policy distillation. *arXiv preprint arXiv:2609.05295*, 2026c.

Yaxuan Li, Yuxin Zuo, Bingxiang He, Jinqian Zhang, Chaojun Xiao, Cheng Qian, Tianyu Yu, Huanang Gao, Wenkai Yang, Zhiyuan Liu, et al. Rethinking on-policy distillation of large language models: Phenomenology, mechanism, and recipe. *arXiv preprint arXiv:2604.13016*, 2026d.

Jianhua Lin. Divergence measures based on the shannon entropy. *IEEE Transactions on Information theory*, 37(1):145–151, 1991.

Xinyu Liu, Darryl Cherian Jacob, Yang Zhou, Jindong Wang, and Pan He. Oisd: On-policy internal self-distillation of language models. *arXiv preprint arXiv:2605.29089*, 2026.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models. *arXiv preprint arXiv:2602.04942*, 2026.

Siyuan Qi, Bangcheng Yang, Kailin Jiang, Xiaobo Wang, Jiaqi Li, Yifan Zhong, Yaodong Yang, and Zilong Zheng. In-context editing: Learning knowledge from self-induced distributions. In *International Conference on Learning Representations*, volume 2025, pp. 77563–77585, 2025.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. FitNets: Hints for thin deep nets. In *International Conference on Learning Representations*, 2015. URL `https://arxiv.org/abs/1412.6550`.

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. On-policy self-distillation for reasoning compression. *arXiv e-prints*, pp. arXiv–2603, 2026.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

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.

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

Charlie Snell, Dan Klein, and Ruiqi Zhong. Learning by distilling context. *arXiv preprint arXiv:2209.15189*, 2022.

Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*, 2026.

Alex Stein, Furong Huang, and Tom Goldstein. Gates: Self-distillation under privileged context with consensus gating. *arXiv preprint arXiv:2602.20574*, 2026.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 4323–4332, 2019.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. Mobilebert: a compact task-agnostic bert for resource-limited devices. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pp. 2158–2170, 2020.

Qwen Team et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 6(7):13, 2025.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in neural information processing systems*, 33:5776–5788, 2020.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. Minilmv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 2140–2151, 2021.

Xinpeng Wang, Leonie Weissweiler, Hinrich Schütze, and Barbara Plank. How to distill your bert: An empirical study on the impact of weight initialisation and distillation objectives. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 1843–1852, 2023.

Yuanda Xu, Hejian Sang, Zhengze Zhou, Ran He, Zhipeng Wang, and Alborz Geramifard. Tip: Token importance in on-policy distillation. *arXiv preprint arXiv:2604.14084*, 2026.

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled rlvr. *arXiv preprint arXiv:2604.03128*, 2026a.

Shenzhi Yang, Guangcheng Zhu, Bowen Song, Haobo Wang, Mingxuan Xia, Xing Zheng, Yingfan Ma, Zhongqi Chen, Weiqiang Wang, Junbo Zhao, et al. Oprd: On-policy representation distillation. *arXiv preprint arXiv:2606.06021*, 2026b.

Wenkai Yang, Weijie Liu, Ruobing Xie, Kai Yang, Saiyong Yang, and Yankai Lin. Learning beyond teacher: Generalized on-policy distillation with reward extrapolation. *arXiv preprint arXiv:2602.12125*, 2026c.

Yuxiao Yang, Xiaoyun Wang, and Weitong Zhang. Ogls-sd: On-policy self-distillation with outcome-guided logit steering for llm reasoning. *arXiv preprint arXiv:2605.12400*, 2026d.

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

Junho Yim, Donggyu Joo, Jihoon Bae, and Junmo Kim. A gift from knowledge distillation: Fast optimization, network minimization and transfer learning. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*, pp. 7130–7138. IEEE, 2017.

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.

Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016.

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

Tianyang Zhao, Kunwar Yashraj Singh, Srikar Appalaraju, Peng Tang, Vijay Mahadevan, R Manmatha, and Ying Nian Wu. No head left behind–multi-head alignment distillation for transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 7514–7524, 2024.

Siqi Zhu, Xuyan Ye, Hongyu Lu, Weiye Shi, and Ge Liu. The many faces of on-policy distillation: Pitfalls, mechanisms, and fixes. *arXiv preprint arXiv:2605.11182*, 2026.

A APPENDIX

The appendix provides additional analyses, ablations, implementation details, and theoretical results supporting the main paper.

B PERFORMANCE ACROSS SAMPLING BUDGETS

We examine whether OPASD improves the reliability of individual responses or primarily benefits from repeated sampling. Figure 6 reports Pass@1 and Pass@12 throughout the Qwen3-4B training run on each benchmark.

OPASD produces more reliable individual responses. The Pass@1 curves show a consistent separation between the two methods. OPASD maintains higher accuracy across all four benchmarks, whereas OPSD gradually loses performance during later training. At the final checkpoint, OPASD achieves an average Pass@1 of 48.33, compared with 30.21 for OPSD. OPASD's attention objective provides later steps with additional guidance about which earlier positions the solution-conditioned teacher emphasizes. Its advantage at Pass@1 therefore reflects stronger individual responses rather than a dependence on repeated attempts.

