TAILORING TEACHING TO APTITUDE: DIRECTION-ADAPTIVE SELF-DISTILLATION FOR LLM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy self-distillation (OPSD) is an emerging LLM post-training paradigm in which the model serves as its own teacher: conditioned on privileged information such as a reference trace or hint, the same policy provides dense token-level supervision on its own rollouts. However, recent studies show that OPSD degrades complex reasoning by suppressing predictive uncertainty, which supports exploration and hypothesis revision. Our token-level analysis shows that this failure arises from applying a uniform direction of teacher supervision across tokens with different uncertainty levels: conformity to the privileged self-teacher suppresses exploration at high entropy, while deviation from the teacher degrades step accuracy at low entropy. Accordingly, we propose **Direction-Adaptive Self-Distillation** (**DASD**), which reframes privileged self-distillation from uniform teacher imitation into entropy-routed directional supervision: high-entropy tokens are pushed away from the privileged teacher to preserve exploration, while low-entropy tokens are pulled toward the teacher to stabilize step-level execution. Across six mathematical reasoning benchmarks, DASD achieves the best macro Avg@16 over strong RLVR and self-distillation baselines. Pass@k, execution-and-exploration, and generalization analyses show that these average gains come from preserving exploration without sacrificing step-level execution. Our anonymous code is available here.

1 INTRODUCTION

Knowledge distillation (Hinton et al., 2015; Kim & Rush, 2016; Furlanello et al., 2018; Xu et al., 2024) and its self-distilling variants (Snell et al., 2022; Song & Zheng, 2026) have recently emerged as efficient ways to improve reasoning in large language models (LLMs) during post-training with reinforcement learning from verifiable rewards (RLVR) (Schulman et al., 2017; Christiano et al., 2017; Ouyang et al., 2022; Shao et al., 2024; Guo et al., 2025; Yu et al., 2025; Zhang et al., 2025; Liu et al., 2025; Chen et al., 2026a). In on-policy self-distillation (OPSD), the model generates its own rollouts as a student, while a privileged conditioning of the same model—given information such as a reference trace, solution hint, or feedback—serves as a teacher that provides dense token-level guidance (Zhao et al., 2026; Li et al., 2026a; Hübotter et al., 2026; Yang et al., 2026; Shenfeld et al., 2026). OPSD is attractive because it couples RLVR's outcome supervision with fine-grained teacher feedback, without requiring a separate external teacher (Shao et al., 2024; Zhao et al., 2026; Hübotter et al., 2026).

Despite this appeal, prior work shows that OPSD can degrade reasoning on complex problems by suppressing uncertainty expressions during reasoning (Kim et al., 2026). Privileged conditioning on reference information r makes the self-teacher concentrate on a confident, solution-conditioned path; imitation then pushes the student toward the same low-uncertainty style. This suppresses markers such as "wait", which help an unprivileged solver explore alternatives, revise hypotheses, and recover from mistakes, thereby weakening the exploratory search and self-correction needed for difficult reasoning.

This diagnosis motivates a natural question: if conformity to the privileged teacher harms reasoning by suppressing uncertainty, can reversing the distillation direction restore it? Our sign-flip probe shows that it cannot. Deviation recovers epistemic markers, but corrupts step-level execution. This

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.

complementary failure indicates that teacher pressure is not uniformly beneficial or harmful; its effect depends on the token regime. Prefix interventions further isolate this mechanism: conformity damages high-entropy forking tokens, where exploration must be preserved, whereas deviation damages low-entropy scaffolding tokens, where stable execution must be maintained.

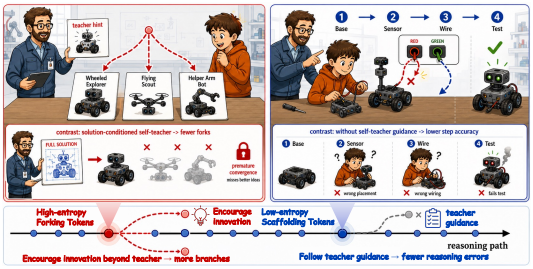


Figure 1: **Conceptual analogy for DASD.** Student entropy switches the role of the solution-conditioned self-teacher. High-entropy forking tokens should move away from the teacher to avoid premature convergence and preserve alternative reasoning paths, whereas low-entropy scaffolding tokens should follow the teacher to prevent routine execution errors.

Building on this dissection, we turn the analysis into a concrete training prescription: **Direction-Adaptive Self-Distillation** (**DASD**). As illustrated in Figure 1, high-entropy forking tokens correspond to decision points where directly imitating the solution-conditioned teacher may prematurely collapse alternative reasoning paths. DASD therefore pushes these tokens away from the teacher to preserve exploration. By contrast, low-entropy scaffolding tokens correspond to routine execution steps, where teacher guidance can help prevent local errors. DASD therefore pulls these tokens toward the teacher to stabilize reasoning. In this way, DASD reframes privileged self-distillation from uniform teacher imitation into entropy-routed directional supervision: the verifier remains the global outcome anchor, while the self-teacher provides a local token-level signal whose direction is adapted according to entropy.

Across six mathematical reasoning benchmarks, it attains the best macro Avg@16 over strong RLVR and self-distillation baselines at every scale, with the largest average gains on harder competition-style benchmarks (OlympiadBench and AIME24) that require maintaining multiple plausible solution paths. Pass@k scaling and execution-and-exploration diagnostics show why: DASD expands exploration while preserving step-level execution, matching the token-level mechanism above.

Our contributions are threefold: (i) we identify a token-level directional mismatch behind OPSD degradation, where teacher conformity suppresses high-entropy forking tokens and teacher deviation corrupts low-entropy scaffolding tokens; (ii) we propose Direction-Adaptive Self-Distillation (DASD), an entropy-routed objective that pushes high-entropy tokens away from the privileged self-teacher and pulls low-entropy tokens toward it; and (iii) we show across six mathematical reasoning benchmarks that DASD improves performance by preserving exploration while maintaining step-level execution.

2 Preliminaries

Group Relative Policy Optimization. We consider the standard RLVR setting in which a policy π_θ generates a response $o = (o_1, \ldots, o_T)$ for a question x, and a programmatic verifier returns a binary reward $R(x, o) \in \{0, 1\}$ (Shao et al., 2024; Guo et al., 2025). Group Relative Policy Optimization (GRPO) (Shao et al., 2024) samples G trajectories from $\pi_{\theta_{\text{old}}}(\cdot \mid x)$, assigns $A_G^{(i)} = (R(x, o^{(i)}) - \mu_G)/\sigma_G$ using the group's reward mean μ_G and standard deviation σ_G, and optimizes the clipped surrogate below, where $\kappa_t^{(i)} = \pi_\theta(o_t^{(i)} \mid x, o_{<t}^{(i)})/\pi_{\theta_{\text{old}}}(o_t^{(i)} \mid x, o_{<t}^{(i)})$:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o^{(i)}|}\sum_{t=1}^{|o^{(i)}|}\min\left(\kappa_t^{(i)}A_G^{(i)}, \text{clip}(\kappa_t^{(i)}, 1-\epsilon_{\text{clip}}, 1+\epsilon_{\text{clip}})A_G^{(i)}\right)\right]. \quad (1)$$

GRPO's limitation is *sparse credit assignment*: every token in a trajectory inherits the same scalar $A_G^{(i)}$, so decisive deductions, routine algebra, and stylistic filler are pushed in the same direction.

On-policy distillation and self-distillation. On-policy distillation (OPD) (Agarwal et al., 2024) densifies supervision by matching the student's next-token distribution to a teacher distribution on the student's own rollouts. OPD normally requires a stronger external teacher; on-policy *self*-distillation (OPSD) (Zhao et al., 2026; Hübotter et al., 2026) instead uses the same policy as a teacher, specialized by *privileged information* r such as a verified trace, solution hint, or environ-

ment feedback:

$$\mathcal{L}_{\mathrm{OPSD}}(\theta) = \mathbb{E}_{(x,r)}\,\mathbb{E}_{o\sim\pi_\theta(\cdot|x)}\left[\frac{1}{|o|}\sum_{t=1}^{|o|} D(\mathrm{sg}[\pi_\theta(\cdot \mid x, r, o_{<t})] \,\|\, \pi_\theta(\cdot \mid x, o_{<t}))\right], \tag{2}$$

where D is a divergence (forward KL with pointwise clipping in Zhao et al. (2026), reverse KL in SDPO); gradients flow only through the student. As in Zhao et al. (2026), our OPSD baseline uses the initial policy as teacher (Appendix D.2).

Why OPSD can fail on reasoning. Despite this efficiency, Kim et al. (2026) show that OPSD can degrade mathematical reasoning. Privileged conditioning resolves the teacher's residual uncertainty, so its outputs contain fewer epistemic markers such as "wait", "hmm", and "alternatively"—tokens that help an unprivileged solver branch, revise, and recover. The student can therefore distill a *solution-conditioned style*: it learns to generate with confidence that depends on r, even though r is unavailable at inference time. Section 3 asks where in a rollout this mismatch does its damage.

3 A Token-Level Anatomy of OPSD's Reasoning Failure

Three probes, run before introducing DASD, each test the hypothesis the previous one leaves open.

Probe 1: Does reversing self-distillation direction restore reasoning? If imitating a privileged teacher harms reasoning by erasing uncertainty, a tempting repair is to reverse the direction and push the student away from that teacher. To isolate teacher-pressure direction from full-distribution privileged-information leakage (Yang et al., 2026), we use the sampled log-evidence gap: $A_t = \log \pi_\theta(o_t \mid x, r, o_{<t}) - \log \pi_\theta(o_t \mid x, o_{<t})$, and optimize $\max_\theta \mathbb{E}_{o\sim\pi_\theta}\big[\sum_t s_t A_t\big]$ with a global sign $s_t \in \{+1, -1\}$. **Conformity** ($s_t \equiv +1$) is the OPSD-like direction: it pulls the student toward the privileged teacher. **Novelty** ($s_t \equiv -1$) is the mirror direction: it pushes the student away. All probes train Qwen3-1.7B on DAPO-Math-17k with a GRPO backbone and are evaluated on AIME24 rollouts. We track three quantities: global reasoning (pass@16 and acc@16, the mean accuracy over 16 samples); execution, measured by StepAcc, the fraction of reasoning steps judged correct by a ProcessBench-style step critic (Zheng et al., 2025); and exploration, measured by $E(y)$, the number of epistemic markers such as "wait", "hmm", and "alternatively" in a response y, with response length as a secondary signal (Appendix D).

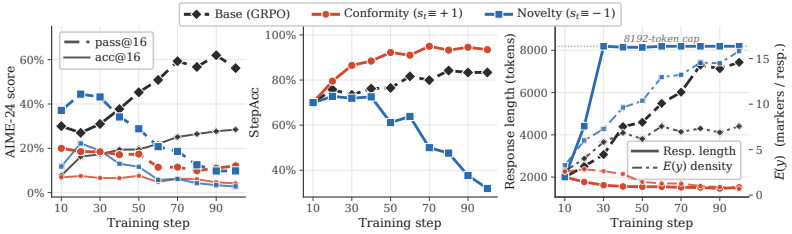


Figure 2: Training trajectories of *Base* (GRPO), *Conformity* ($s_t \equiv +1$), and *Novelty* ($s_t \equiv -1$) on AIME24. (a) Global reasoning (pass@16, acc@16). (b) Rigorous execution (StepAcc). (c) Exploration ($E(y)$, response length). Conformity suppresses exploration without gaining accuracy; Novelty briefly overshoots Base then collapses into verbosity without reasoning.

