A-OPSD: ACTIVATION-SPACE ON-POLICY SELF-DISTILLATION FOR LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy self-distillation (OPSD) improves reasoning models by using a privileged teacher —the same model conditioned on both a problem and its reference solution— to supervise a student that observes only the problem. Existing OPSD methods transfer this privileged signal through dense token-level distribution matching over the student's own rollouts. While effective, this supervision acts only after the internal reasoning of the student and teacher has been projected through the language modeling head, and its cost scales with the full vocabulary. In this work, we ask whether the same signal can be transferred directly in the model's hidden-state space, and study a family of activation-space objectives for OPSD, from plain hidden-state matching to metrics derived from the local geometry of the OPSD loss. We find that even naively matching the teacher and student's hidden-states can match token-level OPSD, while a Gauss–Newton (Fisher) pullback of the OPSD KL through the LM head can outperform it. Across QWEN3 and OLMO-3 models from 1.7B to 8B parameters, Fisher A-OPSD (Activation-Space On-Policy Self-Distillation) improves mathematical reasoning performance by 7.6 points on average across models (51.0 to 58.6), with gains of up to 18.9 points on QWEN3-8B, while requiring substantially less supervision memory.

1 INTRODUCTION

Large Language Models (LLMs) have achieved strong performances across a wide range of tasks, driven in large part by their reasoning capabilities. While these capabilities are present to some extent in pretrained models (Wang & Zhou, 2024), they are typically elicited and strengthened through post-training. Common approaches include Supervised Fine-Tuning (SFT) on reasoning traces (Kim et al., 2023; Zelikman et al., 2022) and Reinforcement Learning, either from preferences (RLHF (Ouyang et al., 2022), DPO (Rafailov et al., 2024)) or from verifiable rewards, such as in GRPO (Shao et al., 2024). Knowledge distillation is another popular approach for improving models post-training, in which a stronger, typically larger, teacher supervises a student model (Wang et al., 2020; Ko et al., 2024; Xu et al., 2024; Hsieh et al., 2023). Such supervision is often performed by either minimizing a divergence between teacher and student output distributions or learning from teacher-generated trajectories (Xu et al., 2024; Li et al., 2025; Fang et al., 2026; Agarwal et al., 2024).

On-Policy Self-Distillation (OPSD) (Zhao et al., 2026) is one such approach. Rather than relying on a larger, more capable model, OPSD uses the same base model, conditioned on privileged information, as the teacher, while the student learns by matching the teacher's output distributions at reasoning decision points. This avoids the cost of a larger teacher and sidesteps the architectural, vocabulary, and tokenization mismatches that arise when the teacher and student are different models.

OPSD, however, operates entirely in output space: the distillation signal is applied only after the model's internal reasoning representations have been projected through the language modeling head into token space. While effective, this token-level matching pushes the student to mimic the teacher's output distribution without directly constraining the hidden states that produced it. Moreover, the supervision signal is defined over the full vocabulary, whose dimensionality can be tens of times larger than that of the model's hidden representations.

We propose Activation-space On-Policy Self-Distillation (A-OPSD), an extension of OPSD that distills directly in the model's hidden-state space. Instead of matching output distributions, A-OPSD aligns teacher and student representations at reasoning decision points, supervising the reasoning

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.

process before it is projected into token space (Figure 1). We study two activation-space objectives: direct hidden-state matching and Fisher-weighted objectives derived from the local geometry of the OPSD loss, and evaluate their ability to transfer reasoning across QWEN3 and OLMO-3 models on mathematical reasoning benchmarks.

We find that even basic ℓ_2 activation matching can recover or exceed token-space OPSD performance in several settings, and that accounting for the geometry induced by the language-modeling head is substantially more reliable. Fisher A-OPSD increases average mathematical reasoning performance from 38.6 to 50.2 on QWEN3-1.7B and from 50.0 to 68.9 on QWEN3-8B, while remaining competitive with OPSD on QWEN3-4B and OLMO-3-7B. Averaged across all four model settings, with an overall improvement from 51.0 to 58.6, while reducing the memory required by the distillation supervision. Our contributions are:

1. We introduce A-OPSD, an extension of OPSD that transfers privileged reasoning behavior by directly aligning teacher and student hidden representations rather than their vocabulary distributions.
2. We derive a Fisher/Gauss–Newton objective by pulling back the OPSD KL divergence through the frozen LM head, providing a principled activation-space approximation to token-space distillation.
3. We show that naive hidden-state matching can recover much of token-space OPSD performance while reducing the dimensionality of the distillation signal from the vocabulary size to the hidden dimension.
4. Across QWEN3 and OLMO-3 models from 1.7B to 8B parameters, Fisher A-OPSD improves the mean three-benchmark score from 51.0 to 58.6 over standard OPSD, with gains of up to 18.9 points, while using substantially less supervision memory.

2 Background and Related Work

2.1 On-Policy Self-Distillation

On-Policy Self-Distillation (OPSD), introduced by Zhao et al. (2026), performs reasoning distillation using a single base model as both teacher and student, differing only in what each is conditioned on. Let p_θ denote the base policy with parameters θ. Given a reasoning dataset $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^{N}$ of problems x_i and their reference solutions y_i, OPSD defines a student policy $p_S(\cdot \mid x) = p_\theta(\cdot \mid x)$ conditioned on the problem alone, and a teacher policy $p_T(\cdot \mid x, y) = p_\theta(\cdot \mid x, y)$ that is additionally conditioned on the reference reasoning and/or answer y.

With on-policy training, the student first samples its own rollout $\hat{y} \sim p_S(\cdot \mid x)$, and supervision is applied along the tokens the student actually generates, so the training distribution matches the student's own generative distribution rather than that of fixed teacher traces. At each step n, both policies are evaluated on this shared prefix to produce next-token distributions over the vocabulary $\mathcal{V}$,

$$p_S^{(n)} = p_S(\cdot \mid x, \hat{y}_{<n}), \qquad p_T^{(n)} = p_T(\cdot \mid x, y, \hat{y}_{<n}),$$

and OPSD minimizes a per-token divergence $D\Big(p_T^{(n)} \,\|\, p_S^{(n)}\Big)$ between them. The teacher distribution is treated as a fixed target: gradients are backpropagated through the student's logits only, so the model distills from itself. Zhao et al. (2026) obtain their strongest results with the forward KL divergence,

$$\mathrm{KL}\Big(p_T^{(n)} \,\|\, p_S^{(n)}\Big) = \sum_{v \in \mathcal{V}} p_T^{(n)}(v) \left(\log p_T^{(n)}(v) - \log p_S^{(n)}(v)\right).$$

The full objective is the expected divergence over problems and on-policy student rollouts,

$$\mathcal{L}_{\text{OPSD}} = \mathbb{E}_{(x,y)\sim\mathcal{S}} \;\; \mathbb{E}_{\hat{y}\sim p_S(\cdot|x)} \sum_{n=1}^{|\hat{y}|} \mathrm{KL}\big(p_T(\cdot \mid x, y, \hat{y}_{<n}) \,\big\|\, p_S(\cdot \mid x, \hat{y}_{<n})\big)\,.$$

This supervision is applied entirely in output space, measured only after each policy's hidden state has been projected through the language-modeling head into a distribution over $\mathcal{V}$. The internal representations that produce these distributions are left unconstrained, a gap our method addresses directly.

