UNMASKING ON-POLICY DISTILLATION: WHERE IT HELPS, WHERE IT HURTS, AND WHY

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation offers dense, per-token supervision for training reasoning models, but it remains unclear when this signal helps and when it hurts. Which teacher should be used, and in self-distillation, which context should serve as the supervisory signal? Answering these questions typically requires costly training runs whose aggregate metrics obscure what happens at individual tokens. We introduce a training-free diagnostic framework that operates per token, per question, and per teacher. We derive an ideal per-node gradient, the update that maximally increases the student's probability of success, and develop a scalable targeted-rollout algorithm to estimate it even for long reasoning chains. The **gradient alignment score**, the cosine similarity between this ideal gradient and a distillation gradient, quantifies how closely a configuration approximates the ideal signal. In distillation training runs with external teachers, the teacher ranking under the better-performing training configuration largely agrees with the score. Across self-distillation settings and external teachers, distillation guidance is more aligned with the ideal on incorrect rollouts than on correct ones, and the best distillation context depends jointly on the student's capacity and the task, motivating per-task, per-token diagnostics.

1 INTRODUCTION

On-policy distillation has become a core post-training technique for reasoning models: Qwen3 (Yang et al., 2025), MiMo (Xiao et al., 2026), and GLM-5 (Zeng et al., 2026) adopt it in their pipelines, and concurrent works (Hübotter et al., 2026; Zhao et al., 2026; Ye et al., 2026; Shenfeld et al., 2026) report strong gains from self-distillation variants. The student is guided at every token by a teacher's distribution (Agarwal et al., 2024; Lu & Lab, 2025): in *teacher distillation*, a larger model provides supervision (Hinton et al., 2015), while in *self-distillation* the student serves as its own teacher with extra context (such as a correct solution) unavailable at test time. Both complement the sparse binary reward of RL methods like GRPO (Shao et al., 2024; Guo et al., 2025) with a dense gradient at every token.

Yet practitioners face decisions with little principled guidance: should the teacher be a larger external model or the student itself with access to a correct solution, and should that solution be shown in full or summarized? Today, these questions require expensive training runs whose aggregate metrics hide what happens at individual tokens. As Figure 1 shows, even within a single reasoning trajectory the teacher's disagreements mix beneficial, neutral, and harmful signals, which cannot be told apart without linking each token to its downstream outcome. We therefore seek to measure, *per token, per question, and per teacher configuration*, how well the teacher's guidance aligns with the behavior that yields correct answers.

We derive an ideal per-token gradient from empirical success probabilities, the direction that maximally improves the student's chance of reaching a correct answer, and show that Dr. GRPO (Liu et al., 2025) recovers it in expectation (Section 2.2). Major distillation objectives (GKD (Agarwal et al., 2024), the single-sample estimator of Lu & Lab (2025), MiniLLM (Gu et al., 2024)) share the same local gradient structure, with the signal coming from success probability for reward-based methods and from the teacher's distribution for distillation. A targeted-rollout algorithm with exponential depth windows estimates the ideal gradient even for long reasoning chains at a user-chosen budget, and the **gradient alignment score**, the cosine between the ideal and distillation gradients

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

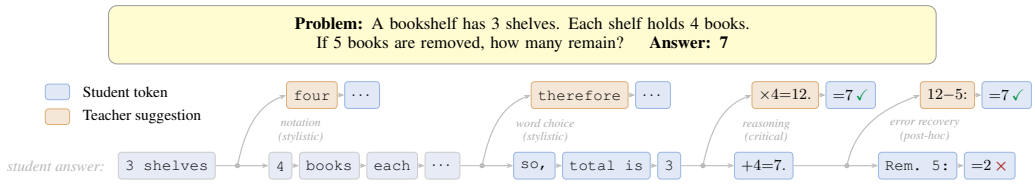


Figure 1: **Not all teacher guidance points toward success.** Generation tree for a bookshelf problem. The teacher's distribution disagrees with the student at four branching points (orange), but not all disagreements are equal: some reflect stylistic preferences ("four" vs. "4", "therefore" vs. "so,") rather than reasoning corrections. Standard distillation treats all four signals equally, mixing noise from irrelevant preferences with updates that actually matter.

at each token, evaluates any teacher configuration offline (Section 2.4). Finally, we check the score against distillation training runs with external teachers (Section 4.3).