Repeated sampling narrows but does not remove the advantage. Pass@12 is more variable because allowing multiple attempts can recover problems that either method solves inconsistently. Nevertheless, OPASD improves the final average Pass@12 from 56.51 to 63.82. The gains are clear on AIME24, AIME25, and HMMT25, while the two methods perform comparably on AIME26. The smaller difference under Pass@12 indicates that additional samples partially compensate for the lower single-response accuracy of OPSD. OPASD still retains an advantage because the same attention-supervised model generates each attempt, so its stronger single-response reliability remains relevant when the sampling budget increases.

C TOKEN LEARNING EFFICIENCY

We examine whether OPASD obtains stronger validation performance from the same generation budget. Figure 7 reports Avg@12 accuracy against the cumulative response tokens generated by the training rollouts throughout Qwen3-4B training on each benchmark. Over the complete 200-step

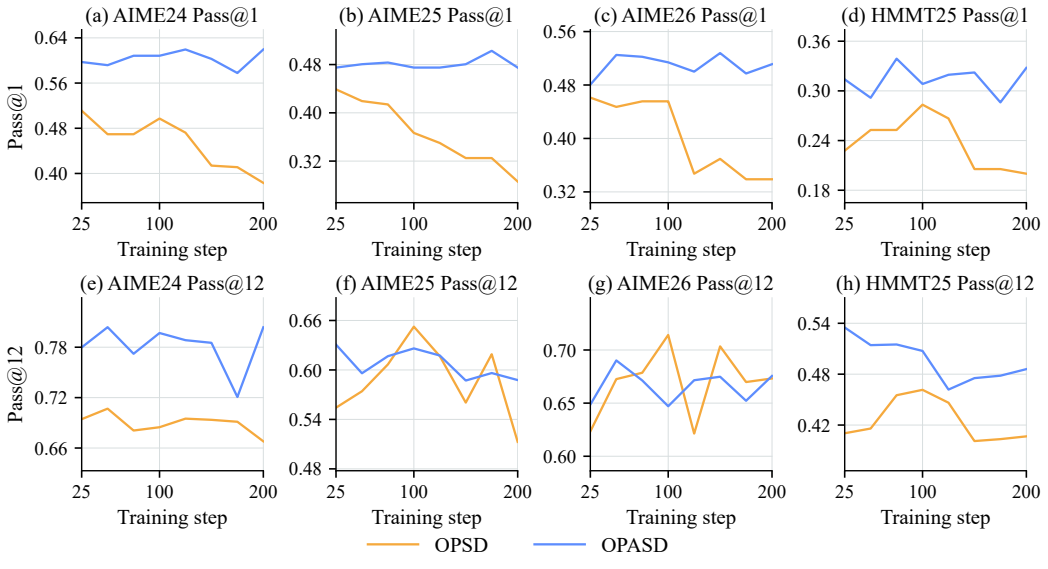

Figure 6: Pass@1 and Pass@12 dynamics of OPSD and OPASD over 200 training steps on Qwen3-4B. OPASD consistently achieves higher Pass@1 across all four benchmarks and retains an overall advantage under Pass@12. Curves show the evaluations recorded for every 25 training steps.

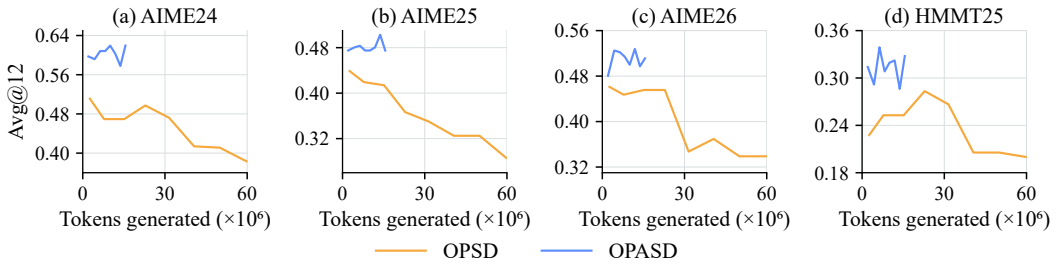

Figure 7: Token-learning efficiency of OPSD and OPASD over 200 training steps on Qwen3-4B. OPASD consistently achieves higher accuracy across all four benchmarks. The curves report validation accuracy at 25 step intervals.

training runs, OPASD generates 15.58 million response tokens compared with 59.73 million for OPSD, reducing token usage by 73.9% while consistently achieving higher validation accuracy across all four benchmarks. These results show that OPASD uses the generated-token budget more effectively by maintaining shorter and more productive reasoning trajectories. This is because OPSD's response-length growth spends much of its budget on repeated uncertainty and reconsideration (Section 4.3.5), which adds tokens without improving answers. OPASD's attention supervision keeps the student focused on relevant context, which is why it needs fewer tokens to reason better.

D Epistemic Verbalization Dynamics

Epistemic verbalization refers to expressions of uncertainty, reconsideration, or self-verification within a reasoning trace. Following Kim et al. (2026a), we measure ten epistemic markers: *wait*, *hmm*, *perhaps*, *maybe*, *actually*, *alternatively*, *seems*, *might*, *likely*, and *check*. We treat these markers as a behavioral proxy for epistemic verbalization rather than a direct measure of the model's internal uncertainty. Tracking these markers alongside response length and accuracy helps reveal whether training changes the model's reasoning style rather than merely the number of generated tokens. We report marker occurrences per 1,000 response tokens to account for differences in response length.