2.2 Output-Space Distillation

Recent methods vary the source of privileged context and the training objective. Shenfeld et al. (2026) condition the self-teacher on an expert demonstration and distill its predictions on student-generated trajectories. Hübotter et al. (2026) condition instead on environment feedback, using the induced token-level distribution shift as dense credit assignment for reinforcement learning. Penaloza et al. (2026) study privileged-information distillation in multi-turn agentic settings, introducing both a joint teacher–student objective and a related on-policy formulation, and Liu et al. (2026) combine full-vocabulary self-distillation with verifier-based policy-gradient updates. In each case, information available to a context-conditioned instance of the model is transferred back into the policy through its output distribution.

Another line of work investigates when this transfer succeeds. He et al. (2026) find that the initial student–teacher gap strongly predicts the eventual gain from OPSD, so the usefulness of a privileged context can often be estimated before training. Others identify failure modes when the teacher's context alters the reasoning being distilled: Kim et al. (2026) show that rich conditioning can hurt out-of-distribution reasoning, while Kaur et al. (2026) find that it suppresses branching and reconsideration in thinking models, particularly over long traces.

Across all these works, supervision itself remains defined in output space: teacher and student representations are first projected through the language-modeling head, and learning is driven by their resulting token distributions. Our work asks an orthogonal question: whether the same privileged signal can instead be transferred directly between the internal representations that produce those distributions.

2.3 Activation-Space Distillation

A separate line of work distills knowledge through internal representations rather than output distributions (Romero et al., 2015; Heo et al., 2018; Wang et al., 2020; Aguilar et al., 2020; Dasgupta & Cohn, 2025; Saadi & Wang, 2026). These methods supervise the student using information from the teacher's hidden representations, rather than relying only on its output distribution. Because teacher and student are typically different models, much of this work focuses on reconciling mismatched representation spaces. Romero et al. (2015) and Heo et al. (2018), for example, use learned mappings between teacher and student activations, while other methods instead match attention relations, representation similarities, or selected teacher subspaces (Wang et al., 2020; Aguilar et al., 2020; Dasgupta & Cohn, 2025; Saadi & Wang, 2026). Under OPSD, teacher and student are differently conditioned instances of the same model, so their hidden states already lie in the same representation space and can be compared directly. This removes the need for learned projections or dimension-invariant matching objectives.

The remaining question is how those differences in activation should be measured. Existing methods typically choose a fixed representational distance, such as MSE or cosine similarity, independently of the output-space objective. We instead derive the activation-space geometry directly from the OPSD KL divergence.

3 Activation-Space OPSD

3.1 Naive Activation Matching

Unlike the methods in Section 2.3, self-distilling in activation space requires no projection of hidden states, since the teacher and student are the same model. Instead, for a model $\mathcal{M}$ and its two versions, the privileged teacher $\mathcal{M}_T$ and the base student $\mathcal{M}_S$, let $h_S^{(n)}$ and $h_T^{(n)}$ be the last-layer activations (the input to the language-modeling head) at decision step n, produced from the shared prefix $(x, \hat{y}_{<n})$ for the student and $(x, y, \hat{y}_{<n})$ for the teacher. One could optimize the student to match the teacher's activations directly with a normalized mean-squared objective:

$$\mathcal{L}_{\text{MSE}} = \mathbb{E}_{(x,y)\sim\mathcal{S}}\, \mathbb{E}_{\hat{y}\sim p_S(\cdot|x)} \sum_{n=1}^{|\hat{y}|} \frac{1}{H} \left\| \bar{h}_S^{(n)} - \bar{h}_T^{(n)} \right\|_2^2, \qquad \bar{h} = \frac{h}{\|h\|_2}.$$

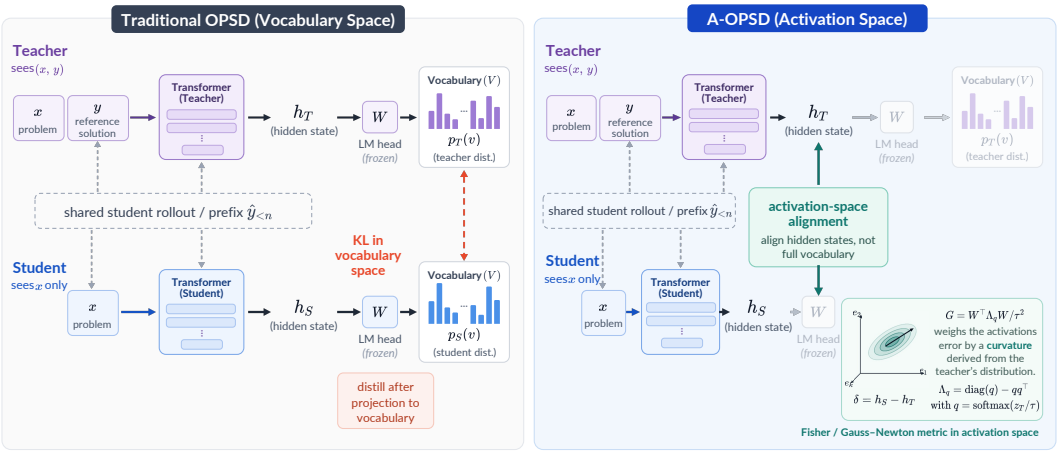


Figure 1: **Overview of OPSD and A-OPSD.** In both methods, the teacher is conditioned on the problem and reference solution while the student observes only the problem, with both evaluated on a shared student-generated prefix. Traditional OPSD distills the teacher through a KL objective at the vocabulary space, whereas A-OPSD aligns teacher and student hidden representations directly and uses a Fisher/Gauss–Newton metric to emphasize activation differences that meaningfully affect the teacher's predictions.

where $\bar{h}$ is the ℓ_2-normalized activation and H is the hidden size. The loss is averaged over the student's own decision tokens.