Finding 1: both uniform signs fail, but they fail on different axes. Figure 2(a) shows that neither sign matches *Base*: Conformity quickly loses pass@16 and never improves acc@16, while Novelty briefly rises above Base before collapsing. The component metrics explain why. Conformity keeps StepAcc at or above Base but cuts response length and sharply suppresses $E(y)$, so what it loses is exploration, not execution. Novelty does the opposite: $E(y)$ and length keep growing after accuracy crashes, while StepAcc deteriorates. The mirror hypothesis therefore fails. Novelty also shows that longer responses and more markers do not by themselves amount to useful exploration.

Probe 2: Which token property separates the two failure modes? The complementary failures in Probe 1 imply that Conformity and Novelty act most strongly on different token populations. Because privileged conditioning mainly removes residual uncertainty (Kim et al., 2026), the student's predictive entropy $H_t = \mathcal{H}[\pi_\theta(\cdot \mid x, o_{<t})]$ is the natural candidate separator. We measure this relationship in two ways: A_t captures the intended signed teacher pressure, while a one-step counterfactual TV shift captures the realized distributional displacement, $D_t^{(s)} = \mathrm{TV}(\pi_\theta(\cdot \mid$

$x, o_{<t})$, $\pi_\theta^{(s\text{-step})}(\cdot \mid x, o_{<t})\big)$, where $\pi_\theta^{(s\text{-step})}$ is the student after one gradient step with sign $s \in \{+1, -1\}$.

Finding 2: the two signs concentrate pressure at opposite entropy regimes. Figure 3(a) shows that strong Conformity pressure concentrates at high-H_t tokens, while strong Novelty pressure concentrates at low-H_t tokens; the signed gap has a positive Spearman correlation with entropy ($\rho_s = +0.52$). Figure 3(b) shows the same split in realized movement: Conformity shifts the student most at high-H_t forks, whereas Novelty shifts it most at low-H_t scaffolding. This aligns with prior entropy analyses of reasoning: high-entropy forking tokens carry much of RLVR's search benefit, while low-entropy tokens execute deterministic scaffolding (Wang et al., 2025; Jin et al., 2026). Entropy therefore connects the two failures mechanistically: uniform attraction attacks the forks that should stay exploratory, while uniform repulsion attacks the scaffolding that should stay stable. Computed from the student alone, H_t needs no privileged information and can serve as the router.

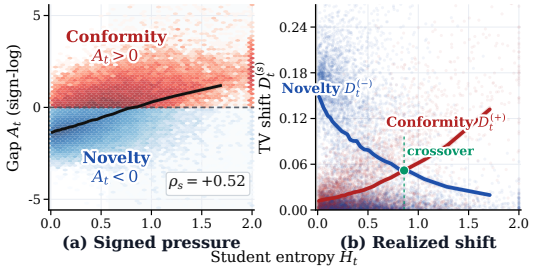


Figure 3: Token-level OPSD pressure vs. student entropy H_t. **(a)** Log-evidence gap A_t ($A_t{>}0$: Conformity; $A_t{<}0$: Novelty), Spearman $\rho_s{=}{+}0.52$. **(b)** TV shift $D_t^{(s)}$: Conformity displaces hardest at high-H_t forks; Novelty at low-H_t scaffolding.

Probe 3: Does entropy routing causally repair the mismatch? The entropy split suggests a diagonal repair: use Conformity on low-H_t scaffolding and Novelty on high-H_t forks. To test this directly, we perform a prefix-intervention probe at inference time, with no gradient update. At each position t, we bucket the token by student entropy ($H_t < \tau_{0.5}$ or $H_t \geq \tau_{0.5}$), where $\tau_{0.5}$ is the trajectory-local median entropy. Within the chosen bucket, we sample o_t from one of two interventional distributions: Conformity substitutes the privileged teacher, $\tilde{\pi}_t^{C} = \pi_\theta(\cdot \mid x, r, o_{<t})$, while Novelty softly suppresses teacher-aligned mass, $\tilde{\pi}_t^{N}(v) \propto \pi_\theta(v \mid x, o_{<t}) \,/\, \pi_\theta(v \mid x, r, o_{<t})^\alpha, \alpha = 0.5$. After the intervened token, generation resumes under the unprivileged student. A Random-H_t control applies the same interventions to entropy-agnostic token samples, isolating entropy from token frequency or position. We report changes in StepAcc and $E(y)$ relative to unmodified AIME24 rollouts. Because the probe acts on the sampling distribution of a fixed model, it isolates the direction of teacher pressure from any choice of training divergence.

Finding 3: correct routing helps, while misrouting reproduces the OPSD failure. Table 1 shows the desired diagonal pattern. On low-H_t scaffolding, Conformity improves step accuracy by 18.3% with only a small expected side-cost in exploration. On high-H_t forks, Novelty increases $E(y)$ by 61.3% with only a modest step-accuracy cost. The off-diagonal cells show the complementary harms predicted by Probe 2: Conformity on high-H_t tokens collapses $E(y)$ by 86.2%, while Novelty on low-H_t tokens drops StepAcc by 79.4%. The Random-H_t control keeps all effects near zero ($|\Delta| \leq 5\%$), ruling out token frequency and position as main explanations.

Table 1: **Prefix interventions** (% change vs. unmodified rollouts). Green/red highlight the predicted target gains/failures; gray rows are entropy-agnostic controls.

Bucket / metric	Conformity	Novelty
Low-H_t scaffolding		
ΔStepAcc	**+18.3%**	−79.4%
$\Delta E(y)$	−4.21%	+5.03%
High-H_t forks		
ΔStepAcc	−2.17%	−6.18%
$\Delta E(y)$	−86.2%	**+61.3%**
Random-H_t control		
ΔStepAcc	+1.84%	−2.09%
$\Delta E(y)$	−3.56%	+4.47%

The probes yield the design rule used next. Uniform teacher pressure harms reasoning because its direction is wrong for one of the two entropy regimes, so the repair is to route the same teacher signal diagonally: attractive on low-H_t scaffolding, repulsive on high-H_t forks. The router uses only the student's own entropy, so the privileged context is needed during training but not at inference. Appendix A gives the full probe protocols.

4 Direction-Adaptive Self-Distillation

DASD turns the probe results into a training objective. It keeps the privileged self-teacher as a source of dense token-level evidence but lets student entropy decide whether that evidence attracts or repels.

Signed self-distillation term. The probes require more than reweighting: the same teacher signal must attract low-entropy scaffolding and repel high-entropy forks. At the prefix $(x, o^{(i)}_{<t})$ of rollout i, the student distribution is $\pi_\theta(\cdot \mid x, o^{(i)}_{<t})$. The privileged self-teacher is the same policy evaluated on the same student-generated prefix with r added to the prompt, $\mathrm{sg}[\pi_\theta(\cdot \mid x, r, o^{(i)}_{<t})]$; the stop-gradient $\mathrm{sg}[\cdot]$ makes it a fixed target during each update. OPSD minimizes a divergence to a privileged teacher and is uniformly attractive. DASD instead multiplies their reverse KL by a signed coefficient:

$$\widetilde{\mathcal{D}}^{(i)}_t(\theta) = \omega^{(i)}_t \mathrm{KL}\Big(\pi_\theta(\cdot \mid x, o^{(i)}_{<t}) \,\Big\|\, \mathrm{sg}\Big[\pi_\theta(\cdot \mid x, r, o^{(i)}_{<t})\Big]\Big), \qquad \omega^{(i)}_t \in [-1, 1]. \tag{3}$$

With $\omega^{(i)}_t > 0$, minimizing the term pulls the student toward the teacher, as in OPSD. With $\omega^{(i)}_t < 0$, it pushes the student away, as needed where imitation would collapse alternative continuations.

Entropy-routed direction and gap-gated magnitude. DASD computes ω_t from quantities measured on each rollout. Student entropy sets the sign, and the teacher–student gap on the sampled token sets how much the token contributes. Define $\delta^{(i)}_t = \log \pi_\theta(o^{(i)}_t \mid x, r, o^{(i)}_{<t}) - \log \pi_\theta(o^{(i)}_t \mid x, o^{(i)}_{<t})$ as the privileged self-teacher's log-probability advantage on the sampled token. Let $\tau^{(i)}_\rho$ be the ρ-quantile of the trajectory entropies, let $\hat{\sigma}^{(i)}_H = \mathrm{MeanAbsDev}_t(\{H^{(i)}_t\})$ be their mean absolute deviation, and let $\tilde{\delta}^{(i)} = \mathrm{median}_t\, |\delta^{(i)}_t|$. With $\bar{\delta}^{(i)}_t = \delta^{(i)}_t / (\tilde{\delta}^{(i)} + \epsilon)$ and logistic sigmoid $\sigma(\cdot)$, DASD sets:

$$\omega^{(i)}_t = \underbrace{\tanh\left(\frac{\tau^{(i)}_\rho - H^{(i)}_t}{\hat{\sigma}^{(i)}_H + \epsilon}\right)}_{\text{entropy router}} \cdot \underbrace{\sigma\Big(|\bar{\delta}^{(i)}_t| - 1\Big)}_{\text{gap gate}}, \quad \tau^{(i)}_\rho = \mathrm{Quantile}_\rho\Big(\{H^{(i)}_t\}_t\Big). \tag{4}$$

The router is positive for tokens below the trajectory's ρ-quantile of entropy and negative above it, which implements the diagonal rule of Table 1. With $\rho = 0.20$, most tokens lie above $\tau^{(i)}_\rho$, but $\tanh$ keeps their repulsion weak unless their entropy exceeds it by a few mean absolute deviations; the sigmoid gate further down-weights gaps below the trajectory's median, so strong repulsion requires both high entropy and a large gap.

Verifier-anchored objective. The signed term says which way each token should move, but not whether the trajectory is correct. DASD keeps the verifier for that purpose. With $A^{(i)}_G = (R(x, o^{(i)}) - \mu_G)/\sigma_G$ the group-relative verifier advantage over the G rollouts of prompt x, the training loss is

$$\ell_{\mathrm{DASD}}(\theta) = -\mathbb{E}\Big[\tfrac{1}{G}\textstyle\sum_{i=1}^{G} \frac{1}{|o^{(i)}|} \sum_{t=1}^{|o^{(i)}|} \min\Big(\kappa^{(i)}_t A^{(i)}_G, \mathrm{clip}(\kappa^{(i)}_t, 1-\epsilon_{\mathrm{clip}}, 1+\epsilon_{\mathrm{clip}}) A^{(i)}_G\Big)\Big] + \beta\, \mathbb{E}\Big[\tfrac{1}{G}\textstyle\sum_{i=1}^{G} \frac{1}{|o^{(i)}|} \sum_{t=1}^{|o^{(i)}|} \widetilde{\mathcal{D}}^{(i)}_t(\theta)\Big]. \tag{5}$$

The first term is the GRPO objective; the second is the signed self-distillation term, weighted by β ($\beta = 1$ in all experiments; Appendix I.2). The two terms have separate roles: the verifier term rewards correct trajectories, and the signed term redistributes credit among the tokens within them. Section 6 (RQ3) evaluates each in isolation.