OPASD remains stable while OPSD exhibits increasing epistemic verbalization. Figure 8 reports epistemic-marker density in the periodic validation responses. Across the four benchmarks, OPASD remains within a narrow range of 12-14 markers per 1,000 tokens throughout training. In contrast, OPSD ranges from 14 to 23 markers and generally increases as training progresses. Figure 4 shows that OPSD's responses also grow longer as its validation accuracy declines, while OPASD remains stable on both measures. For OPSD, increasing reconsideration therefore accompanies more generations without better final answers. OPASD instead supervises how the student attends to earlier reasoning alongside next-token predictions, and maintains shorter responses and higher accuracy.

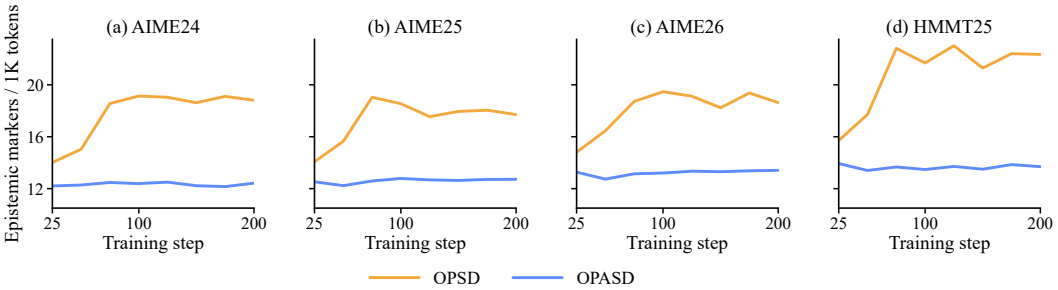

Figure 8: Epistemic-marker density throughout training. Marker frequency per 1,000 tokens in periodic Qwen3-4B validation responses. OPASD maintains stable epistemic verbalization across all four benchmarks, whereas OPSD exhibits a substantial increase as training progresses.

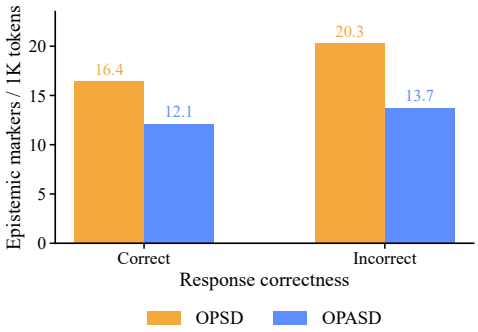

Figure 9: Epistemic-marker density by response correctness. Marker frequency per 1,000 tokens at the final Qwen3-4B checkpoint, pooled across all four benchmarks. OPASD produces fewer markers than OPSD for both correct and incorrect responses. This pattern is consistent with OPASD avoiding the excessive reconsideration observed in OPSD.

OPASD maintains lower epistemic-marker density than OPSD regardless of response correctness. Figure 9 separates the final-checkpoint responses according to correctness. For correct responses, OPSD produces 16.4 epistemic markers per 1,000 tokens, whereas OPASD produces 12.1. For incorrect responses, OPSD produces 20.3 markers, compared with 13.7 for OPASD. Incorrect responses therefore contain more epistemic verbalization under both methods, suggesting that frequent reconsideration is associated with unresolved reasoning. However, OPASD consistently produces fewer markers than OPSD in both groups. This gap within each group shows that OPASD's lower marker density is not simply a consequence of its higher accuracy. Its attention supervision guides the use of earlier reasoning, supporting more focused continuations regardless of the final outcome.

OPASD reduces repeated uncertainty while retaining explicit verification. Figure 10 separates the overall density into its individual markers at the final checkpoint. OPSD uses *wait*, *maybe*, *alternatively*, and *perhaps* more frequently than OPASD. The largest difference appears for *wait*, with 7.42 occurrences per 1,000 tokens for OPSD compared with 3.58 for OPASD. In contrast, OPASD uses *check* more frequently, with 1.21 occurrences compared with 0.46 for OPSD. OPASD thus uses fewer markers of uncertainty and alternative search without uniformly reducing epistemic language. Its attention objective provides guidance over earlier positions to use when reconsidering a step, supporting a more focused response rather than an extended search through alternatives. This matters because the change accompanies shorter responses and higher final-answer accuracy, not merely fewer epistemic markers.

E Attention Alignment Dynamics

We examine whether attention alignment remains effective as the student-generated trajectories evolve during training. Figure 11 tracks the JSD between student attention and projected solution-conditioned teacher attention on Qwen3-4B. The loss decreases rapidly during early training and continues to decline thereafter, falling from an average of 0.0055 over the first 20 steps to 0.0029 over the final 20 steps. Because the teacher's target is recomputed for each new trajectory, the decline shows that OPASD keeps aligning attention on the reasoning it currently generates. This sustained alignment allows the teacher's guidance over earlier information to remain relevant as the student's reasoning changes throughout training.

F Robustness across random seeds

To assess sensitivity to training randomness, we conduct two additional Qwen3-4B runs with seeds 1 and 123, complementing the original run with seed 42. All runs use the same training data, hyperparameters, compute budget, and experimental configuration. Table 5 reports the results across seeds. OPASD achieves an average score of 50.88 ± 0.14 (mean $\pm$ standard deviation), with a 95% confidence interval (CI) of $[50.53, 51.24]$. The small variation across the tested seeds indicates that OPASD's performance is stable with respect to training randomness.