While this naive geometric matching already recovers much of OPSD's performance, as we show in Section 4.2, it has two main drawbacks: the first is that such a Euclidean metric weights all H directions of the residual equally, regardless of how they contribute to the model's output. Second, in such an objective, the student is asked to produce activations from privileged information it does not have, and therefore, even after the student learns and exhausts all the useful signals from the teacher, the delta $\bar{h}_S^{(n)} - \bar{h}_T^{(n)}$ never vanishes, which pushes the student to take random steps that degrade performance (Section 4.3).

3.2 Fisher Metric Objective

To address the first drawback, we replace the geometric metric with one derived from the local geometry of the OPSD objective itself. The language-modeling head is a fixed linear read-out $z = Wh$, $W \in \mathbb{R}^{V \times H}$; following Zhao et al. (2026), we use LoRA adapters that target only the attention and MLP projections (`q,k,v,o,gate,up,down`), W is identical and frozen across the teacher and student, which is what makes the following construction exact. The next-token distribution at a hidden state h is

$$p(\cdot \mid h) = \mathrm{softmax}(Wh/\tau),$$

and, holding the teacher's state $g = h_T^{(n)}$ fixed, we view OPSD's forward KL at a single decision step as a function of the student's state $h = h_S^{(n)}$:

$$\Phi(h) = \mathrm{KL}\big(p(\cdot \mid g) \,\big\|\, p(\cdot \mid h)\big).$$

Since $\Phi(h) \geq 0$ with equality when $p(\cdot \mid h) = p(\cdot \mid g)$, the point $h = g$ is a global minimum. The Taylor expansion of Φ around that point with $\delta = h - g$ gives us:

$$\Phi(g + \delta) = \Phi(g) + \nabla_h \Phi\big|_{h=g}^{\top} \delta + \tfrac{1}{2}\, \delta^\top \nabla_h^2 \Phi\big|_{h=g}\, \delta + O(\|\delta\|^3).$$

The constant term vanishes, $\Phi(g) = 0$. The linear term also vanishes since $h = g$ is a minimum:

$$\nabla_h \Phi = \tfrac{1}{\tau} W^\top \big(p(\cdot \mid h) - p(\cdot \mid g)\big), \qquad \nabla_h \Phi\big|_{h=g} = 0.$$

The quadratic term is therefore the leading contribution. Differentiating the gradient once more,

$$\nabla_h^2 \Phi = \tfrac{1}{\tau^2}\, W^\top \Big(\mathrm{diag}\big(p(\cdot \mid h)\big) - p(\cdot \mid h)\, p(\cdot \mid h)^\top\Big) W,$$

which at $h = g$ is evaluated at the teacher's distribution. Writing $q = p(\cdot \mid g)$ and $\Lambda_q = \mathrm{diag}(q) - qq^\top$, we obtain the Gauss–Newton (Fisher) metric of the read-out,

$$G = \tfrac{1}{\tau^2}\, W^\top \Lambda_q W, \qquad \Phi(g + \delta) = \tfrac{1}{2}\, \delta^\top G\, \delta + O(\|\delta\|^3).$$

We define the activation-space objective as this quadratic form, with the teacher's state and the metric detached:

$$\ell_{\mathrm{Fisher}}(h) = \tfrac{1}{2}\, \delta^\top G\, \delta.$$

Expanding this quadratic form gives us an interpretable variance identity. Writing $d_v = w_v^\top \delta / \tau$ for the perturbation that δ induces in token v's logit,

$$\tfrac{1}{2}\, \delta^\top G\, \delta = \tfrac{1}{2}\Big[\sum_v q_v d_v^2 - \big(\textstyle\sum_v q_v d_v\big)^2\Big] = \tfrac{1}{2}\, \mathrm{Var}_{v \sim q}\big[\, w_v^\top \delta / \tau\,\big].$$

The loss is the variance, under the teacher's next-token distribution, of the logit perturbation caused by the student's representation error

$$\mathcal{L}_{\mathrm{Fisher}} = \mathbb{E}_{(x,y)\sim\mathcal{S}}\, \mathbb{E}_{\hat{y}\sim p_S(\cdot|x)} \sum_{n=1}^{|\hat{y}|} \tfrac{1}{2}\, \mathrm{Var}_{v\sim q^{(n)}}\big[\, w_v^\top \delta^{(n)} / \tau\,\big], \qquad \delta^{(n)} = h^{(n)} - g^{(n)},$$

averaged over the student's own decision tokens, where $q^{(n)} = p(\cdot \mid g^{(n)})$ is the teacher's (detached) next-token distribution.

Top-k Constraint. Evaluating the variance over the full vocabulary requires $W\delta^{(n)}$: an $O(VH)$ matmul per rollout token whose V-dimensional output is retained for the backward pass, which is prohibitive. Instead, we restrict the variance to the teacher's top-k tokens and renormalize $\tilde{q}_j = q_{v_j} / \sum_{l \le k} q_{v_l}$:

$$\ell_{\mathrm{Fisher}}^{(n)} = \tfrac{1}{2}\Big[\sum_{j\le k} \tilde{q}_j d_j^2 - \big(\textstyle\sum_{j\le k} \tilde{q}_j d_j\big)^2\Big], \qquad d_j = w_{v_j}^\top \delta^{(n)} / \tau.$$

3.3 Per-Token Loss Cap

The Fisher metric resolves the first of the MSE loss, but not the second. Part of the teacher's predictive advantage stems from the reference solution y whose resulting hidden-state is genuinely uncomputable from the student's input, so even at convergence, some residual remains and the per-token loss settles at a significant positive value. This residual is not spread uniformly across the trajectory: it concentrates in a small subset of stubborn, high-loss tokens—the decision points where the teacher's advantage rests most heavily on the unseen solution. Under a plain quadratic these few tokens dominate the gradient, so prolonged training keeps forcing the student toward an unattainable target on them grinding into the weights. Accuracy therefore peaks once the learnable signal has been absorbed and then collapses with further training.

We therefore cap the per-token loss, clipping each token's Fisher loss at a fixed value c:

$$\mathcal{L}_{\mathrm{Fisher}}^{c} = \mathbb{E}_{(x,y)\sim\mathcal{S}}\, \mathbb{E}_{\hat{y}\sim p_S(\cdot|x)} \sum_{n=1}^{|\hat{y}|} \min\big(\ell_{\mathrm{Fisher}}^{(n)},\ c\big).$$

Above the cap, the objective is flat and its gradient vanishes, so the stubborn high-residual tokens can no longer dominate the update or be ground into the weights; the learnable tokens, whose loss lies below c, continue to train normally.