Token-level estimator. The KL term in Eq. (5) is defined over full next-token distributions, whereas PPO updates the log-probabilities of sampled tokens. Proposition 1 gives the per-token form of Eq. (5): for a sampled token, its gradient is a policy gradient whose advantage is A_G plus a signed teacher correction.

Proposition 1 (Token-level form of the DASD objective)**.** *Fix $x, o_{<t}, r$ and a sampled token o_t. Let $p_t = \pi_\theta(\cdot \mid x, o_{<t})$, let $q_t = \mathrm{sg}[\pi_\theta(\cdot \mid x, r, o_{<t})]$, and define $\delta_t = \log q_t(o_t) - \log p_t(o_t)$. Treating q_t, A_G, and the measured routing coefficient ω_t as fixed during the current update,*

$$\ell_t(\theta) = -A_G \log p_t(o_t) + \beta\, \omega_t\, \mathrm{KL}(p_t \| q_t), \quad \nabla_\theta \ell_t \overset{\mathrm{MC}}{\doteq} -\big(A_G + \beta\, \omega_t \delta_t\big) \nabla_\theta \log p_t(o_t).$$

Here $\doteq$ denotes Monte Carlo score-function form after dropping action-independent baseline terms.

Appendix B gives the proof. The signed KL term of Eq. (5) thus becomes an additive per-token correction $\beta\omega_t\delta_t$ to the verifier advantage. For numerical stability, DASD normalizes the gap within each trajectory and uses the advantage $\hat{A}^{(i)}_t = A^{(i)}_G + \beta\omega^{(i)}_t \bar{\delta}^{(i)}_t$ in the standard clipped PPO surrogate (Algorithm 1, Appendix C). In the limiting cases, $\beta = 0$ recovers GRPO, $\omega_t \equiv +1$ recovers uniform attraction (the OPSD direction), and $\omega_t \equiv -1$ recovers uniform repulsion (the Novelty probe). DASD adds one forward pass of the policy on (x, r, o) per rollout, the same privileged pass that OPSD, SDPO, and RLSD require; r is one of the G rollouts already sampled, so no extra generation is needed.

Table 2: Main results under the shared protocol (Avg@16, %; non-thinking mode, 8,192-token budget, final checkpoints; Section 5). **Bold**/underline mark best/second-best within each scale. Pass@16 is in Table 7.

Method	MATH500	Minerva	HMMT25	Olympiad	AIME24	AIME25	Avg
Qwen3-1.7B	58.4	17.5	7.9	33.4	13.1	10.6	23.5
+GRPO (Shao et al., 2024)	68.2	22.6	11.5	40.8	32.5	27.5	33.8
+HEPO (Wang et al., 2025)	68.4	22.8	15.6	42.8	32.5	29.2	35.2
+OPSD (Zhao et al., 2026)	66.7	21.6	10.0	39.5	32.1	25.6	32.6
+SDPO (Hübotter et al., 2026)	67.4	21.2	10.6	39.6	31.7	26.0	32.8
+RLSD (Yang et al., 2026)	68.7	25.1	15.2	39.6	**35.4**	26.7	35.1
+SRPO (Li et al., 2026a)	62.8	20.1	10.2	36.6	23.3	17.1	28.4
+DASD (ours)	**70.4**	**25.3**	**17.7**	**45.4**	35.0	**32.1**	**37.7**
Qwen3-4B	67.0	24.6	15.2	43.1	23.1	18.5	31.9
+GRPO (Shao et al., 2024)	72.6	28.2	22.1	47.0	55.0	49.8	45.8
+HEPO (Wang et al., 2025)	72.7	27.1	21.2	50.4	53.3	45.2	45.0
+OPSD (Zhao et al., 2026)	71.2	26.7	21.7	46.0	54.2	48.1	44.7
+SDPO (Hübotter et al., 2026)	71.9	27.0	21.5	45.8	53.5	49.0	44.8
+RLSD (Yang et al., 2026)	73.0	28.5	20.4	44.9	56.5	47.7	45.2
+SRPO (Li et al., 2026a)	69.6	26.1	17.7	45.0	40.0	34.4	38.8
+DASD (ours)	**75.0**	**31.0**	**25.0**	**57.5**	**58.3**	**50.0**	**49.5**
Qwen3-8B	67.2	24.7	13.1	44.3	27.9	21.0	33.0
+GRPO (Shao et al., 2024)	73.0	27.7	20.8	49.4	59.4	48.8	46.5
+HEPO (Wang et al., 2025)	73.0	28.7	20.4	50.3	66.7	49.0	48.0
+OPSD (Zhao et al., 2026)	71.7	26.2	19.8	48.8	58.3	48.3	45.5
+SDPO (Hübotter et al., 2026)	72.4	26.5	19.8	48.1	58.8	47.5	45.5
+RLSD (Yang et al., 2026)	73.0	29.7	20.8	49.4	62.3	49.8	47.5
+SRPO (Li et al., 2026a)	70.0	25.9	17.5	46.8	43.8	34.6	39.7
+DASD (ours)	**76.0**	**33.0**	**25.0**	**55.8**	**71.7**	**53.3**	**52.5**

5 EXPERIMENTS

Models and data. We train Qwen3-1.7B, 4B, and 8B (Yang et al., 2025); diagnostics and ablations use Qwen3-1.7B. All mathematical training uses DAPO-Math-17k (Yu et al., 2025), whose reference answers define the binary verifier reward. For self-distillation methods, the privileged context r is a rollout of the same problem that the verifier marked correct, inserted with the teacher template in Appendix D; the student never sees r during rollouts or evaluation.

Training. We compare DASD with verifier-only methods (GRPO (Shao et al., 2024), HEPO (Wang et al., 2025)) and privileged self-distillation methods (OPSD (Zhao et al., 2026), SDPO (Hübotter et al., 2026), RLSD (Yang et al., 2026), SRPO (Li et al., 2026a)); Appendix D describes each. All methods are implemented in `verl` (Sheng et al., 2025) and share one budget: each step samples $G = 8$ rollouts for each of $B = 128$ prompts, with PPO mini-batch size 32, learning rate 3×10^{-6} with cosine decay, and 200 steps. We evaluate the final checkpoint of every run. Baseline-specific hyperparameters follow the original papers. DASD uses $\rho = 0.20$ and $\beta = 1$ (sensitivity in Appendix I).

Evaluation. We evaluate on MATH500 (Hendrycks et al., 2021b), Minerva (Lewkowycz et al., 2022), HMMT25, OlympiadBench (He et al., 2024), AIME24, and AIME25 in non-thinking mode, with $K = 16$ samples per problem, temperature 1.0, top-$p = 0.9$, and an 8,192-token limit, and report Avg@16 and Pass@16. Thinking-mode decoding with a 38,912-token limit (Yang et al., 2025; Zhao et al., 2026) yields higher absolute accuracies, with the untrained Qwen3-1.7B checkpoint at 50.6 on AIME24 and 34.7 on AIME25, close to the 51.5 and 36.7 reported by Zhao et al. (2026) (Appendix F); we repeat the main comparison in thinking mode below.

Main results. Table 2 shows that DASD attains the best macro Avg@16 at all three scales, improving over the strongest baseline by 2.5 (1.7B), 3.7 (4B), and 4.5 (8B) points. Averaged over scales, the gain over GRPO ranges from 2.5 points on MATH500 to 7.2 on OlympiadBench, with 6.0 on AIME24 (Figure 7). Pass@16 gives the same ordering: DASD has the best macro Pass@16 at every scale (57.9, 68.0, and 69.0), and OPSD and SDPO end 1.0 to 2.3 points below GRPO (Table 7). OPSD and SDPO, whose teacher is uniformly attractive, do not surpass GRPO at any scale; RLSD, which rescales the teacher signal inside a verifier-anchored update, is the strongest self-distillation baseline. A DASD training step takes 120 s, versus 109 s for GRPO and 127 s for OPSD (Appendix D.4).

Self-distillation under prolonged optimization. Uniform self-distillation is reported to degrade as optimization continues (Kim et al., 2026). A separate controlled experiment (Appendix G) follows the LoRA setup and thinking-mode evaluation of Zhao et al. (2026) and varies only the learning rate

Table 4: **Cross-domain evaluation** (Qwen3-8B, non-thinking). For each domain, every method starts from Qwen3-8B, is trained on that domain's training set, and is evaluated on benchmarks of the same domain. Code: EvalPlus pass@1 (HumanEval+, MBPP+) and pass@1 (LiveCodeBench v6); science: accuracy; tool use: BFCL overall score and API-Bank Level-1 (offline subset). Avg is the mean of the benchmark columns within a domain. **Bold**/underline: best/second-best per column.

	Code generation				Scientific reasoning			Tool use		
Training set	*Skywork-OR1 + KodCode*				*SciKnowEval (3,901)*			*ToolAlpaca tool use (4,046)*		
Method	HumanEval+	MBPP+	LCB v6	**Avg**	GPQA-D	MMLU	**Avg**	BFCL	API-Bank	**Avg**
Qwen3-8B (base)	75.00	65.34	25.19	55.18	37.37	66.60	51.99	51.58	77.84	64.71
+GRPO	77.44	68.25	29.77	58.49	38.88	67.78	53.33	54.78	78.10	66.44
+OPSD	75.61	64.55	25.95	55.37	36.76	63.76	50.26	51.47	77.84	64.66
+DASD (ours)	**79.27**	**69.61**	**31.01**	**59.96**	**39.48**	**68.98**	**54.23**	**56.50**	**79.84**	**68.17**

and the number of sequences distilled per step (Table 11). With 32 per step, as in Zhao et al. (2026), OPSD improves on the untrained checkpoint (50.6 to 56.4 on AIME24); with 1,024, the per-step volume of our main runs, it ends below it at every learning rate (36.9 to 48.1). After thinking-mode training, OPSD trails GRPO by 5.2 and 6.7 points on AIME24 and AIME25 (Table 9). The benefit of uniform attraction thus depends on keeping optimization exposure small. DASD uses the same privileged context and budget yet ranks first at every scale (Table 2); with the fixed-teacher sign-flip probes of Section 3, this places the limitation in the direction of teacher pressure, not in the privileged signal.

Thinking mode and generation budget. We repeat the comparison with thinking enabled in both training and evaluation: Qwen3-1.7B is trained for 200 steps with 512 sequences per step and evaluated with a 32K-token limit (Table 9). The ranking is unchanged: DASD reaches 59.0 on AIME24 and 44.4 on AIME25, 5.9 and 5.2 points above GRPO, a margin at least as large as in non-thinking mode. Evaluating the non-thinking models with 16K and 32K-token limits likewise leaves DASD ahead of GRPO and RLSD on both AIME sets (Table 10). Direction-adaptive supervision therefore helps across decoding modes and generation budgets.

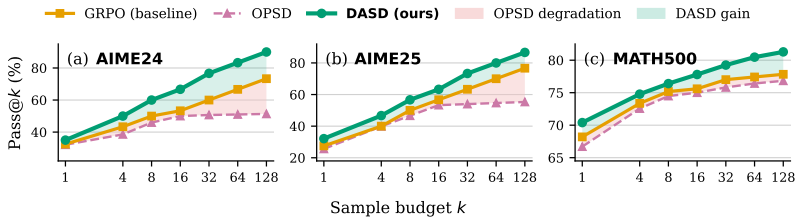


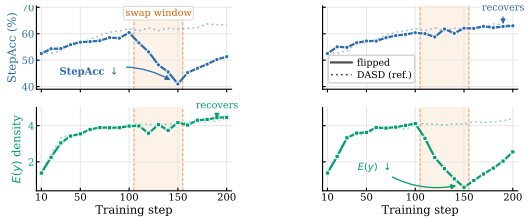
Figure 4: Pass@k of the Qwen3-1.7B checkpoints on AIME24, AIME25, and MATH500 (decoding settings of the main protocol, 128 samples per problem). DASD stays above GRPO and OPSD at every sample budget; on AIME24 and AIME25 its margin over GRPO grows with k.