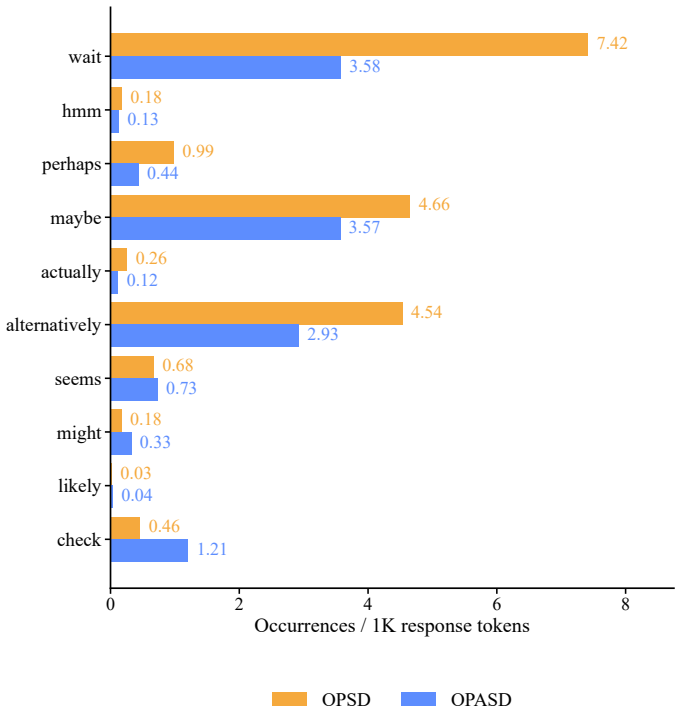

Figure 10: Composition of epistemic verbalization at the final checkpoint. Frequency of each epistemic marker per 1,000 response tokens, pooled across all four benchmarks. OPSD's higher overall density is driven primarily by *wait*, *maybe*, *alternatively*, and *perhaps*, whereas OPASD uses *check* more frequently.

G Test-Time Generation Budget

We examine whether the advantage of OPASD persists across different test-time generation budgets. Both methods are trained with an 8K-token response cap and evaluated with maximum generation lengths of 16K, 28K, and 38K tokens. As shown in Figure 12, increasing the generation budget improves the average performance of both methods. OPASD outperforms OPSD on every benchmark at all three budgets, with average gains of 8.40, 6.25, and 5.81 points at 16K, 28K, and 38K, respectively. These results show that the advantage of OPASD remains consistent across test-time generation budgets. OPASD stays ahead at every budget because its attention supervision teaches the

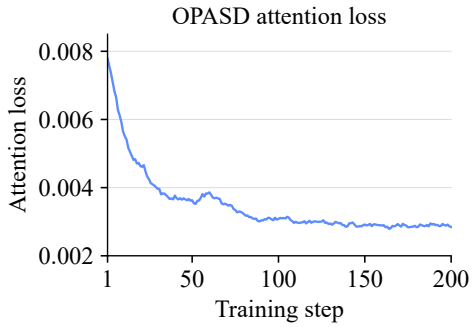

Figure 11: Attention-alignment dynamics of OPASD on Qwen3-4B. The divergence between student attention and projected solution-conditioned teacher attention decreases throughout training. The curve is an exponential moving average with smoothing factor 0.7.

student to focus on the context emphasized by the solution-conditioned teacher, helping it reason better.

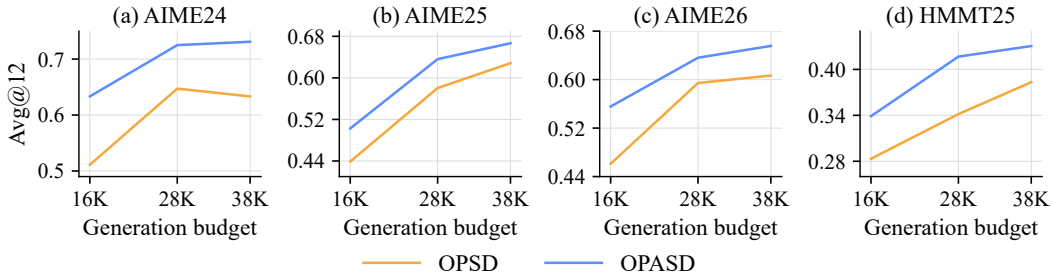

Figure 12: Avg@12 performance of OPSD and OPASD across test-time generation budgets on Qwen3-4B. OPASD consistently outperforms OPSD across all four benchmarks at 16K, 28K, and 38K tokens, while both methods generally benefit from larger generation budgets.

H Handling Privileged Attention

We compare three ways to handle teacher attention on positions unavailable to the student using Qwen3-1.7B in Table 6. Student-support projection restricts teacher attention to student-visible positions and renormalizes it. Redistribution transfers the excluded mass to shared prompt positions according to the teacher's attention, while shuffled redistribution randomly changes which shared positions receive that mass. Student-support projection achieves the highest average score of 34.02, compared with 31.39 for redistribution and 32.15 for shuffled redistribution. This advantage is consistent with projection preserving the teacher's relative attention among positions the student can use, whereas redistribution changes those relative weights and shuffling adds arbitrary variation. Theorem O.2 shows that projection is the unique visible-support distribution closest to the full teacher attention under the attention divergence.

I Token-Loss Clipping

We compare three clipping strategies for the token-distillation loss on Qwen3-1.7B in Table 7. Pointwise clipping operates on individual vocabulary-level contributions, whereas per-token clipping is applied to the complete KL divergence at each generated position. Using the same threshold of 0.05,

Table 5: Robustness of OPASD across random seeds. Avg@12 accuracy (%) across three Qwen3-4B training runs with seeds 1, 42, and 123. All runs use the same experimental configuration. The final rows report the mean, sample standard deviation, and 95% confidence interval across 3 seeds.