Table 1: **Accuracy across model sizes and methods.** avg@12 (%) for AIME24/HMMT25 and avg@4 (%) for MATH-500. All methods train for 100 steps under the shared configuration of Section 4.1. Every cell is the mean ± standard deviation over three independent training seeds.

Model	Method	MATH-500	AIME24	HMMT25	Avg
QWEN3-1.7B	Base	71.7±0.1	13.8±0.2	4.5±0.6	30.0
	SFT	77.9±4.7	20.5±0.7	8.1±0.7	35.5
	OPSD	76.0±1.1	26.9±1.2	12.8±1.0	38.6
	A-OPSD$_{\text{MSE}}$	78.7±1.7	27.1±4.1	14.4±0.3	40.1
	A-OPSD$_{\text{Fisher}}$	**88.7**±0.7	**41.6**±1.0	**20.4**±0.3	**50.2**
QWEN3-4B	Base	83.7±0.1	23.6±1.9	11.9±1.0	39.7
	SFT	81.4±0.4	21.9±2.7	12.8±1.3	38.7
	OPSD	89.9±0.9	**53.6**±1.3	**30.9**±1.1	**58.1**
	A-OPSD$_{\text{MSE}}$	86.2±0.7	26.4±1.0	14.7±0.7	42.4
	A-OPSD$_{\text{Fisher}}$	**91.7**±1.8	50.5±6.5	29.6±1.4	57.3
OLMO-3-7B	Base	91.3±0.2	50.5±4.3	25.1±1.4	55.6
	SFT	92.0±0.5	52.9±1.9	**26.7**±1.5	57.2
	OPSD	**92.5**±0.4	53.6±1.0	26.1±0.5	57.4
	A-OPSD$_{\text{MSE}}$	91.9±0.2	51.8±0.2	24.5±1.4	56.1
	A-OPSD$_{\text{Fisher}}$	**92.5**±0.6	**55.3**±1.3	26.0±1.7	**57.9**
QWEN3-8B	Base	84.1±0.7	25.8±1.0	11.2±0.2	40.4
	SFT	80.6±0.1	22.2±0.8	9.8±0.7	37.5
	OPSD	85.2±7.6	46.8±5.4	18.1±3.1	50.0
	A-OPSD$_{\text{MSE}}$	**95.6**±0.2	**73.5**±1.5	40.6±1.3	**69.9**
	A-OPSD$_{\text{Fisher}}$	95.2±0.7	70.3±0.5	**41.1**±1.1	68.9

4 MAIN EXPERIMENTS

4.1 SETUP

Models. We experiment with 4 models from the families of QWEN3 (Yang et al., 2025) and OLMO-3 (Olmo et al., 2025), with sizes ranging from 1.7B to 8B parameters.

Dataset. We train all models and methods on the OPENTHOUGHTS (Guha et al., 2025) dataset using only the mathematical reasoning subset.

Benchmarks. All models and methods are evaluated across 3 mathematical reasoning benchmarks, varying in the level of difficulty. Namely, we use: MATH-500 (Lightman et al., 2023), AMIE24 (Zhang & Math-AI, 2024), and HMMT25 (Dekoninck et al., 2026).

Baselines. We compare our methods against OPSD as introduced by Zhao et al. (2026), and a Supervised Fine-Tuning (SFT) baseline (Wei et al., 2022).

All training runs are done using LORA (Hu et al., 2021) with an effective batch size of 32. For A-OPSD and OPSD, the teacher is conditioned on the entire reference solution from the training dataset. We report Avg@4 for MATH-500 and Avg@12 for AMIE24 and HMMT25. For QWEN3 models, we report non-thinking mode performance; we report on thinking mode in Appendix A. Complete details of hyperparameters and the evaluation protocol are available in Appendix C.

4.2 MAIN RESULTS

Table 1 shows that activation-space distillation is competitive with, and often stronger than, standard OPSD across model scales. Fisher A-OPSD delivers the clearest overall gains: averaged across the four model settings, the mean three-benchmark score improves from 51.0 with OPSD to 58.6, a $+7.6$-point increase. The largest improvements occur on QWEN3-1.7B, where Fisher A-OPSD

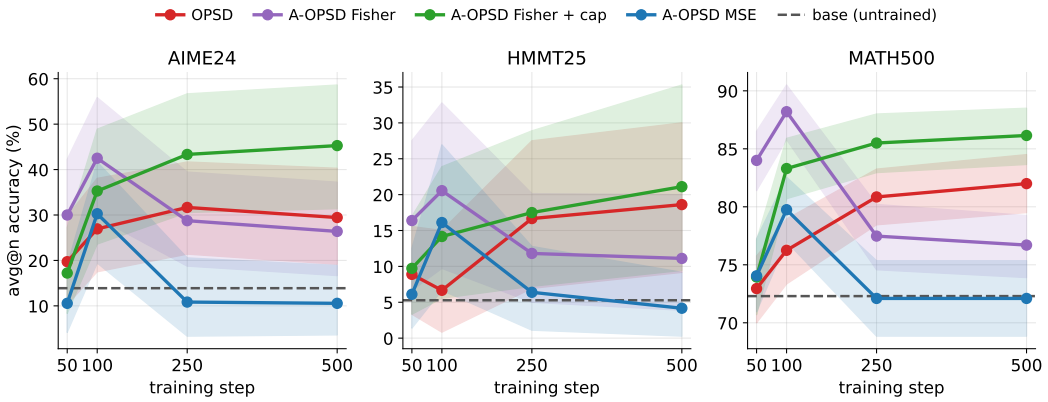


Figure 2: **Downstream performance across training on QWEN3-1.7B.** Fisher A-OPSD improves rapidly during early training, reaching its strongest performance around step 100, but degrades with continued optimization. Capping the Fisher metric stabilizes training and maintains or improves performance through later checkpoints. Shaded regions denote confidence intervals.

raises the average from 38.6 to 50.2, and on QWEN3-8B, where it reaches 68.9 compared with 50.0 for OPSD. On OLMO-3-7B, Fisher A-OPSD is also slightly stronger overall (57.9 vs. 57.4), while on QWEN3-4B it remains comparable to OPSD (57.3 vs. 58.1). Naive MSE activation matching is more variable across scales. It slightly improves over OPSD on QWEN3-1.7B and substantially outperforms it on QWEN3-8B, reaching the highest overall score in that setting (69.9), but lags behind on QWEN3-4B. This contrast suggests that direct activation matching can be effective, but incorporating the local output-space geometry through the Fisher objective provides substantially more consistent behavior across models.