Pass@k scaling. Avg@16 does not show whether a method only sharpens its first attempt or also reaches more problems. Figure 4 reports Pass@k for k from 1 to 128. On AIME24, DASD starts close to GRPO at Pass@1 and rises to 66.7% at $k = 16$, 83.3% at $k = 64$, and 90.0% at $k = 128$. OPSD stays close to GRPO up to $k = 16$ but gains little beyond it. The widening gap indicates that DASD solves more distinct problems when given more samples, not only the same problems more reliably.

Execution and exploration. Table 3 separates the two components of reasoning quality measured in Section 3, computed on AIME24 rollouts of the final Qwen3-1.7B checkpoints (definitions in Appendix D). OPSD matches or slightly exceeds GRPO on execution but loses most of its exploration: epistemic markers, revisions, and trigram diversity all drop sharply. DASD improves over both on execution (step accuracy 63.7 vs. 58.4 for GRPO) and exceeds GRPO on every exploration metric (4.3 vs. 3.8 markers per response), making it the only method that improves execution and exploration together.

Table 3: **Execution and exploration** on AIME24 rollouts of the final Qwen3-1.7B checkpoints. Higher is better; red marks OPSD's loss of exploration.

Metric	**GRPO**	**OPSD**	**DASD**
Execution quality			
Step accuracy ↑	58.4	59.1	**63.7**
First-error step ↑	4.2	4.4	**5.1**
Correct-step ratio ↑	61.3	62.0	**66.8**
Exploration quality			
Markers $E(y)$ ↑	3.8	0.7	**4.3**
Revision rate ↑	0.41	0.06	**0.47**
Distinct-3 ↑	0.72	0.51	**0.81**
Outcome anchor			
Avg@16 ↑	32.5	32.1	**35.0**



(a) Flip low-H_t routing (b) Flip high-H_t routing

Figure 5: Selective flips of one routing arm during DASD training (Qwen3-1.7B). Inside the shaded window, the sign of one arm is reversed; dotted curves show unperturbed DASD. **(a)** Flipping the low-H_t arm lowers StepAcc while $E(y)$ only fluctuates. **(b)** Flipping the high-H_t arm lowers $E(y)$ while StepAcc is barely affected. Both recover after the original routing is restored.

Cross-domain evaluation. We test whether DASD, developed on mathematics, also improves post-training in code generation, scientific reasoning, and tool use. In each domain, every method starts from Qwen3-8B, trains on that domain's data with a shared optimization setting, and is evaluated in non-thinking mode on separate benchmarks of the same domain (Table 4; setting in Appendix H). DASD attains the best score on every benchmark and the best average in all three domains: 59.96 vs. 58.49 for GRPO in code generation, 54.23 vs. 53.33 in scientific reasoning, and 68.17 vs. 66.44 in tool use. In tool use, DASD is also best in every BFCL category, including non-live, live, and multi-turn calls (Table 13). OPSD, in contrast, falls below the base model in scientific reasoning, on MBPP+, and on BFCL. Direction-adaptive supervision thus avoids the degradation of uniform self-distillation outside mathematics and improves on GRPO on every benchmark, by 0.6 to 1.8 points. On mathematics, DASD also has the best average with Olmo-3-7B-Instruct (Team Olmo, 2025) and Llama-3.1-8B-Instruct (Grattafiori et al., 2024) backbones (56.75 and 14.14, vs. 54.90 and 12.85 for GRPO; Appendix H).

6 Analysis

RQ1: Do the two routing arms play distinct roles? If DASD works as the probes predict, its attractive arm should protect execution and its repulsive arm should protect exploration. We test this during training by reversing the sign of one arm for a window of training steps and then restoring it. Figure 5 shows the predicted separation. Reversing the low-H_t arm, so that the student is pushed away from the teacher on scaffolding tokens, lowers StepAcc from about 60% to about 41%, while $E(y)$ stays near its unperturbed level. Reversing the high-H_t arm, so that forks are pulled toward the teacher, drives $E(y)$ from about 4 to below 1, while StepAcc tracks the unperturbed run. Both metrics recover once the original routing is restored. The two arms therefore control different reasoning competences, as the probes of Section 3 predict.

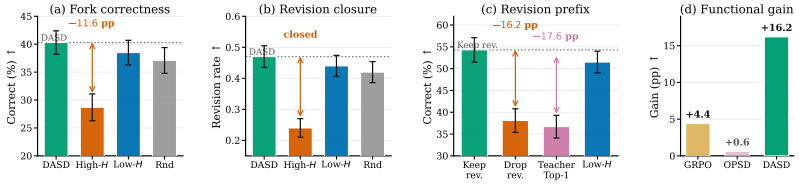


Figure 6: Causal intervention tests on DASD forks and revisions. **(a,b)** Replacing selected tokens with the privileged teacher's top choice measures the effect of high-H_t, low-H_t, and random-position interventions on correctness and revision behavior. **(c)** At matched DASD revision prefixes, preserving, suppressing, or teacher-forcing the revision continuation tests its effect on final correctness. **(d)** The preserve–suppress comparison is shown across DASD, GRPO, and OPSD revisions.

RQ2: Do the protected forks and the revisions they enable matter for correctness? We intervene on trained policies in two ways. First, we test *where* DASD acts: in each rollout, we replace the highest-H_t token with the privileged teacher's top choice, using the lowest-H_t token and a random position as controls. Second, we test *which behavior* at these positions matters. Starting from the same prefix that ends in a revision marker, we compare three continuations: *preserve* lets the model carry out its own revision; *suppress* removes the revision and continues along the pre-revision line of reasoning; and *teacher-force* continues with the privileged teacher. Figure 6(a,b) shows that high-H_t forks carry the effect: forcing them toward the teacher lowers final correctness and removes the revisions that DASD preserves, while low-H_t and random replacements have much smaller effects. Figure 6(c,d) shows that DASD's revisions change outcomes: preserving them rather than suppressing them raises final accuracy by 16.2 points, whereas GRPO and OPSD revisions add much less (4.4 and 0.6 points). The protected forks are thus where functional self-correction happens.

Table 5: **Ablations** (Qwen3-1.7B, Avg@16 under the shared protocol). **Left:** baselines and variants that remove one ingredient of DASD; "teacher-free" rows keep DASD's entropy router but use no privileged forward pass. **Right:** replacing one design choice while holding the other two fixed; $E(y)$ is the number of epistemic markers per response. Shaded rows are DASD; full definitions in Appendix I.

Variant	AIME24	AIME25
Baselines		
GRPO ($\beta = 0$)	32.5	27.5
OPSD (uniform attract.)	32.1	25.6
Teacher-free, same router		
Router + noise	31.9	26.9
Router + entropy reg.	33.5	28.3
Privileged teacher, partial		
Low-H_t attraction only	16.3	11.7
Directional only (no A_G)	34.6	31.5
DASD (full)	**35.0**	**32.1**

Variant	AIME24	AIME25	$E(y)$
A. Routing signal			
Position / token frequency	17.7 / 20.4	15.8 / 17.7	1.6 / 2.1
Gradient norm	25.0	22.7	2.8
Attention entropy	30.8	27.9	3.9
Student entropy H_t	**35.0**	**32.1**	**4.3**
B. Direction map			
Constant -1 (Novelty)	12.1	10.6	**5.6**
Hard threshold / linear ramp	31.5 / 33.5	28.3 / 30.6	4.0 / 4.1
$\tanh$ **direction**	**35.0**	**32.1**	4.3
C. Reliability gate			
No gate / fixed threshold	29.4 / 28.5	26.3 / 25.6	3.8 / 3.7
Magnitude only	30.8	27.7	4.0
Gap-reliability gate	**35.0**	**32.1**	**4.3**

RQ3: Which components are necessary? Table 5 (left) removes one ingredient at a time. Two teacher-free controls keep the entropy router but drop every teacher-dependent quantity. Replacing the teacher correction with zero-mean noise gives no gain over GRPO. An entropy regularizer that sharpens low-entropy tokens and raises entropy at forks gains about one point over GRPO but stays 1.5 and 3.8 points below DASD on AIME24 and AIME25. The teacher gap δ_t therefore carries token-specific information that entropy shaping alone does not provide. Attraction on low-entropy tokens alone is the weakest variant: without repulsion at forks it sharpens already-confident tokens until the rollouts of a prompt agree and the group advantage, which OPSD lacks, carries little signal. Removing the verifier advantage costs less than one point, so most of the gain comes from the direction-adaptive term. DASD also beats GRPO at every tested $\beta \in \{0.5, 1, 2\}$, by 0.6 to 4.6 points (Appendix I.2). The router quantile is similarly robust: $\rho = 0.50$ stays within 3.4 points of the default $\rho = 0.20$ on every metric, whereas $\rho = 0.05$ and $\rho = 0.95$, which give almost all tokens one direction, lose 9.0 and 13.5 points on AIME24 (Table 15). In Table 5 (right), student entropy, the smooth $\tanh$ map, and the gap gate each beat their alternatives.

7 Related Work

Entropy-guided policy optimization. Several verifier-only methods use token entropy to reshape policy-gradient credit: restricting updates to high-entropy forking tokens (Wang et al., 2025), adding entropy bonuses (Cheng et al., 2026), combining reward polarity with entropy (He et al., 2026b), probing signed update directions (Huang et al., 2026), or treating low-entropy tokens specially (Liu et al., 2026; Chen et al., 2026b). In these methods entropy sets *how much* pressure a token receives, while the direction is fixed by the verifier. DASD uses entropy to choose the *sign* of privileged teacher supervision; entropy shaping alone does not reproduce its effect (Table 5).

On-policy self-distillation. OPSD removes the external teacher of on-policy distillation (Agarwal et al., 2024) by conditioning the same model on privileged information (Zhao et al., 2026; Hübotter et al., 2026; Yang et al., 2026; Li et al., 2026a; Penaloza et al., 2026; Stein et al., 2026; Zhang et al., 2026c). Uniform attraction to a solution-conditioned teacher suppresses epistemic markers, leaks privileged style, and hurts out-of-distribution reasoning (Kim et al., 2026; Yang et al., 2026; Fu et al., 2026), and the degradation grows with optimization exposure (Kim et al., 2026). Existing variants adjust the magnitude, filtering, gating, or scheduling of teacher pressure, and entropy-aware OPD uses entropy to balance forward and reverse KL (Jin et al., 2026); in all of them the teacher remains attractive. DASD changes the sign by token, and our diagnosis refines the response-level observation of Kim et al. (2026) by locating where the damage occurs (Appendix K).