Seed / Statistic	AIME24	AIME25	AIME26	HMMT25	Average
Seed 1	63.27	50.83	55.33	34.72	51.04
Seed 42	63.33	50.27	55.56	33.89	50.76
Seed 123	63.54	50.63	54.87	34.36	50.85
Mean	63.38	50.58	55.25	34.32	50.88
Std	0.14	0.28	0.35	0.42	0.14
95% CI	±0.35	±0.70	±0.87	±1.03	±0.36

Table 6: Handling teacher-only attention mass. Student-support projection achieves a higher average score than redistribution or shuffled redistribution.

Method	AIME24	AIME25	AIME26	HMMT25	Average
Redistribution	40.00	31.38	**34.44**	19.72	31.39
Shuffled Redistribution	42.22	31.67	33.89	20.83	32.15
Student-Support Projection	**43.05**	**35.27**	**34.44**	**23.33**	**34.02**

Table 7: Effect of token-loss clipping. Per-token clipping achieves the highest average performance compared with point-wise clipping and no clipping.

Clipping Strategy	AIME24	AIME25	AIME26	HMMT25	Average
Point-wise Clipping	40.00	28.61	33.61	**21.11**	30.83
Per-token Clipping	**41.94**	32.50	**35.83**	**21.11**	**32.85**
No Clipping	40.00	**33.33**	34.72	20.83	32.22

Table 8: Comparison of distillation objectives on Qwen3-1.7B. Full OPASD achieves the highest average accuracy and outperforms both token-only OPSD and attention-only distillation.

Training Objective	AIME24	AIME25	AIME26	HMMT25	Average
Token only (OPSD)	36.70	28.33	32.78	18.33	29.04
Attention only	40.56	30.27	33.61	19.44	30.97
Token + Attention (OPASD)	**43.05**	**35.27**	**34.44**	**23.33**	**34.02**

per-token clipping achieves the highest average score of 32.85, compared with an average score of 32.22 without clipping and 30.83 with point-wise clipping. This is because point-wise clipping removes the teacher's strongest token preferences at each position, weakening the token signal that the attention objective complements. Per-token clipping instead preserves the full signal and only limits positions with very large divergence, keeping the two objectives better balanced.

J Comparison of Distillation Objectives

Table 8 compares token-only OPSD against attention-only OPASD and full OPASD on Qwen3-1.7B. Attention-only distillation achieves an average accuracy of 30.97 and exceeds token-only OPSD by 1.93 points. Full OPASD combines token- and attention-level distillation and achieves the highest average accuracy of 34.02. This represents an improvement of 3.05 points over attention-only distillation and 4.98 points over token-only OPSD. Token distillation aligns the student's next-token distribution with the teacher's, while attention distillation aligns its attention over student-visible positions with the projected teacher target. Together, these signals supervise both next-token predictions and attention to earlier positions, thus yielding higher accuracy than either objective alone.

K Controlling for Token-Loss Clipping

The original OPSD and OPASD configurations use different token-loss clipping strategies. To test whether this difference accounts for OPASD's improvement, we repeat token-only OPSD with the per-token clipping used by OPASD. As shown in Table 9, this control scores 29.09 on average, close to the original OPSD score of 29.04. With per-token clipping held fixed, OPASD scores 34.02 and outperforms the token-only control on all four benchmarks. The advantage therefore persists when clipping is matched. This is because per-token clipping does not change the token-level signal enough on its own to affect performance, and its benefit appears only when it balances the token loss against the attention objective. The improvement of OPASD over OPSD therefore comes from attention supervision rather than from the clipping strategy.

Table 9: Effect of attention supervision on Qwen3-1.7B with per-token clipping held fixed. The original OPSD configuration is included for reference.

Method	AIME24	AIME25	AIME26	HMMT25	Average
OPSD (point-wise clipping)	36.70	28.33	32.78	18.33	29.04
OPSD (per-token clipping)	35.83	29.56	32.80	18.16	29.09
OPASD (per-token clipping)	**43.05**	**35.27**	**34.44**	**23.33**	**34.02**

Question. Let A be the set of positive integer divisors of 2025. Let B be a randomly selected subset of A. The probability that B is a nonempty set with the property that the least common multiple of its element is 2025 is m/n, where m and n are relatively prime positive integers. Find m+n.

Base

Okay, so I need to find the probability that a randomly selected subset B of the set A ... $2025 = 3^4 \times 5^2$ ... Thus, we count the number of subsets of A that include at least one of each of the required elements: 81 and 25 . ... So, there are $2^{13} = 8192$ subsets of A that include at least one 81 and one 25 . ... m + n = 8192 + 32767 = 40959

OPSD

Okay, so I need to solve this problem where I have to find the probability ... At least one element in B must have exponent 4 on 3 (i.e., a = 4) ... At least one element in B must have exponent 2 on 5 (i.e., b = 2) ... These two conditions are not independent due to possible overlaps ... $|A_1 \cap A_2| = 2\text{^}(15 - 8) = 2^7$ ... Favorable subsets = $2^{15} - 2^{12} - 2^{10} + 2^7$ ... m + n = 217 + 256 = 473

OPASD

Okay, so I need to solve this probability problem where I have to find the probability ... The LCM of a subset of numbers is the product of the highest powers of all primes that appear in the factorizations of the numbers in the subset. ... The element $3^4 \times 5^2$ is the only one that appears in both groups, so we subtract it once. ... Thus, the number of such elements is 3 + 5 - 1 = 7 , and the number of subsets is $2^8 = 256$ . ... Valid subsets = $2^{15} - 2^{12} - 2^{10} + 2^8 = 32768 - 4096 - 1024 + 256 = 27904$ ... 27904/32768 = 109/128 ... m + n = 237

Figure 13: Reasoning on an AIME25 probability problem. The base model uses an incorrect counting condition, OPSD miscounts the overlap in inclusion-exclusion, and OPASD reaches the correct answer. Highlighted excerpts mark the decisive steps.