4.3 LONGER HORIZON TRAINING

Our main results in Table 1 evaluate all methods after 100 training steps. To study whether these gains persist with continued optimization, we extend training on Qwen3-1.7B to 500 steps on a single seed. As shown in Figure 2, the uncapped activation-space objectives peak early and then deteriorate. MSE drops sharply after step 100 across all three benchmarks, while Fisher A-OPSD falls from 42.5 to 26.5 on AIME24, 20.5 to 11.0 on HMMT25, and 88.0 to 76.7 on MATH-500 by step 500. Thus, the strong performance observed at the 100-step evaluation point is not maintained under prolonged training.

The per-token Fisher cap largely removes this degradation. Rather than peaking and declining, the capped variant continues to improve after step 100, reaching 45.3 on AIME24, 21.0 on HMMT25, and 86.2 on MATH-500 at step 500. Figure 3 provides a corresponding optimization view: the uncapped Fisher objective largely plateaus around step 100, yet its gradient norm remains near or above the clipping threshold throughout the remainder of training. The capped objective suppresses the persistent high-loss tokens responsible for these updates, allowing longer training without the collapse observed with uncapped Fisher. In Appendix B, we show that minimizing $\mathcal{L}_{\text{Fisher}}$ also reduces the token-space forward KL, reaching a lower value than OPSD itself on held-out problems.

4.4 MEMORY EFFICIENCY

A second advantage of activation-space distillation is that the supervision signal no longer needs to be represented over the full vocabulary. Figure 4 isolates the memory required by the distillation objective. Although all objectives scale approximately linearly with the number of decision tokens, activation-space matching has a substantially smaller constant factor. At $N = 4096$ decision tokens on QWEN3-1.7B, standard OPSD requires 10,692 MiB for the supervision term, compared with 2,213 MiB for Fisher A-OPSD and only 192 MiB for direct MSE matching.

The same advantage persists across model sizes. Because the Qwen3 models share the same vocabulary, the memory cost of token-space OPSD remains approximately 10.7 GiB across 1.7B,

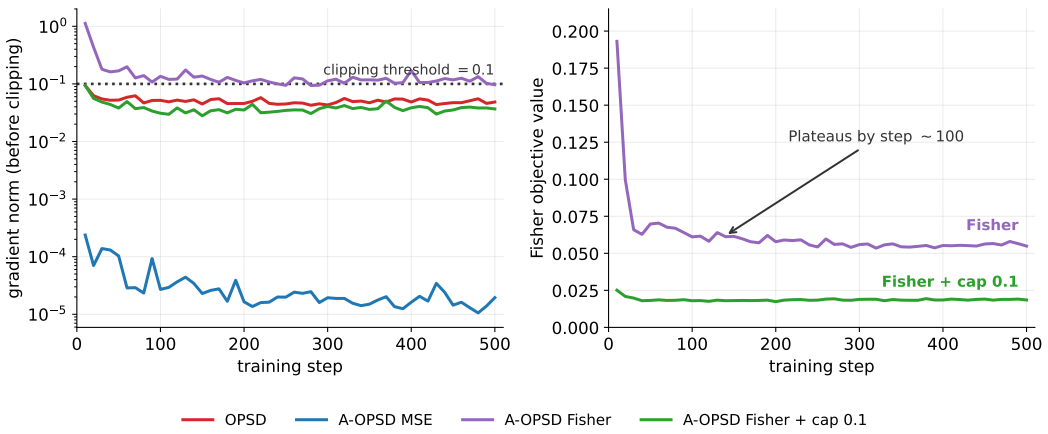


Figure 3: **Optimization dynamics on QWEN3-1.7B.** Left: Gradient norms. Fisher remains near or above the 0.1 clipping threshold throughout training, whereas the other objectives remain below it. Right: The Fisher objective largely plateaus by step 100, despite its gradients remaining clipping-limited. As shown in Figure 2, continued Fisher training beyond this point coincides with degradation in downstream reasoning performance.

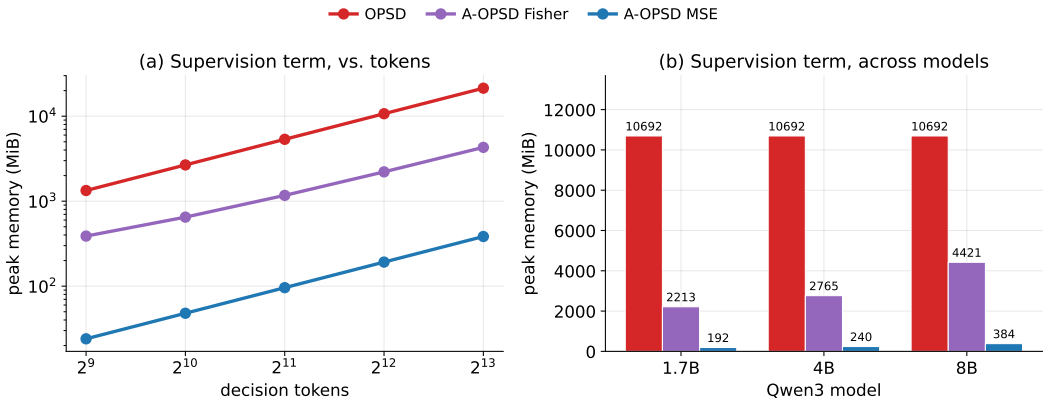


Figure 4: **Supervision-memory scaling.** Left: Peak memory of the supervision term as the number of decision tokens increases on Qwen3-1.7B. Right: Supervision memory across Qwen3 model sizes at $N = 4096$ tokens. Fisher objective uses $k = 64$ for the top-k tokens. Activation-space objectives require substantially less memory than token-space OPSD.

4B, and 8B models. In contrast, the activation-space objectives scale with the hidden representation. Fisher A-OPSD uses 2,213, 2,765, and 4,421 MiB across the three model sizes, corresponding to roughly $4.8\times$, $3.9\times$, and $2.4\times$ less supervision memory than OPSD. Direct MSE is cheaper still, requiring only 192–384 MiB.

These results highlight the main computational benefit of moving distillation before the vocabulary projection: A-OPSD replaces dense vocabulary-space supervision with objectives whose memory footprint is tied primarily to the hidden-state dimension. The Fisher objective introduces additional cost relative to plain activation matching, but retains a substantial memory advantage over standard OPSD while providing the stronger performance observed in Table 1.