8 Conclusion

We traced the reasoning degradation of on-policy self-distillation to the direction of teacher supervision at the token level: uniform attraction to a solution-conditioned teacher suppresses exploration at high-entropy forks, and uniform repulsion corrupts execution at low-entropy scaffolding. Direction-Adaptive Self-Distillation (DASD) lets the student's entropy set the sign of the teacher correction and keeps the verifier as the trajectory-level anchor. Under a shared training budget, DASD improves macro Avg@16 over RLVR and self-distillation baselines at three Qwen3 scales, keeps its advantage in thinking mode, widens Pass@k coverage, and improves step-level execution without losing self-correction. For dense self-distillation signals, the direction of supervision deserves as much design attention as its strength.

Reproducibility Statement

Section 5 and Appendix D specify models, data, prompt templates, the shared training budget, hyperparameters, and the evaluation protocol. Appendices F and G compare evaluation protocols and report OPSD across training regimes. Algorithm 1 gives the full procedure, Appendix B proves Proposition 1, and the anonymous repository contains training and evaluation code with configurations for all methods.

Ethics Statement

This work uses public models, datasets, and benchmarks, and involves no human subjects or private data. Improved post-training of reasoning models could also benefit misuse of such models; we release training code rather than new model weights (Appendix L).

LLM Usage

We used LLMs to assist with editing and polishing the text of the manuscript. They did not contribute to the research ideas, methodology, experiments, analysis, or results.

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. In *International Conference on Learning Representations (ICLR)*, 2024.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *Science China Information Sciences*, 69(6), 2026a. doi: 10.1007/s11432-025-4665-8.

Xinzhu Chen, Xuesheng Li, Zhongxiang Sun, and Weijie Yu. Beyond high-entropy exploration: Correctness-aware low-entropy segment-based advantage shaping for reasoning LLMs. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 32970–32984, 2026b. URL https://aclanthology.org/2026.findings-acl.1650/.

Xiwen Chen, Jingjing Wang, Wenhui Zhu, Peijie Qiu, Xuanzhao Dong, Yueyue Deng, Hejian Sang, Zhipeng Wang, Alborz Geramifard, and Feng Luo. SODA: Semi on-policy black-box distillation for large language models. *arXiv preprint arXiv:2604.03873*, 2026c.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 30377–30385. AAAI Press, 2026. doi: 10.1609/AAAI.V40I36.40290. URL https://doi.org/10.1609/aaai.v40i36.40290.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Kehua Feng, Xinyi Shen, Weijie Wang, Xiang Zhuang, Yuqi Tang, Qiang Zhang, and Keyan Ding. SciKnowEval: Evaluating multi-level scientific knowledge of large language models. *arXiv preprint arXiv:2406.09098*, 2024.

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.

Tommaso Furlanello, Zachary C. Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born-again neural networks. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018.

Zhibin Gou, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. ToRA: A tool-integrated reasoning agent for mathematical problem solving. In *International Conference on Learning Representations (ICLR)*, 2024.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://doi.org/10.1038/s41586-025-09422-z.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025.

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*, 2026a.

Yuhang He, Haodong Wu, Siyi Liu, Hongyu Ge, Hange Zhou, Keyi Wu, Zhuo Zheng, Qihong Lin, Zixin Zhong, and Yongqi Zhang. Where hindsight credit can reside: A signed-capacity view of token updates in RLVR, 2026b. URL https://arxiv.org/abs/2604.11056.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021a.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021b.

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

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In *International Conference on Learning Representations (ICLR)*, 2024.

Kexin Huang, Haoming Meng, Junkang Wu, Jinda Lu, Chiyu Ma, Ziqian Chen, Xue Wang, Bolin Ding, Jiancan Wu, Xiang Wang, Xiangnan He, Guoyin Wang, and Jingren Zhou. On the direction of rlvr updates for llm reasoning: Identification and exploitation, 2026. URL https://arxiv.org/abs/2603.22117.

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. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations (ICLR)*, 2025.

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? In *Conference on Language Modeling (COLM)*, 2026.

Yoon Kim and Alexander M. Rush. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2016.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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*, 2026a.

Minghao Li, Yingxiu Zhao, Bowen Yu, Feifan Song, Hangyu Li, Haiyang Yu, Zhoujun Li, Fei Huang, and Yongbin Li. API-Bank: A comprehensive benchmark for tool-augmented LLMs. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2023.

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

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. In *International Conference on Learning Representations (ICLR)*, 2024.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Keliang Liu, Dingkang Yang, Ziyun Qian, Weijie Yin, Yuchi Wang, Hongsheng Li, Jun Liu, Peng Zhai, Yang Liu, and Lihua Zhang. Reinforcement learning meets large language models: A survey of advancements and applications across the LLM lifecycle. *arXiv preprint arXiv:2509.16679*, 2025.

Shiqi Liu, Zeyu He, Guojian Zhan, Letian Tao, Zhilong Zheng, Jiang Wu, Yinuo Wang, Yang Guan, Kehua Sheng, Bo Zhang, Keqiang Li, Jingliang Duan, and Shengbo Eben Li. Stapo: Stabilizing reinforcement learning for llms by silencing rare spurious tokens, 2026. URL https://arxiv.org/abs/2602.15620.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, Yansong Tang, and Dongmei Zhang. WizardMath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. In *International Conference on Learning Representations (ICLR)*, 2025.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-Refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Long Ouyang, Jeffrey 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. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E. Gonzalez. The berkeley function calling leaderboard (BFCL): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=2GmDdhBdDk.

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.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. In *Conference on Language Modeling (COLM)*, 2024.

Hejian Sang, Yuanda Xu, Zhengze Zhou, Ran He, Zhipeng Wang, and Jiachen Sun. Crisp: Compressed reasoning via iterative self-policy distillation. *arXiv preprint arXiv:2603.05433*, 2026.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. In *International Conference on Learning Representations (ICLR)*, 2016.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. In *ICLR 2026 Workshop on Lifelong Agents: Learning, Aligning, Evolving*, 2026. URL https://openreview.net/forum?id=HlWA3V6iKF.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, pp. 12791297, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400711961. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura Culp, Lechao Xiao, Maxwell L. Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yundi Qian, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research (TMLR)*, 2024.

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.

Qiaoyu Tang, Ziliang Deng, Hongyu Lin, Xianpei Han, Qiao Liang, Boxi Cao, and Le Sun. ToolAlpaca: Generalized tool learning for language models with 3000 simulated cases. *arXiv preprint arXiv:2306.05301*, 2023.

Team Olmo. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Jianze Wang, Ying Liu, Jinlong Chen, Xuchun Hu, Qilong Zhang, Yu Cao, Jun Wang, Hua Yang, Yong Xie, and Qianglong Chen. MAD-OPD: Breaking the ceiling in on-policy distillation via multi-agent debate. *arXiv preprint arXiv:2605.01347*, 2026.

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

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=yfcpdY4gMP.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models. *Transactions on Machine Learning Research (TMLR)*, 2022a.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022b.

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. *arXiv preprint arXiv:2402.13116*, 2024.

Zhangchen Xu, Yang Liu, Yueqin Yin, Mingyuan Zhou, and Radha Poovendran. KodCode: A diverse, challenging, and verifiable synthetic dataset for coding. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025.

An Yang, Baosong Yang, Binyuan Hui, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

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*, 2026.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Tianzhu Ye, Li Dong, Zewen Chi, Xun Wu, Shaohan Huang, and Furu Wei. Black-box on-policy distillation of large language models. *arXiv preprint arXiv:2511.10643*, 2025.

Tianzhu Ye, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. Online experiential learning for language models. *arXiv preprint arXiv:2603.16856*, 2026a.

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

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T. Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. MetaMath: Bootstrap your own mathematical questions for large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=2a36EMSSTp.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, Yu Fu, Xingtai Lv, Yuchen Zhang, Sihang Zeng, Shang Qu, Haozhan Li, Shijie Wang, Yuru Wang, Xinwei Long, Fangfu Liu, Xiang Xu, Jiaze Ma, Xuekai Zhu, Ermo Hua, Yihao Liu, Zonglin Li, Huayu Chen, Xiaoye Qu, Yafu Li, Weize Chen, Zhenzhao Yuan, Junqi Gao, Dong Li, Zhiyuan Ma, Ganqu Cui, Zhiyuan Liu, Biqing Qi, Ning Ding, and Bowen Zhou. A survey of reinforcement learning for large reasoning models. *arXiv preprint arXiv:2509.08827*, 2025.

Ruixiang Zhang, Richard He Bai, Huangjie Zheng, Navdeep Jaitly, Ronan Collobert, and Yizhe Zhang. Embarrassingly simple self-distillation improves code generation. *arXiv preprint arXiv:2604.01193*, 2026a.

Xinsen Zhang, Zhenkai Ding, Tianjun Pan, Run Yang, Chun Kang, Xue Xiong, and Jingnan Gu. Opsdl: On-policy self-distillation for long-context language models, 2026b. URL https://arxiv.org/abs/2604.17535.

Yaocheng Zhang, Yuanheng Zhu, Wenyue Chong, Songjun Tu, et al. π-Play: Multi-agent self-play via privileged self-distillation without external data. *arXiv preprint arXiv:2604.14054*, 2026c.

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. In *Proceedings of the 43rd International Conference on Machine Learning (ICML)*, 2026.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. ProcessBench: Identifying process errors in mathematical reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2025.

APPENDIX OVERVIEW

1. Appendix A: protocols of the three diagnostic probes.
2. Appendix B: proof of Proposition 1 and the implemented advantage.
3. Appendix C: DASD pseudocode.
4. Appendix D: privileged context, prompts, hyperparameters, baselines, metrics, and training cost.
5. Appendix E: Pass@16 on all benchmarks and per-benchmark gains.
6. Appendix F: evaluation protocol, thinking mode, and generation budget.
7. Appendix G: OPSD under different optimization exposure.
8. Appendix H: setting and additional results of the cross-domain and cross-family studies.
9. Appendix I: sensitivity to ρ and β, and ablation definitions.
10. Appendix J: future directions.
11. Appendix K: extended related work.
12. Appendix L: code, statistical reporting, compute, assets, and broader impact.

A PROBE PROTOCOLS FOR SECTION 3

All probes use Qwen3-1.7B trained on DAPO-Math-17k with a GRPO backbone and are measured on AIME24 rollouts. $\mathrm{StepAcc}$ and $E(y)$ are defined in Appendix D.3.

Probe 1: uniform sign. To separate the direction of teacher pressure from full-vocabulary leakage of privileged information, the probe uses the log-evidence gap of the emitted token,

$$A_t = \log \pi_\theta(o_t \mid x, r, o_{<t}) - \log \pi_\theta(o_t \mid x, o_{<t}), \tag{6}$$

and optimizes $\mathbb{E}_{o\sim\pi_\theta}[\sum_t s_t A_t]$ with a global sign: $s_t \equiv +1$ (Conformity) or $s_t \equiv -1$ (Novelty). Each run lasts 100 steps; $\mathrm{pass@16}$, $\mathrm{acc@16}$, $\mathrm{StepAcc}$, $E(y)$, and response length are tracked on intermediate checkpoints (Figure 2).

Probe 2: pressure versus entropy. For each token of the AIME24 rollouts, we record the student entropy $H_t = \mathcal{H}[\pi_\theta(\cdot \mid x, o_{<t})]$, the gap A_t of Eq. (6), and the realized one-step displacement $D_t^{(s)} = \mathrm{TV}\big(\pi_\theta(\cdot \mid x, o_{<t}),\ \pi_\theta^{(s\text{-step})}(\cdot \mid x, o_{<t})\big)$, where $\pi_\theta^{(s\text{-step})}$ is the student after one gradient step with sign $s \in \{+1, -1\}$. Figure 3 plots both quantities against H_t; ρ_s is the Spearman correlation between A_t and H_t.