L Additional Qualitative Results

Figures 13 and 14 compare the instruction-tuned base model, token-only OPSD, and OPASD on an AIME25 probability problem and an AIME26 geometry problem. In the probability example, the base model treats the specific divisors 81 and 25 as necessary for the LCM condition, overlooking other divisors that supply the same prime powers. OPSD identifies the exponent conditions but miscounts their overlap, whereas OPASD accounts for the shared divisor and reaches the correct answer. In the geometry example, the base model's coordinates make the required reflex vertex convex, while OPSD's revised coordinates place B, C, and D on a line. OPASD preserves the nonconvex configuration and uses the resulting area constraint to count the valid side lengths. In both cases, the distinction is whether the constraints established during the setup are carried through to the final calculation. OPASD's attention supervision guides the student toward the earlier constraints needed to complete the final calculation.

M Hyperparameters

We report the training and evaluation configurations for our OPSD and OPASD experiments in Tables 10 and 11, respectively. For both methods, we use a Thinking-Mode-off student and a Thinking-Mode-on teacher. OPASD uses the same token-level distillation objective as OPSD and additionally applies attention distillation with $\lambda_{\text{attn}} = 0.5$.

N Notation Summary

Table 12 summarizes the notation used throughout the OPASD formulation.

O Theoretical Analysis

This appendix provides a theoretical characterization of OPASD's student-support projection. We show that, among all attention distributions supported on positions visible to the student, the pro-

Question. Let ABCDE be a nonconvex pentagon with internal angles angle A = angle E = 90° and angle B = angle D = 45°. Suppose that DE < AB, AE = 20, BC = 14√2, and points B,C, and D lie on the same side of line AE. Suppose further that AB is an integer with AB < 2026 and the area of pentagon ABCDE is an integer multiple of 16. Find the number of possible values of AB.

Base

Okay, let me try to tackle this problem step by step. First, let me make sure I understand the problem correctly. ... C = (14, x + 14) ... D = (20, x + 8) ... = 20x + 164 ... 20x + 164 ≡ 0 (mod 16) ... x ≡ 3 (mod 4) ... First term: 3 ... Last term: the largest number less than 2026 and congruent to 3 modulo 4, which is 2023 ... n = 2023 - 3/4 + 1 = 2020/4 + 1 = 505 + 1 = 506 ... 506

OPSD

Okay, so I need to solve this geometry problem about a nonconvex pentagon. ... Therefore coordinates C: (14, a-14). ... Point D(20,k)=D(20,a-20) ... Therefore the area is20*(a-10). ... a=10+4k ... a>20 ... The count is503 -3 +1 =501. ... 501

OPASD

Okay, so I need to solve this geometry problem about a nonconvex pentagon with specific angles and some side lengths. ... C = (14, b - 14) ... d = b - 8 ... 20|b - 8.2| ... This implies that 5b - 41 must be divisible by 4 ... b ≡ 1 (mod 4) ... b > 14 ... The smallest such b is 17 (since 17 ≡ 1 (mod 4)), and the largest is 2025. ... n = 2025 - 17/4 + 1 = 2008/4 + 1 = 502 + 1 = 503 ... 503

Figure 14: Reasoning on an AIME26 geometry problem with an integer-area constraint. The base model and OPSD derive incorrect constraints from their coordinate constructions, while OPASD reaches the correct count. Highlighted excerpts mark the decisive steps.

jected teacher attention uniquely minimizes the generalized Jensen–Shannon divergence to the full privileged-teacher attention distribution.

Table 10: Training hyperparameters for OPSD and OPASD.

Parameter	**OPSD**	**OPASD**
Training data	OpenThoughts-Math-30K (OPSD split)	
Teacher	Frozen teacher, conditioned on the reference solution	
Student thinking mode	Disabled	Disabled
Teacher thinking mode	Enabled	Enabled
LoRA rank r	64	64
LoRA scaling α	128	128
LoRA target modules	`q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`	
Optimizer	AdamW	AdamW
Learning rate	5×10^{-6}	5×10^{-6}
Effective batch size	64	64
Training steps	200	200
Maximum prompt length	2,048	2,048
Maximum completion length	8,192	8,192
Temperature	1.1	1.1
Top-p	0.95	0.95
Top-k	20	20
Token-loss weight λ_{tok}	1.0	1.0
Token divergence	Forward KL	Forward KL
Token-loss clipping	Point-wise	Per-token
Clipping threshold	0.05	0.05
Distillation temperature	1.1	1.1
Vocabulary support	Full vocabulary	Full vocabulary
Attention-loss weight λ_{attn}	-	0.5
Attention divergence	-	Jensen-Shannon

Table 11: Evaluation hyperparameters.