5 TOP-k ABLATIONS

The Fisher objective requires weighting activation errors by the local geometry induced by the teacher's output distribution. Computing this geometry over the full vocabulary is prohibitively

Table 2: **Fisher top-k ablation** on QWEN3-1.7B. AIME24 is measured at step 100. Memory is the supervision-term peak at N=4096 decision tokens, and relative to OPSD's peak of 10,692 MiB.

k	AIME24	memory (MiB)	vs. OPSD
4	37.8	192	55.7×
8	39.4	305	35.1×
16	43.9	577	18.5×
32	43.3	1123	9.5×
64	42.5	2213	4.8×
128	46.7	4395	2.4×
$V = 151{,}936$	*out of memory*		

Table 3: **Top-k restriction of the OPSD KL.** AIME24 accuracy on QWEN3-1.7B, changing only the KL support to the teacher's top-64 tokens.

Method	Step 25	Step 50	Step 75	Step 100	Best
Base (untrained)			—		13.9
OPSD, top-64 KL	13.3	15.3	17.5	19.4	19.4
OPSD, full vocabulary			—		**26.9**

expensive, so we approximate it using only the teacher's top-k tokens. Table 2 studies this approximation on QWEN3-1.7B. Performance is already strong with relatively small k: increasing k from 4 to 16 improves AIME24 from 37.8 to 43.9, while values between $k = 16$ and $k = 64$ remain similar. Larger k increases memory linearly, reaching 4,395 MiB at $k = 128$, while constructing the full-vocabulary Fisher metric is out of memory.

Importantly, the effectiveness of top-k is specific to the Fisher approximation rather than simply discarding the tail of the vocabulary distribution. Table 3 applies the same restriction directly to the standard OPSD KL objective. Restricting OPSD to the teacher's top-64 tokens reaches only 19.4 on AIME24, compared with 26.9 for the full-vocabulary KL under the same training setup. In contrast, Fisher A-OPSD with $k = 64$ reaches 42.5. Thus, a small top-k subset is sufficient for estimating the local output-space geometry used to weight activation errors, but is not sufficient to replace the full output distribution when distillation is performed directly in vocabulary space.

6 CONCLUSION

We introduced A-OPSD, an activation-space formulation of on-policy self-distillation that transfers the privileged teacher signal before projection into vocabulary space. Because the teacher and student are differently conditioned instances of the same model, their hidden states can be compared directly without learned projections. Building on this observation, we studied both naive activation matching and a Fisher/Gauss–Newton objective obtained by pulling back the OPSD KL through the frozen LM head. Across QWEN3 and OLMO-3 models from 1.7B to 8B parameters, Fisher A-OPSD improves average performance on mathematical reasoning benchmarks from 51.0 to 58.6 over standard OPSD, while substantially reducing the memory required by the supervision term.

Our experiments also highlight an important optimization issue in activation-space distillation. Both naive MSE and uncapped Fisher objectives can degrade under longer training, even after the supervision objective has largely converged. For Fisher A-OPSD, we address this with a simple per-token loss cap that removes gradients from persistent high-loss tokens and stabilizes longer-horizon training.

Together, these results suggest that privileged reasoning signals do not need to be distilled through the full output distribution: they can be transferred effectively in activation space, while retaining the output-space geometry that makes OPSD effective and reducing its supervision cost.

LIMITATIONS

Our work has several limitations. First, we do not attempt to address the broader challenges associated with applying OPSD to thinking models, and instead focus primarily on mathematical reasoning tasks. As a result, it remains unclear how well A-OPSD generalizes to other reasoning domains or settings with substantially different generation behavior. Second, while we explore several variants of the activation-space objective and key training parameters, a more extensive hyperparameter search may lead to further improvements. Our exploration was limited by the computational cost of running repeated training and evaluation across multiple settings. Future work could investigate broader task domains, larger models, and a more systematic search over objective design and training hyperparameters.

AI USE STATEMENT

We used AI-based tools and coding agents during the development of this work to assist with writing, code implementation, debugging, and running experiments. These tools were used primarily to improve efficiency during implementation and manuscript preparation. They were not used to generate the core research idea, formulate the proposed method, or determine the main research direction. All methodological decisions, experimental design, analysis, and interpretation of the results were carried out by the authors.

REPRODUCIBILITY STATEMENT

Much of our experimental setup builds on Zhao et al. (2026) and the public repository released with their work. We reuse their implementation and evaluation setup where applicable, while introducing the modifications required for A-OPSD and the activation-space objectives studied in this work. Details of our experimental setup, training configurations, hyperparameters, and evaluation settings are provided in Appendix C. We also report the relevant implementation choices needed to reproduce our main experiments and ablations.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes, 2024. URL https://arxiv.org/abs/2306.13649.

Gustavo Aguilar, Yuan Ling, Yu Zhang, Benjamin Yao, Xing Fan, and Chenlei Guo. Knowledge distillation from internal representations, 2020. URL https://arxiv.org/abs/1910.03723.

Sayantan Dasgupta and Trevor Cohn. Improving language model distillation through hidden state matching. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=IcVSKhVpKu.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: Matharena as an evaluation platform for mathematics with llms. 2026. URL https://arxiv.org/abs/2605.00674.

Luyang Fang, Xiaowei Yu, Jiazhang Cai, Yongkai Chen, Shushan Wu, Zhengliang Liu, Zhenyuan Yang, Haoran Lu, Xilin Gong, Yufang Liu, Terry Ma, Wei Ruan, Ali Abbasi, Jing Zhang, Tao Wang, Ehsan Latif, Weihang You, Hanqi Jiang, Wei Liu, Wei Zhang, Soheil Kolouri, Xiaoming Zhai, Dajiang Zhu, Wenxuan Zhong, Tianming Liu, and Ping Ma. Knowledge distillation and dataset distillation of large language models: Emerging trends, challenges, and future directions, 2026. URL https://arxiv.org/abs/2504.14772.

Etash Guha, Ryan Marten, Sedrick Keh, Negin Raoof, Georgios Smyrnis, Hritik Bansal, Marianna Nezhurina, Jean Mercat, Trung Vu, Zayne Sprague, Ashima Suvarna, Benjamin Feuer, Liangyu Chen, Zaid Khan, Eric Frankel, Sachin Grover, Caroline Choi, Niklas Muennighoff, Shiye Su,

Wanjia Zhao, John Yang, Shreyas Pimpalgaonkar, Kartik Sharma, Charlie Cheng-Jie Ji, Yichuan Deng, Sarah Pratt, Vivek Ramanujan, Jon Saad-Falcon, Jeffrey Li, Achal Dave, Alon Albalak, Kushal Arora, Blake Wulfe, Chinmay Hegde, Greg Durrett, Sewoong Oh, Mohit Bansal, Saadia Gabriel, Aditya Grover, Kai-Wei Chang, Vaishaal Shankar, Aaron Gokaslan, Mike A. Merrill, Tatsunori Hashimoto, Yejin Choi, Jenia Jitsev, Reinhard Heckel, Maheswaran Sathiamoorthy, Alexandros G. Dimakis, and Ludwig Schmidt. Openthoughts: Data recipes for reasoning models, 2025. URL https://arxiv.org/abs/2506.04178.