Probe 3: prefix interventions. This probe modifies sampling from a fixed model and involves no training. Each token is assigned to the low- or high-entropy bucket by the trajectory-local median entropy $\tau_{0.5}$. For a token in the target bucket, the sampling distribution is replaced by either the privileged teacher, $\tilde{\pi}_t^{\mathrm{C}} = \pi_\theta(\cdot \mid x, r, o_{<t})$ (Conformity), or a distribution that down-weights teacher-favored tokens, $\tilde{\pi}_t^{\mathrm{N}}(v) \propto \pi_\theta(v \mid x, o_{<t}) / \pi_\theta(v \mid x, r, o_{<t})^\alpha$ with $\alpha = 0.5$ (Novelty). After the intervened token, generation resumes under the unprivileged student. The Random-H_t control applies the same interventions at entropy-agnostic positions. Table 1 reports relative changes in $\mathrm{StepAcc}$ and $E(y)$ with respect to unmodified rollouts.

B PROOF OF PROPOSITION 1

Fix a prompt x, prefix $o_{<t}$, privileged context r, and emitted token o_t. Write $p_\theta(v) = \pi_\theta(v \mid x, o_{<t})$ and $q(v) = \mathrm{sg}[\pi_\theta(v \mid x, r, o_{<t})]$, so that q is a fixed reference distribution during the update, and let $\delta_t = \log q(o_t) - \log p_\theta(o_t)$. The verifier advantage A_G and the routing coefficient ω_t are treated as fixed.

Proof. The local loss is $\ell_t(\theta) = -A_G \log p_\theta(o_t) + \beta\omega_t \,\mathrm{KL}(p_\theta \| q)$. The verifier term contributes $-A_G \nabla_\theta \log p_\theta(o_t)$. Because q does not depend on θ,

$$\mathrm{KL}(p_\theta \| q) = \sum_v p_\theta(v)\big(\log p_\theta(v) - \log q(v)\big),$$

and $\nabla_\theta p_\theta(v) = p_\theta(v)\nabla_\theta \log p_\theta(v)$ gives

$$\begin{aligned}\nabla_\theta \mathrm{KL}(p_\theta \| q) &= \sum_v p_\theta(v)\nabla_\theta \log p_\theta(v)\big(1 + \log p_\theta(v) - \log q(v)\big) \\ &= \mathbb{E}_{v\sim p_\theta}\left[(\log p_\theta(v) - \log q(v))\nabla_\theta \log p_\theta(v)\right],\end{aligned}$$

where the constant 1 vanishes because $\mathbb{E}_{v\sim p_\theta}[\nabla_\theta \log p_\theta(v)] = 0$. Estimating the expectation with the emitted token o_t yields

$$\nabla_\theta \mathrm{KL}(p_\theta \| q) \xrightarrow{\mathrm{MC}} (\log p_\theta(o_t) - \log q(o_t))\nabla_\theta \log p_\theta(o_t) = -\delta_t \nabla_\theta \log p_\theta(o_t).$$

Multiplying by $\beta\omega_t$ and adding the verifier term gives $\nabla_\theta \ell_t \overset{\mathrm{MC}}{\doteq} -\big(A_G + \beta\omega_t\delta_t\big)\nabla_\theta \log p_\theta(o_t)$. □

Implemented advantage. The implementation replaces δ_t with the trajectory-normalized gap $\bar{\delta}_t = \delta_t/(\tilde{\delta} + \epsilon)$, where $\tilde{\delta} = \mathrm{median}_t\,|\delta_t|$ is treated as a constant of the current rollout. The normalization preserves the sign of the teacher contrast and puts the correction on a comparable scale across trajectories, so the implemented advantage $\hat{A}_t = A_G + \beta\omega_t\bar{\delta}_t$ is the coefficient of Proposition 1 with the teacher term rescaled per trajectory. The gate $\sigma(|\bar{\delta}_t| - 1)$ in Eq. (4) uses the same normalization, so its threshold of 1 corresponds to the median absolute gap of the trajectory.

C Algorithm Pseudocode

Algorithm 1 gives one DASD training iteration in the notation of Section 4.

D Experimental Details

D.1 Privileged Context and Prompt Templates

All mathematical training uses DAPO-Math-17k (Yu et al., 2025), whose reference final answers define the binary verifier reward. The dataset contains no worked solutions, so r cannot be a reference solution as in Zhao et al. (2026). Following SDPO (Hübotter et al., 2026), r is a rollout of the same prompt, sampled in the current training step, that the verifier marked correct. It requires no annotation or stronger model and matches the student's own reasoning style. r is inserted only into the teacher prompt; the student never sees it during rollouts or evaluation. If none of the G rollouts of a prompt is correct, no r exists and the teacher term is set to zero; all G rewards are then 0, so A_G is also zero and the prompt contributes no gradient, as under GRPO. All self-distillation baselines receive r constructed in the same way.

We use the student and self-teacher templates of SDPO (Hübotter et al., 2026), which are identical to those of Kim et al. (2026). The student prompt is:

```
{question}
Please reason step by step, and put your final answer within
\boxed{}.
```

The privileged self-teacher scores the same student prefix under the prompt:

```
{question}
Please reason step by step, and put your final answer within
\boxed{}.
Correct solution:  {previously correct solution}
Correctly solve the original question.
```

D.2 Hyperparameters and Baselines

Table 6 lists the settings of the main mathematical experiments. Training and evaluation both use non-thinking mode (`enable_thinking=False`).

Algorithm 1 Direction-Adaptive Self-Distillation (DASD)

Require: policy π_θ and old policy $\pi_{\theta_{\text{old}}}$; verifier R; prompts $x \sim \mathcal{D}$; rollout count G; router quantile ρ (default 0.20); coefficients β, ϵ; PPO clip ϵ_{clip}

for each training batch **do**
  Sample rollouts $\{o^{(i)}\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$ without privileged context
  Compute group-relative verifier advantage $A_G^{(i)} \leftarrow (R(x, o^{(i)}) - \mu_G)/\sigma_G$ over the G rollouts for prompt x
  ▷ Main experiments use $G = 8$ for every method.
  Select privileged context r: a rollout $o^{(j)}$ with $R(x, o^{(j)}) = 1$; if none exists, skip the teacher term for x ($\hat{A}_t^{(i)} = A_G^{(i)}$) ▷ Appendix D
  for each rollout i and token position t **do**
    $p_t^{(i)} \leftarrow \pi_\theta(\cdot \mid x, o_{<t}^{(i)})$ ▷ student branch, gradient on
    $q_t^{(i)} \leftarrow \text{sg}[\pi_\theta(\cdot \mid x, r, o_{<t}^{(i)})]$ ▷ privileged self-teacher, stop-gradient
    $H_t^{(i)} \leftarrow \mathcal{H}(p_t^{(i)})$
    $\delta_t^{(i)} \leftarrow \log q_t^{(i)}(o_t^{(i)}) - \log p_t^{(i)}(o_t^{(i)})$
  end for
  for each rollout i **do**
    $\tau_\rho^{(i)} \leftarrow \text{Quantile}_\rho(\{H_t^{(i)}\}_t)$
    $\hat{\sigma}_H^{(i)} \leftarrow \text{MeanAbsDev}(\{H_t^{(i)}\}_t)$
    $\tilde{\delta}^{(i)} \leftarrow \text{median}_t\, |\delta_t^{(i)}|$
    for each token t **do**
      $\bar{\delta}_t^{(i)} \leftarrow \delta_t^{(i)}/(\tilde{\delta}^{(i)} + \epsilon)$
      $g_t^{(i)} \leftarrow \sigma(|\bar{\delta}_t^{(i)}| - 1)$ ▷ gap-reliability gate
      $z_t^{(i)} \leftarrow \tanh\Big((\tau_\rho^{(i)} - H_t^{(i)})/(\hat{\sigma}_H^{(i)} + \epsilon)\Big)$ ▷ positive at low entropy, negative at high entropy
      $\omega_t^{(i)} \leftarrow z_t^{(i)} g_t^{(i)}$
      $\hat{A}_t^{(i)} \leftarrow A_G^{(i)} + \beta \omega_t^{(i)} \bar{\delta}_t^{(i)}$
      $\kappa_t^{(i)} \leftarrow \pi_\theta(o_t^{(i)} \mid x, o_{<t}^{(i)})/\pi_{\theta_{\text{old}}}(o_t^{(i)} \mid x, o_{<t}^{(i)})$
    end for
  end for
  Update θ by maximizing

$$\mathbb{E}_{i,t}\Big[\min\Big(\kappa_t^{(i)} \hat{A}_t^{(i)},\ \text{clip}(\kappa_t^{(i)}, 1 - \epsilon_{\text{clip}}, 1 + \epsilon_{\text{clip}})\hat{A}_t^{(i)}\Big)\Big].$$

end for
return updated policy π_θ

Table 6: Training and evaluation settings for the main mathematical experiments.

Shared by all methods		**DASD-specific**	
Framework	`verl`	Router quantile ρ	0.20 (per rollout)
Prompts per step B	128	Teacher weight β	1
Rollouts per prompt G	8	Gate	$\sigma(\lvert\bar{\delta}_t\rvert - 1)$
Mini-batch size	32	Normalization	median $\lvert\delta_t\rvert$ per trajectory
Learning rate	3×10^{-6}, cosine decay		
Training steps	200	**Evaluation**	
Decoding mode	non-thinking	Samples per problem K	16
Reported checkpoint	final	Temperature / top-p	1.0 / 0.9
		Generation limit	8,192 tokens

Baselines. Baseline-specific hyperparameters follow the original papers, and all self-distillation baselines condition the teacher on the same r as DASD. GRPO uses a sequence-level verifier advantage. HEPO restricts policy-gradient updates to high-entropy forking tokens. OPSD minimizes Eq. (2) with D the full-vocabulary forward KL, per-token pointwise clipping, and the teacher branch

frozen at the initial policy, following Zhao et al. (2026); its objective contains no reward term. SDPO replaces the GRPO advantage with the teacher–student log-ratio $\log \pi_\theta(o_t \mid x, r, o_{<t}) - \log \pi_\theta(o_t \mid x, o_{<t})$, using the default objective of Hübotter et al. (2026) rather than its optional mixture with the GRPO advantage. In OPSD and SDPO the verifier therefore acts only through the selection of r. RLSD converts the teacher–student log-evidence ratio into a token-level credit adjustment inside a verifier-anchored update. SRPO applies GRPO-style reinforcement to correct rollouts and SDPO-style correction to failed rollouts, with entropy-aware weighting.

D.3 Execution and Exploration Metrics

$\mathrm{StepAcc}$ is the fraction of reasoning steps judged correct by a step-level critic that follows the ProcessBench protocol (Zheng et al., 2025): each response is split into steps at paragraph breaks, and the critic receives the problem together with the numbered steps. First-error step is the mean index of the first incorrect step, so larger values mean later errors. Correct-step ratio is the fraction of the same steps accepted by a second step judge, independent of the ProcessBench critic. $E(y)$ is the number of epistemic markers (e.g., "wait", "hmm", "alternatively") in a response y. Revision rate is the fraction of responses in which an epistemic marker is followed by a substantively different continuation, operationalized as a trigram overlap of at most 0.3. Distinct-3 is the ratio of unique to total trigrams across the 16 rollouts of a prompt. Table 3 computes these metrics on AIME24 rollouts of the final checkpoints.