Parameter	Value
Maximum new tokens	16,384
Thinking mode	Enabled
Temperature	0.6
Top-p	0.95
Top-k	20
Min-p	0.0
Samples per prompt	12
Sampling seed	0

Table 12: Notation summary. Symbols used in the formulation and theoretical analysis of OPASD.

Symbol	Meaning	First Use
Data, models, and trajectories		
$\mathcal{S}$, N	Reasoning dataset and number of problem–solution pairs	Sec. 3.1
x_i, $y_i^\star$	Problem and its verified reference solution	Sec. 3.1
p_θ, p_{θ_0}	Student model with trainable parameters θ; frozen initial model used by the teacher	Sec. 3.1
p_S, p_T	Student and privileged teacher policies	Eq. (1)
$\hat{y}$, T	Student-generated reasoning trajectory and its length	Eq. (2)
t, $\hat{y}_{<t}$	Reasoning step and preceding trajectory prefix	Eq. (3)
$\mathcal{B}$	Training minibatch sampled from $\mathcal{S}$	Alg. 1
Token and attention distributions		
p_t^S, p_t^T	Student and teacher next-token distributions at reasoning step t	Eq. (3)
N_h, h	Number of attention heads and attention-head index	Eq. (5)
A^S, A^T	Head-averaged final-layer attention matrices of the student and teacher	Eq. (5)
Q_t^S, Q_t^T	Student and teacher query positions corresponding to token $\hat{y}_t$	Sec. 3.2
V_t^S, V_t^T	Student- and teacher-visible attention supports at reasoning step t	Sec. 3.2
$\mathcal{P}$	Reference-solution positions available only to the teacher	Sec. 3.2
A_t^S, A_t^T	Student and teacher attention distributions obtained by restricting the head-averaged attention rows to their respective visible supports and renormalizing	Sec. 3.2
$\widetilde{A}_t^T$	Teacher attention projected and renormalized onto the student-visible support	Eq. (6)
Objectives and hyperparameters		
D_{tok}	Token-distribution divergence	Eq. (4)
D_{attn}	Generalized Jensen–Shannon divergence between projected teacher attention and student attention	Eq. (8)
β, $M_t^{(\beta)}$	JSD interpolation coefficient and corresponding mixture distribution	Eq. (8)
$\ell_{\mathrm{attn}}(t)$	Per-step attention-distillation loss	Eq. (8)
$\mathcal{L}_{\mathrm{tok}}$	Token-level distillation objective	Eq. (4)
$\mathcal{L}_{\mathrm{attn}}$	Attention-level distillation objective	Eq. (9)
λ_{attn}	Attention-loss weight	Eq. (10)
$\mathcal{L}$	Final joint-distillation objective	Eq. (10)
Theoretical analysis		
ρ_t	Teacher attention mass assigned to reference-solution positions	App. O.1
D_β	Generalized Jensen–Shannon divergence used in the theoretical analysis	App. O.1
α_t	Effective JSD interpolation coefficient on the student-visible support	App. O.2
$C_\beta(\rho)$	Minimum divergence determined by the teacher attention mass outside the student-visible support	App. O.2

O.1 Preliminaries

Attention distributions. For a fixed problem x, reference solution $y^\star$, sampled trajectory $\hat{y}$, and reasoning step t, let V_t^S and V_t^T denote the student and teacher attention supports defined in Sec. 3.2.

The reference-solution positions available only to the teacher are denoted by $\mathcal{P}$, with

$$V_t^T = V_t^S \cup \mathcal{P}, \qquad V_t^S \cap \mathcal{P} = \varnothing.$$

Let A_t^T denote the teacher attention of Sec. 3.2, renormalized over V_t^T; this leaves Eq. (6) unchanged. We denote the teacher attention mass assigned to the reference-solution positions by $\rho_t = \sum_{i\in\mathcal{P}} A_t^T(i)$. The remaining mass on V_t^S is therefore $1-\rho_t$, and the student-support projection in Eq. (6) can be written as

$$1-\rho_t = \sum_{i\in V_t^S} A_t^T(i), \qquad \widetilde{A}_t^T(i) = \frac{A_t^T(i)}{1-\rho_t}, \quad i \in V_t^S.$$

For the nonempty student-visible support used in OPASD, the denominator in Eq. (6) is positive under softmax attention, so $\rho_t < 1$ and the projection is well defined. Equivalently, $\widetilde{A}_t^T$ is the teacher attention conditioned on the attended position lying in V_t^S. As shown in Eq. (7), this conditioning preserves the teacher's relative attention between student-visible positions.

Attention divergence. For the analysis below, we write $D_\beta(p,q)$ for the generalized Jensen–Shannon divergence used in Eq. (8),

$$D_\beta(p,q) = \beta D_{\mathrm{KL}}(p\|M) + (1-\beta)D_{\mathrm{KL}}(q\|M), \qquad M = \beta p + (1-\beta)q,$$

where $\beta \in (0,1)$ and p and q are probability distributions on the same finite set. We use the standard convention $0\log(0/a) = 0$ for $a \geq 0$. The divergence is finite because $M \geq \beta p$ and $M \geq (1-\beta)q$ entrywise. At $\beta = \frac{1}{2}$, D_β reduces to the Jensen–Shannon divergence (Lin, 1991). The teacher distribution appears as the first argument, matching the ordering used in Eq. (8).