Tommy He, Jerome Sieber, and Matteo Saponati. A predictive law for on-policy self-distillation from world feedback, 2026. URL https://arxiv.org/abs/2605.30070.

Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons, 2018. URL https://arxiv.org/abs/1811.03233.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes, 2023. URL https://arxiv.org/abs/2305.02301.

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, 2021. URL https://arxiv.org/abs/2106.09685.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation, 2026. URL https://arxiv.org/abs/2601.20802.

Simran Kaur, Narutatsu Ri, Yinghui He, Liam Fowl, and Sanjeev Arora. Rethinking on-policy self-distillation for thinking models. *arXiv preprint arXiv:2607.05184*, 2026. URL https://arxiv.org/abs/2607.05184.

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?, 2026. URL https://arxiv.org/abs/2603.24472.

Seungone Kim, Se June Joo, Doyoung Kim, Joel Jang, Seonghyeon Ye, Jamin Shin, and Minjoon Seo. The cot collection: Improving zero-shot and few-shot learning of language models via chain-of-thought fine-tuning, 2023. URL https://arxiv.org/abs/2305.14045.

Jongwoo Ko, Sungnyun Kim, Tianyi Chen, and Se-Young Yun. Distillm: Towards streamlined distillation for large language models, 2024. URL https://arxiv.org/abs/2402.03898.

Zhuochun Li, Yuelyu Ji, Rui Meng, and Daqing He. Learning from committee: Reasoning distillation from a mixture of teachers with peer-review. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, Vienna, Austria, July 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.217. URL https://aclanthology.org/2025.findings-acl.217/.

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

Yifeng Liu, Shiyuan Zhang, Yifan Zhang, and Quanquan Gu. Self-distilled policy gradient, 2026. URL https://arxiv.org/abs/2606.04036.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari

Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Emiliano Penaloza, Dheeraj Vattikonda, Nicolas Gontier, Alexandre Lacoste, Laurent Charlin, and Massimo Caccia. Privileged information distillation for language models, 2026. URL https://arxiv.org/abs/2602.04942.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL https://arxiv.org/abs/2305.18290.

Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets, 2015. URL https://arxiv.org/abs/1412.6550.

Khouloud Saadi and Di Wang. What should feature distillation transfer in llms? a task-tangent geometry view, 2026. URL https://arxiv.org/abs/2507.10155.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning, 2026. URL https://arxiv.org/abs/2601.19897.

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, 2020. URL https://arxiv.org/abs/2002.10957.

Xuezhi Wang and Denny Zhou. Chain-of-thought reasoning without prompting, 2024. URL https://arxiv.org/abs/2402.10200.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. Finetuned language models are zero-shot learners, 2022. URL https://arxiv.org/abs/2109.01652.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. A survey on knowledge distillation of large language models, 2024. URL https://arxiv.org/abs/2402.13116.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. Star: Bootstrapping reasoning with reasoning, 2022. URL https://arxiv.org/abs/2203.14465.

Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024, 2024.

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.

A THINKING MODE: AN OPEN PROBLEM

Qwen3 models can be evaluated in a thinking mode that emits an explicit reasoning trace before answering. Everything in the main paper is measured with thinking disabled, matching the mode the student rolls out in during training. Evaluated with thinking enabled, our activation-space objectives do not significantly improve on the untrained model. This limitation is concurrently reported by Kaur et al. (2026), where they indicates it is a property of privileged on-policy distillation rather than of the activation-space objective specifically. We do not attempt to solve it.

A.1 THE OBSERVATION

Table 4 evaluates Qwen3-1.7B with thinking enabled, under the same protocol as the main results (temperature 1.0, `avg@12`, 38,912 generated tokens). Differences are paired bootstraps over per-problem accuracy against the untrained base model, which removes problem-difficulty variance.

Table 4: **Thinking-mode accuracy is indistinguishable from the untrained model** (Qwen3-1.7B, `avg@12`, temperature 1.0). Δ is a paired bootstrap over problems against the base row, combining the 60 AIME24 and HMMT25 problems.

Model (thinking mode)	AIME24	HMMT25	Δ vs. base (combined)
Base, untrained	48.9	21.7	—
A-OPSD Fisher, step 100	48.6	23.6	$+0.8$ $[-3.3, +4.9]$, p=0.69
A-OPSD MSE, step 100	47.2	21.4	-1.0 $[-4.2, +2.1]$, p=0.53

Neither objective moves thinking-mode accuracy: the combined intervals contain zero comfortably, and training four hundred steps longer does not change it. This is not a small-model artifact — at 4B and 8B the best thinking-mode checkpoint we have is likewise within noise of the base model.

A.2 IT DEGRADES WHEN THE STUDENT THINKS DURING TRAINING

The natural fix — let the student roll out with thinking enabled, so that training and evaluation share a mode — makes matters worse rather than better.

Table 5: **Training the student in thinking mode degrades thinking-mode accuracy** (Qwen3-1.7B, AIME24 `avg@12`, paired bootstrap against the base model over 30 problems).

Student rollout mode	AIME24	Δ vs. base
Non-thinking (our recipe)	48.6	-0.3 $[-7.5, +6.7]$, p=0.95
Thinking, 8k rollout budget	46.1	-2.8 $[-6.4, +0.8]$, p=0.14
Thinking, 1k rollout budget	41.4	$\mathbf{-7.5}$ $[\mathbf{-12.5}, \mathbf{-2.5}]$, p=0.003

The 1k-budget run is a significant loss, and the 8k run trends the same way. The trajectory of the latter (step 25: 46.4, 50: 43.6, 75: 42.8, 100: 46.1) never returns to the base model's 48.9. Matching the training and evaluation modes therefore does not recover the gap; it converts a null result into a regression.

A.3 WHY THIS HAPPENS

Kaur et al. (2026) report the same failure at a larger scale and localize its mechanism. Training five thinking models with privileged solution context lowers `avg@16` for every one of them, with Qwen3-1.7B — our backbone — degrading most ($0.372 \rightarrow 0.308$). Critically for us, they also run the configuration we use, disabling thinking during training while evaluating with it enabled, and find it preserves the base model's average without improving it ($0.591 \rightarrow 0.590$). Table 4 is that result reproduced independently.