D.4 Training Cost

DASD adds one forward pass of the policy on the privileged prompt per rollout, as do OPSD, SDPO, and RLSD. On Qwen3-1.7B with $G = 8$, one training step including rollout generation takes 109.0 s for GRPO, 127.0 s for OPSD, and 120.2 s for DASD on the hardware of Appendix L; DASD is thus 10% slower per step than GRPO.

E Pass@16 and Per-Benchmark Gains

Table 7 reports Pass@16 for the runs of Table 2. DASD has the best macro Pass@16 at every scale (57.9, 68.0, and 69.0 for Qwen3-1.7B, 4B, and 8B), and OPSD and SDPO end 1.0 to 2.3 points below GRPO, as under Avg@16. The one benchmark where DASD is not best is AIME25 at 4B, where GRPO reaches 86.7 against 83.3.

Table 7: Pass@16 (%) on all six mathematical reasoning benchmarks for the runs of Table 2. **Bold**/underline mark best/second-best within each scale.

Method	MATH500	Minerva	HMMT25	Olympiad	AIME24	AIME25	Avg
Qwen3-1.7B	73.4	32.0	33.3	53.4	26.7	30.0	41.5
+GRPO (Shao et al., 2024)	75.6	37.5	23.3	54.3	53.3	56.7	50.1
+HEPO (Wang et al., 2025)	76.2	36.8	36.7	56.4	60.0	60.0	54.3
+OPSD (Zhao et al., 2026)	75.0	36.4	26.7	53.1	50.0	53.3	49.1
+SDPO (Hübotter et al., 2026)	74.8	35.7	23.3	54.8	50.0	53.3	48.7
+RLSD (Yang et al., 2026)	75.6	39.3	33.3	52.8	60.0	46.7	51.3
+SRPO (Li et al., 2026a)	74.8	35.7	30.0	53.9	50.0	43.3	47.9
+DASD (ours)	**77.8**	**40.4**	**40.0**	**59.0**	**66.7**	**63.3**	**57.9**
Qwen3-4B	76.2	36.0	36.7	58.6	56.7	43.3	51.2
+GRPO (Shao et al., 2024)	76.6	38.2	50.0	57.0	80.0	**86.7**	64.8
+HEPO (Wang et al., 2025)	76.6	36.8	43.3	61.6	80.0	76.7	62.5
+OPSD (Zhao et al., 2026)	76.2	37.1	46.7	58.2	76.7	80.0	62.5
+SDPO (Hübotter et al., 2026)	76.0	38.6	46.7	56.8	76.7	83.3	63.0
+RLSD (Yang et al., 2026)	75.8	37.1	40.0	54.0	76.7	76.7	60.0
+SRPO (Li et al., 2026a)	76.4	36.8	43.3	57.9	66.7	60.0	56.8
+DASD (ours)	**78.4**	**41.2**	**53.3**	**65.0**	**86.7**	83.3	**68.0**
Qwen3-8B	75.6	34.6	33.3	59.8	53.3	40.0	49.4
+GRPO (Shao et al., 2024)	75.6	37.5	46.7	60.7	80.0	80.0	63.4
+HEPO (Wang et al., 2025)	76.2	37.1	43.3	60.5	86.7	76.7	63.4
+OPSD (Zhao et al., 2026)	75.4	36.8	43.3	61.0	76.7	76.7	61.7
+SDPO (Hübotter et al., 2026)	75.8	36.4	43.3	59.8	80.0	76.7	62.0
+RLSD (Yang et al., 2026)	76.2	39.7	43.3	59.2	83.3	76.7	63.1
+SRPO (Li et al., 2026a)	76.0	36.0	43.3	59.8	66.7	60.0	57.0
+DASD (ours)	**78.8**	**42.6**	**53.3**	**65.6**	**90.0**	**83.3**	**69.0**

Figure 7 averages the Avg@16 gains of Table 2 over the three scales. The gain over GRPO ranges from 2.5 points on MATH500 to 7.2 on OlympiadBench, and the gain over OPSD from 3.9 to 8.1 on the same two benchmarks.

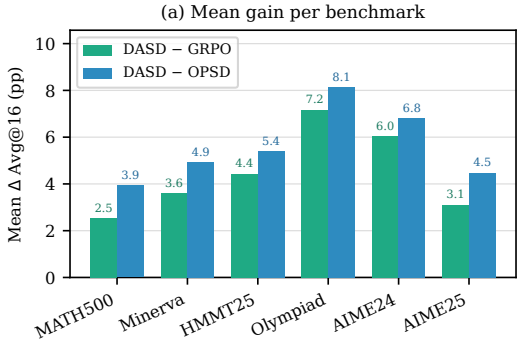


Figure 7: Mean Avg@16 gain of DASD over GRPO and over OPSD on each benchmark, averaged over the three model scales of Table 2. Both gains are largest on OlympiadBench and AIME24 and smallest on MATH500.

F Evaluation Protocol, Thinking Mode, and Generation Budget

Effect of the evaluation protocol. Reported accuracies of Qwen3 models depend strongly on the decoding mode and the generation limit. Table 8 evaluates the untrained Qwen3-1.7B checkpoint under our main protocol and under the thinking-mode protocol of Zhao et al. (2026) (38,912-token limit, top-$p = 0.95$, 12 samples per problem). Under the latter, our evaluation is within 2 points of their reported numbers, so the lower absolute accuracies in Table 2 reflect the non-thinking 8,192-token protocol, not a weaker checkpoint.

Table 8: The untrained Qwen3-1.7B checkpoint under two evaluation protocols (accuracy, %).

Protocol	Source	AIME24	AIME25
Ours: non-thinking, 8,192 tok., top-p 0.9, Avg@16	our evaluation	13.1	10.6
Thinking, 38,912 tok., top-p 0.95, Avg@12	our evaluation	50.6	34.7
Thinking, 38,912 tok., top-p 0.95, Avg@12	Zhao et al. (2026)	51.5	36.7

Thinking-mode training and evaluation. Qwen3-1.7B is trained with thinking enabled for 200 steps with 512 sequences per step and evaluated in thinking mode with a 32K-token limit and 16 samples per problem (Table 9). DASD exceeds GRPO by 5.9 and 5.2 points on AIME24 and AIME25, whereas OPSD trails GRPO by 5.2 and 6.7 points.

Table 9: Thinking-mode training and evaluation on Qwen3-1.7B (Avg@16, %; 32K-token limit). †Untrained checkpoint under the protocol of Table 8 (Avg@12, 38,912-token limit), shown for reference only; comparisons are among the trained rows.

Method	AIME24	AIME25
Qwen3-1.7B (thinking)†	50.6	34.7
+GRPO	53.1	39.2
+OPSD	47.9	32.5
+DASD (ours)	**59.0**	**44.4**

Generation budget. Table 10 re-evaluates the non-thinking Qwen3-1.7B GRPO, RLSD, and DASD checkpoints with 16K and 32K-token limits (16 samples per problem). DASD stays ahead of GRPO and RLSD at both limits.

Table 10: Non-thinking Qwen3-1.7B checkpoints evaluated with larger generation budgets (Avg@16, %).

Benchmark	Budget	GRPO	RLSD	DASD
AIME24	16K	32.5	38.5	**40.2**
AIME24	32K	33.5	39.4	**41.0**
AIME25	16K	28.3	28.8	**31.7**
AIME25	32K	29.4	28.5	**30.6**

G Self-Distillation under Prolonged Optimization

Optimization exposure. Zhao et al. (2026) train OPSD with LoRA on 32 sequences per step for 100 steps and report the best checkpoint; our main runs distill 1,024 rollouts per step ($B = 128$, $G = 8$) for 200 steps and report the final checkpoint. Table 11 isolates the per-step volume. Every run follows the setup of Zhao et al. (2026) (LoRA rank 64, $\alpha = 128$, 100 steps, thinking-mode evaluation) and varies only the number of sequences per step and the learning rate. With 32 sequences per step, OPSD improves on the untrained checkpoint (50.6) at the two smaller learning rates and reaches 56.4 at 5×10^{-6}, close to the 57.2 that Zhao et al. (2026) report for this configuration (their best checkpoint within 100 steps). With 1,024 sequences per step, it ends below the untrained checkpoint at every learning rate, and larger learning rates degrade it further.

Table 11: OPSD on Qwen3-1.7B after 100 training steps, evaluated with the thinking-mode protocol of Zhao et al. (2026) (AIME24 Avg@12, %). The untrained checkpoint scores 50.6 under this protocol.

Learning rate	32 seq./step	1,024 seq./step
1×10^{-6}	54.2	48.1
5×10^{-6}	**56.4**	44.2
1×10^{-5}	48.1	36.9

H Cross-Domain and Cross-Family Evaluation

The cross-domain study (Table 4) changes the domain and fixes the backbone to Qwen3-8B; the cross-family study (Table 14) fixes the domain to mathematics and changes the backbone.

Cross-domain setting. Table 12 lists the training data and evaluation of each domain. GRPO, OPSD, and DASD start from Qwen3-8B in non-thinking mode and share the training data and optimization setting of each domain: 200 steps, 32 prompts per step, $G = 8$ rollouts per prompt, and learning rate 10^{-6}. DASD keeps $\rho = 0.20$ and $\beta = 1$ from the mathematical experiments. The science and tool-use training sets follow prior self-distillation work (Hübotter et al., 2026; Shenfeld et al., 2026).

Each domain uses a binary verifier reward computed from the reference data. *Code:* the program in the final code block is executed against the unit tests shipped with Skywork-OR1 and KodCode, and $R = 1$ if it passes all of them. *Science:* SciKnowEval questions are four-option multiple choice answered in the `<answer>` format of Hübotter et al. (2026), and $R = 1$ if the extracted letter matches the reference. *Tool use:* the model answers in the ToolAlpaca `Action`/`Action Input` format, and $R = 1$ if the called tool and the parsed JSON arguments match the reference. The privileged context r is constructed as in mathematics (Appendix D.1), so the teacher receives no error messages, test outputs, or other environment feedback, and the domains differ from mathematics only in the verifier.

Table 12: Training data and evaluation of the cross-domain study. All runs start from Qwen3-8B (non-thinking).

Domain	Training data	Evaluation benchmarks	Metric
Code generation	Skywork-OR1 (He et al., 2025) + KodCode (Xu et al., 2025)	HumanEval+, MBPP+ (Liu et al., 2023); LiveCodeBench v6 (Jain et al., 2025)	EvalPlus pass@1; pass@1
Scientific reasoning	SciKnowEval (Feng et al., 2024), four subjects merged (3,901)	GPQA Diamond (Rein et al., 2024); MMLU, full test set (Hendrycks et al., 2021a)	accuracy
Tool use	ToolAlpaca-based tool-use set (Tang et al., 2023; Shenfeld et al., 2026) (4,046)	BFCL (Patil et al., 2025); API-Bank Level-1, offline subset (Li et al., 2023)	overall score; score

BFCL breakdown. BFCL uses a 128K-token context in non-thinking mode. DASD is best in every BFCL category (Table 13).

Table 13: BFCL overall and category scores on Qwen3-8B (non-thinking).