Lemma O.1 (Nonnegativity of D_β)**.** *For $\beta \in (0,1)$, $D_\beta(p,q) \geq 0$, with equality if and only if $p = q$.*

Proof. By Gibbs' inequality, both KL terms in $D_\beta(p,q)$ are nonnegative and equal zero only when their two arguments are identical. Since $\beta \in (0,1)$, $D_\beta(p,q) = 0$ requires $p = M$ and $q = M$, where $M = \beta p + (1-\beta)q$. Hence $p = q$. Conversely, $p = q$ gives $D_\beta(p,q) = 0$. □

O.2 Optimality of Student-Support Projection

We now compare the projected teacher attention with any other attention distribution q supported on V_t^S. When comparing q with A_t^T over V_t^T, we take $q(i) = 0$ for $i \in \mathcal{P}$. Define

$$\alpha_t = \frac{\beta(1-\rho_t)}{1-\beta\rho_t}, \qquad C_\beta(\rho) = \beta\rho\log\frac{1}{\beta} + \beta(1-\rho)\log\frac{1-\rho}{1-\beta\rho} - (1-\beta)\log(1-\beta\rho).$$

Theorem O.2 (Optimality of the projected target)**.** *For $\beta \in (0,1)$ and any attention distribution q supported on V_t^S,*

$$D_\beta(A_t^T, q) = C_\beta(\rho_t) + (1-\beta\rho_t)D_{\alpha_t}(\widetilde{A}_t^T, q). \tag{11}$$

Hence $\widetilde{A}_t^T$ is the unique minimizer of $D_\beta(A_t^T, q)$ over all attention distributions supported on V_t^S. The minimum value $C_\beta(\rho_t)$ is zero if and only if $\rho_t = 0$ and is strictly increasing in ρ_t.

Proof. Write $s = 1-\rho_t$ and $Z = 1-\beta\rho_t$. Since $\rho_t < 1$ and $\beta \in (0,1)$, both are positive. Moreover,

$$\alpha_t = \frac{\beta s}{Z}, \qquad 1-\alpha_t = \frac{1-\beta}{Z},$$

and $Z - \beta s = 1-\beta > 0$. Hence $0 < \alpha_t < 1$.

Let $M = \beta A_t^T + (1-\beta)q$ be the mixture in $D_\beta(A_t^T, q)$. Since q assigns zero mass to $\mathcal{P}$, $M = \beta A_t^T$ on $\mathcal{P}$. On V_t^S, we have $A_t^T = s\widetilde{A}_t^T$. Defining

$$R = \alpha_t \widetilde{A}_t^T + (1-\alpha_t)q,$$

we obtain

$$M = \beta s\widetilde{A}_t^T + (1-\beta)q = ZR \qquad \text{on } V_t^S.$$

Since $R \geq \alpha_t \widetilde{A}_t^T$ and $R \geq (1-\alpha_t)q$ entrywise, the KL terms below are finite.

Splitting the teacher KL term over $\mathcal{P}$ and V_t^S gives

$$\begin{aligned}\beta D_{\mathrm{KL}}(A_t^T \| M) &= \beta\rho_t \log\frac{1}{\beta} + \beta s \sum_{i \in V_t^S} \widetilde{A}_t^T(i) \log \frac{s\widetilde{A}_t^T(i)}{ZR(i)} \\ &= \beta\rho_t \log\frac{1}{\beta} + \beta s \log\frac{s}{Z} + \beta s D_{\mathrm{KL}}(\widetilde{A}_t^T \| R).\end{aligned}$$

Since q assigns zero mass to $\mathcal{P}$, the second KL term is

$$(1-\beta) D_{\mathrm{KL}}(q\|M) = -(1-\beta)\log Z + (1-\beta) D_{\mathrm{KL}}(q\|R).$$

The terms independent of R are exactly $C_\beta(\rho_t)$. Using $\beta s = Z\alpha_t$ and $1-\beta = Z(1-\alpha_t)$, the remaining terms give

$$\begin{aligned}D_\beta(A_t^T, q) &= C_\beta(\rho_t) + Z\Big[\alpha_t D_{\mathrm{KL}}(\widetilde{A}_t^T \| R) + (1-\alpha_t) D_{\mathrm{KL}}(q\|R)\Big] \\ &= C_\beta(\rho_t) + (1-\beta\rho_t) D_{\alpha_t}(\widetilde{A}_t^T, q),\end{aligned}$$

which proves Eq. (11).

By Lemma O.1, $D_{\alpha_t}(\widetilde{A}_t^T, q) \geq 0$, with equality if and only if $q = \widetilde{A}_t^T$. Since $1-\beta\rho_t > 0$, the projected teacher attention is the unique minimizer.

Finally, $C_\beta(0) = 0$. Differentiating gives

$$C'_\beta(\rho) = \beta \log \frac{1-\beta\rho}{\beta(1-\rho)},$$

where the non-logarithmic terms cancel because $\beta(1-\rho) + (1-\beta) = 1-\beta\rho$. For $0 \leq \rho < 1$, $1-\beta\rho > \beta(1-\rho)$ because their difference is $1-\beta > 0$. Hence $C'_\beta(\rho) > 0$, so $C_\beta(\rho)$ is zero if and only if $\rho = 0$ and is strictly increasing in ρ. □

Theorem O.2 provides a theoretical justification for the student-support projection. Among all attention distributions supported on the student-visible positions, $\widetilde{A}_t^T$ uniquely minimizes the generalized Jensen–Shannon divergence to the full privileged-teacher attention, while the minimum divergence increases with the teacher attention mass assigned to positions unavailable to the student.