Their proposed mechanism is the suppression of *forking* positions: high-entropy points in a rollout where several continuations remain plausible and lead down different reasoning paths. Conditioning

the teacher on the reference solution collapses the teacher's uncertainty at exactly these points, so the distillation signal stops rewarding the branch-taking behavior that test-time search depends on, and can invert the credit assigned to self-correction cues.

B Does the Activation Objective Reduce the Token-Space KL?

Section 3.2 derives the Fisher objective as the second-order pullback of OPSD's forward KL through the frozen LM head. That derivation is local: it says the two objectives agree to second order at the teacher's state, not that descending one descends the other over a hundred optimizer steps. This appendix measures the token-space KL directly at every checkpoint of both 1.7B runs and finds that it does — and, unexpectedly, that A-OPSD ends training with a *lower* token-space KL than OPSD, which descends that KL by construction.

Protocol. For every 25-step checkpoint of both matched 100-step runs, plus the untrained base as step 0, we score held-out OpenThoughts problems (indices 20,000+, past the ≈3.2k examples a 100-step run consumes) under q = the base model with the privileged prompt (problem + reference solution, thinking on) and p = the checkpoint with the student prompt (problem only, thinking off). The teacher is the frozen base model, exactly as in training, so q is identical at every step. One trajectory set is sampled *once* from the untrained base policy under the student prompt and shared by all nine checkpoints, so every checkpoints scores the same 23,838 decision tokens over the same 48 problems; the teacher never generates.

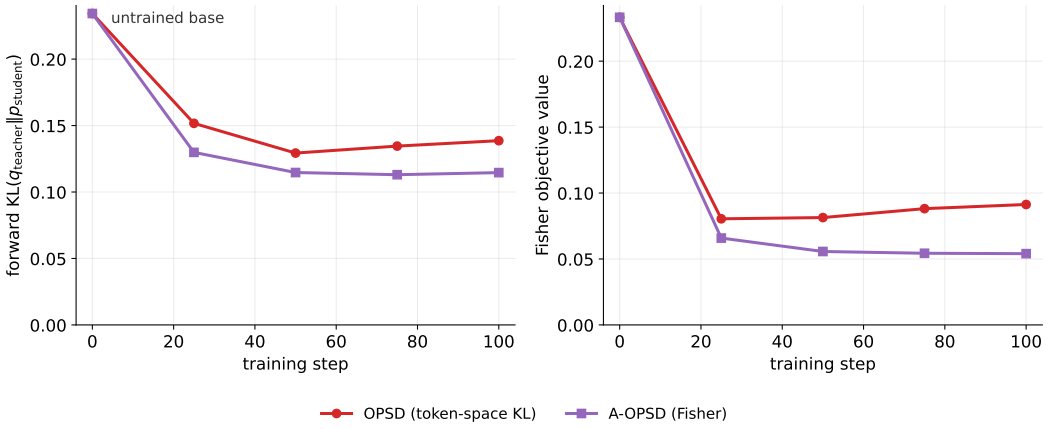


Figure 5: **Both objectives across training** (Qwen3-1.7B, fixed base-policy rollouts). **Left:** the token-space forward KL that OPSD minimizes directly. A-OPSD reaches a lower value despite supervising only hidden states. **Right:** the Fisher objective over the same checkpoints, showing the same pattern.

The activation objective can be a better KL minimizer. From the untrained base, A-OPSD reduces the token-space KL by 51.1% against OPSD's 40.8%, and the Fisher objective by 76.8% against 60.9% (Figure 5). A-OPSD therefore wins on both metrics, including the one OPSD optimizes by construction.

C EXPERIMENTAL DETAILS

C.1 TRAINING SETTINGS & EVALUATION PROTOCOL.

Table 6 lists the training configuration and Table 7 the evaluation protocol used in the paper by default.

Table 6: **Training configuration.**

Setting	Value
Backbones	Qwen3-1.7B, Qwen3-4B, Qwen3-8B, OLMO-3-7B-INSTRUCT
Training data	OpenThoughts math subset, 30k problems with reference solutions
Adapter	LoRA, r=64, α=128, dropout 0.05, on `q,k,v,o,gate,up,down`
Optimizer	AdamW, β=$(0.9, 0.999)$, no weight decay, gradient clip 0.1
Batch / steps / seeds	effective batch 32, 100 steps, 3 seeds
Top-k (A-OPSD)	$k = 64$ unless specified otherwise
Teacher	the student's base model with adapters disabled and the reference solution in context
Reasoning mode	student non-thinking; teacher thinking (Qwen3), non-thinking (OLMO)
Rollouts	on-policy via vLLM; temperature 1.1, top-p 0.95, top-k 20, $\leq$ 1024 tokens
Precision / hardware	bf16, FlashAttention-2, 4×A100-80GB

Table 7: **Evaluation protocol.**

Setting	Value
Benchmarks	AIME24 (30 problems), HMMT25 (30), MATH-500 (500)
Samples per problem	12 (AIME24, HMMT25), 4 (MATH-500)
Metric	`avg@`n, accuracy averaged over the n samples; answer read from the final `\boxed{}`
Reasoning mode	non-thinking
Sampling	temperature 1.0, top-p 0.8
Generation budget	38,912 new tokens (context 40,960)
Reporting	mean $\pm$ standard deviation over 3 training seeds (3 evaluation repeats for base models)
Engine	vLLM, tensor-parallel 4

C.2 PER-TOKEN FISHER LOSS DISTRIBUTION AND THE CHOICE OF THE CAP VALUE

The cap value c (Section 3.3) is set from the empirical distribution of the per-token Fisher loss $\ell^{(n)}_{\text{Fisher}}$ near convergence, measured over the masked decision tokens of training batches (Qwen3-1.7B, standard configuration). The distribution is extremely heavy-tailed: the median token is essentially already matched, while the mean is dominated by a small high-loss tail — the irreducible, solution-dependent residue that drives the over-training collapse.

Table 8: Per-token Fisher loss percentiles near convergence (ranges over training batches). The median is ≈ 0 while the maximum is hundreds of times the mean: the loss is carried by a small subset of stubborn tokens. We set the cap at $c \approx$ p90 (c=0.1).

	p50	**p75**	**p90**	**p95**	**p99**	**max**	**mean**
$\ell^{(n)}_{\text{Fisher}}$	0.000–0.003	0.01–0.04	0.08–0.14	0.18–0.31	0.7–1.7	9–52	0.045–0.09