Method	Overall	Non-live	Live	Multi-turn
Qwen3-8B (base)	51.58	84.13	65.35	6.25
+GRPO	<u>54.78</u>	<u>86.17</u>	<u>69.17</u>	<u>8.00</u>
+OPSD	51.47	83.38	66.39	7.63
+DASD (ours)	**56.50**	**88.27**	**70.75**	**8.50**

Cross-family setting. Olmo-3-7B-Instruct and Llama-3.1-8B-Instruct are trained with GRPO, OPSD, and DASD on DAPO-Math-17k under the main protocol of Section 5; the Qwen3-8B rows repeat Table 2. DASD has the best average on all three backbones (Table 14). OPSD ends above the untrained model on Qwen3-8B and Olmo-3-7B-Instruct but below it on Llama-3.1-8B-Instruct. On Llama-3.1-8B-Instruct, AIME accuracy stays below 5% for every method, and the gain of DASD comes mainly from MATH500 (36.58 vs. 34.79 for GRPO).

Table 14: Cross-family evaluation on mathematics (Avg@16, %). All backbones are trained on DAPO-Math-17k; Qwen3-8B rows are from Table 2. Avg is the mean of the three benchmarks. **Bold**: best within each backbone.

Model / method	MATH500	AIME24	AIME25	Avg
Qwen3-8B	67.2	27.9	21.0	38.7
+GRPO	73.0	59.4	48.8	60.4
+OPSD	71.7	58.3	48.3	59.4
+DASD (ours)	**76.0**	**71.7**	**53.3**	**67.0**
Olmo-3-7B-Instruct	69.89	42.92	31.04	47.95
+GRPO	71.98	51.88	40.83	54.90
+OPSD	68.58	45.83	35.83	50.08
+DASD (ours)	**73.58**	**54.17**	**42.50**	**56.75**
Llama-3.1-8B-Instruct	33.79	2.08	0.63	12.17
+GRPO	34.79	3.13	0.63	12.85
+OPSD	30.58	0.00	0.00	10.19
+DASD (ours)	**36.58**	**4.17**	**1.67**	**14.14**

I Sensitivity and Ablation Details

I.1 Router Quantile ρ

The quantile ρ sets the trajectory-local threshold $\tau_\rho^{(i)}$: tokens below it are attracted to the teacher and tokens above it are repelled. Table 15 varies ρ with all other settings fixed. The default $\rho = 0.20$ is best, $\rho = 0.50$ is within 3.4 points on every metric, and accuracy drops at both extremes, where almost all tokens receive the same direction.

Table 15: Sensitivity of DASD to the router quantile ρ (Qwen3-1.7B). Small ρ repels most tokens (Novelty-biased); large ρ attracts most tokens (Conformity-biased).

Router quantile ρ	MATH500		AIME24		AIME25	
	Avg@16	Pass@16	Avg@16	Pass@16	Avg@16	Pass@16
0.05 (Novelty-biased)	67.9	75.6	26.0	60.0	22.7	53.3
0.20 (default)	**70.4**	**77.8**	**35.0**	**66.7**	**32.1**	**63.3**
0.50	69.9	77.4	33.5	63.3	30.4	60.0
0.75	69.1	76.6	29.8	63.3	26.7	56.7
0.95 (Conformity-biased)	67.7	75.2	21.5	53.3	19.0	50.0

I.2 Teacher Weight β

The coefficient β in Eq. (5) weights the signed term against the verifier term; $\beta = 0$ recovers GRPO. DASD improves over GRPO at every tested $\beta > 0$, and accuracy peaks at the default $\beta = 1$ (Table 16).

Table 16: Sensitivity of DASD to β on Qwen3-1.7B (Avg@16, %).

Setting	AIME24	AIME25
Qwen3-1.7B (no training)	13.1	10.6
GRPO ($\beta = 0$)	32.5	27.5
DASD, $\beta = 0.5$	34.4	30.2
DASD, $\beta = 1$ (default)	**35.0**	**32.1**
DASD, $\beta = 2$	33.1	29.8

I.3 Ablation Definitions

All variants in Table 5 are trained under the shared protocol.

Left panel. *Router + noise* keeps the entropy router $z_t = \tanh((\tau_\rho - H_t)/(\hat{\sigma}_H + \epsilon))$ of Eq. (4) but replaces the teacher correction $g_t\bar{\delta}_t$ in $\hat{A}_t$ with independent zero-mean noise, normalized per trajectory in the same way. *Router + entropy regularizer* removes every teacher-dependent quantity (q_t, δ_t, and the gate) and adds $\lambda z_t H_t$ to the GRPO loss, which sharpens low-entropy tokens ($z_t > 0$) and raises entropy at high-entropy tokens ($z_t < 0$) without a privileged forward pass. *Low-entropy attraction only* keeps A_G and applies the attractive teacher term only to tokens with $z_t > 0$. *Directional only* removes A_G but keeps the same privileged context r, so the update is driven entirely by the signed teacher correction.

Right panel. Each block replaces one component while keeping the other two at their defaults. *A. Routing signal* replaces H_t as the routing variable with sinusoidal relative position, inverse training-corpus token frequency, the gradient norm $\|\nabla_\theta \log \pi_\theta(o_t)\|$, or the mean attention-head entropy at token t. *B. Direction map* replaces the $\tanh$ map with a constant -1 (uniform repulsion), a hard threshold $\mathrm{sign}(\tau_\rho^{(i)} - H_t)$, or a clipped linear map from normalized entropy to $[-1, 1]$. *C. Reliability gate* replaces $\sigma(|\bar{\delta}_t| - 1)$ with no gate, a hand-tuned fixed threshold $\mathbf{1}[|\delta_t| > c]$, or the unthresholded magnitude $|\bar{\delta}_t|$.

J Future Directions

Student entropy is a hand-designed proxy for decision points. A learned router, weaker privileged signals such as partial hints or environment feedback, and other dense-supervision settings, including distillation from external teachers, process reward models, and agentic post-training, would test how far direction-adaptive supervision extends beyond privileged self-distillation on verifiable tasks.

K Extended Related Work

RLVR and outcome-level credit assignment. RLVR builds on PPO (Schulman et al., 2017), generalized advantage estimation (Schulman et al., 2016), and RLHF (Christiano et al., 2017; Ouyang et al., 2022), but replaces the learned reward model with a programmatic verifier on tasks whose final answers can be checked (Cobbe et al., 2021). GRPO (Shao et al., 2024) and refinements such as DAPO (Yu et al., 2025) underlie recent reasoning models including DeepSeek-R1 (Guo et al., 2025), Qwen3 (Yang et al., 2025), and Olmo 3 (Team Olmo, 2025). The verifier is reliable but sparse: every token in a trajectory inherits the same sequence-level signal.

Process-level and token-level densification. Process reward models and step-level value estimators score intermediate steps (Uesato et al., 2022; Lightman et al., 2024; Wang et al., 2024; Zheng et al., 2025) at the cost of an additional learned model. Verifier-only methods instead use model-internal proxies to reallocate credit: high-entropy forking tokens (Wang et al., 2025), entropy-based exploration (Cheng et al., 2026), entropy and reward polarity (He et al., 2026b), signed base-to-RLVR change (Huang et al., 2026), and low-entropy tokens (Liu et al., 2026; Chen et al., 2026b). Entropy-aware distillation and SRPO use uncertainty to focus or filter supervision (Jin et al., 2026; Li et al., 2026a). These methods use entropy to decide how much a token is updated; DASD uses it to decide the direction of teacher pressure.

On-policy distillation and privileged self-teachers. On-policy distillation (OPD) (Agarwal et al., 2024) extends knowledge distillation (Hinton et al., 2015; Kim & Rush, 2016; Furlanello et al., 2018) by scoring the student's own rollouts with a stronger teacher. Recent work surveys its design space (Song & Zheng, 2026), characterizes when it succeeds (Li et al., 2026b), distills from context or deployment experience (Ye et al., 2026b;a), relaxes white-box teacher access (Ye et al., 2025; Chen et al., 2026c), uses multiple debating teachers (Wang et al., 2026), analyzes failure modes of sampled-token OPD (Fu et al., 2026), and balances forward and reverse KL by student entropy (Jin et al., 2026). Self-distillation variants compress reasoning (Sang et al., 2026), turn binary rewards into dense supervision through self-revision (He et al., 2026a), and extend to long-context and code settings (Zhang et al., 2026b;a). OPSD removes the external teacher by conditioning the same model on privileged information (Zhao et al., 2026; Hübotter et al., 2026; Li et al., 2026a); related formulations include a joint teacher–student objective (Penaloza et al., 2026), consensus gating of unreliable tutor signals (Stein et al., 2026), multi-agent self-play (Zhang et al., 2026c), and RLSD's reading of the teacher–student log-evidence ratio as token-level credit (Yang et al., 2026), the formulation closest to DASD. Uniform attraction to a solution-conditioned teacher suppresses epistemic markers and shortens responses (Kim et al., 2026; Yang et al., 2026; Fu et al., 2026). The methods above adjust the magnitude, filtering, scheduling, or placement of teacher pressure, and in all of them the pressure remains attractive.

Reasoning elicitation and self-improvement. Chain-of-thought prompting (Wei et al., 2022b; Kojima et al., 2022; Wei et al., 2022a), self-consistency (Wang et al., 2023), and tree search (Yao et al., 2023) elicit reasoning at inference time; STaR (Zelikman et al., 2022), ReST-style self-training (Singh et al., 2024), and Self-Refine (Madaan et al., 2023) bootstrap it from the model's own outputs, and data or tool-use pipelines such as MetaMath (Yu et al., 2024), WizardMath (Luo et al., 2025), and ToRA (Gou et al., 2024) target mathematics directly. Huang et al. (2024) find that intrinsic self-correction without external feedback can degrade performance, which motivates keeping the verifier as DASD's outcome anchor.

L Code, Statistical Reporting, Compute, Assets, and Broader Impact

Code. The anonymous repository (`https://anonymous.4open.science/r/ICLR2027DASD37656`) contains the DASD training and evaluation code, configuration files for every reported method, and scripts that rebuild the main tables and figures from per-question logs. The paper introduces no new dataset or model checkpoint.

Statistical reporting. All reported accuracies average over the samples drawn per problem on fixed public benchmark instances, with one training run per configuration.

Compute. All runs use 8 NVIDIA H20 GPUs with 96 GB memory each; per-step training time is given in Appendix D.4.

Existing assets. The experiments use public assets, credited by citation: Qwen3 (Yang et al., 2025), Olmo 3 (Team Olmo, 2025), and Llama 3.1 (Grattafiori et al., 2024) models; DAPO-Math-17k (Yu et al., 2025); MATH500 (Hendrycks et al., 2021b), Minerva (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), AIME24/AIME25, and HMMT25; ProcessBench (Zheng et al., 2025); EvalPlus (Liu et al., 2023), LiveCodeBench (Jain et al., 2025), GPQA (Rein et al., 2024), MMLU (Hendrycks et al., 2021a), BFCL (Patil et al., 2025), and API-Bank (Li et al., 2023); the training sets in Table 12; and `verl` (Sheng et al., 2025). The repository records the license or terms of use of each asset.

Broader impact. The study uses public benchmarks and involves no private data, crowdsourcing, or human subjects. More effective reasoning post-training could also benefit misuse of such models; we release training code rather than new model weights or datasets.