Key findings. Applying this framework to Qwen3-0.6B and Qwen3-1.7B across 8 teacher configurations on BoolQ and MMLU, we find that:

- **The score agrees with training for external teachers.** In distillation runs with three external teachers and both students, the teacher ranking under the better-performing training configuration largely agrees with the score, while the ranking from an individual run can shift with training hyperparameters such as the learning rate.
- **Distillation guidance is more reliable on incorrect rollouts.** On correct paths the teacher's signal is weakly aligned with the ideal; on failing rollouts it pushes more consistently toward success, in every setting.
- **The best teacher depends on student capacity and context.** Summarizing the demonstration helps the 1.7B student but not 0.6B, and external teachers match the best self-distillation context only for the larger student. One hypothesis is *comprehensibility*: the signal is only useful if the student can parse what it is given.
- **No universal recipe exists.** Wrong demonstrations lower alignment on BoolQ and MMLU but rank first on some AIME case-study questions, so the best configuration shifts with the task, motivating per-task diagnostics.

2 Methodology

2.1 Not all teacher guidance is useful

The teacher's distribution may differ from the student's for many reasons: a stylistic preference, a nudge along a productive reasoning path, or a redirection toward a different continuation. These cases cannot be distinguished from the teacher's probabilities alone, and only some improve the student's chance of reaching a correct answer (Figure 1). To tell them apart, we connect the teacher's token-level signal to downstream outcomes.

Given G trajectories sampled from the student π_θ on a prompt q, shared prefixes form a *generation tree* in which each node u is a token position and each edge a next-token choice. We write P_θ^k for the student's probability of token k at u, obtained by a softmax over its logits z, with z_j the logit of token j (Appendix A lists all notation). Let N_u^k be the number of rollouts through u that choose k and S_u^k the number of those that reach a correct answer; the *success probability* P_{succ}^k of choosing k at u is estimated as $\hat{P}_{\text{succ}}^k = S_u^k / N_u^k$. Because targeted rollouts (Section 2.5) are student samples conditioned on a prefix, this estimate stays unbiased for each child even when some children are sampled more than others.

With this quantity in hand, we can ask precisely: does the teacher push probability mass toward high-P_{succ} tokens, or away from them?

Independent of any training algorithm, a teacher is good at node u if it places more mass on success-leading tokens than the student does. We define the **teacher advantage**:

$$\mathrm{Adv}(u) = \sum_{k \in \mathcal{S}_u} P^k_{\mathrm{te}}\, \hat{P}^k_{\mathrm{succ}} - \sum_{k \in \mathcal{S}_u} P^k_{\theta}\, \hat{P}^k_{\mathrm{succ}}, \tag{2.1}$$

where $\mathcal{S}_u$ is the set of tokens at node u with at least N_{sig} visits ($N_{\mathrm{sig}} = 20$ in our experiments; Section 2.5), P^k_{te} is the teacher's probability of token k, and both distributions are renormalized over $\mathcal{S}_u$. At 20 visits, the standard error of each child's success estimate is at most about 0.11. A positive advantage means the teacher "knows better" at this node; a negative advantage means following it would hurt.

A good teacher is not sufficient, however: one also needs an algorithm that translates the teacher's knowledge into a useful gradient, and different algorithms (GKD (Agarwal et al., 2024), single-sample estimators (Lu & Lab, 2025), MiniLLM (Gu et al., 2024)) use the teacher differently, producing very different gradients from the same teacher. To evaluate any (teacher, algorithm) pair, we need an **ideal reference**: the gradient that would optimally improve the student's success probability at each node. Unless stated otherwise, the alignment scores we report use the GKD gradient.

2.2 The ideal reference gradient

At each node u, the ideal local objective is to maximize the student's probability of reaching a correct answer from this point:

$$L_{\mathrm{ideal}}(u) = \sum_k P^k_{\theta}\, P^k_{\mathrm{succ}}. \tag{2.2}$$

This is the expected success rate under the student's current distribution at node u. Since L_{ideal} is maximized, we refer to it as an objective rather than a loss. Throughout the paper, every gradient we compare is expressed as an ascent direction on the student's success probability, so distillation losses, which are minimized, enter with their sign flipped. The gradient of this objective with respect to the student's logit z_j at this node is obtained via the softmax Jacobian $\partial P^k_{\theta}/\partial z_j = P^k_{\theta}(\delta_{kj} - P^j_{\theta})$, where δ_{kj} is the Kronecker delta (1 if $k = j$, 0 otherwise):

$$\begin{aligned} \frac{\partial L_{\mathrm{ideal}}}{\partial z_j} &= \sum_k P^k_{\mathrm{succ}} \cdot P^k_{\theta}(\delta_{kj} - P^j_{\theta}) \\ &= P^j_{\theta}\, P^j_{\mathrm{succ}} - P^j_{\theta} \sum_k P^k_{\theta} P^k_{\mathrm{succ}} \\ &= P^j_{\theta} \left(P^j_{\mathrm{succ}} - \bar{P}_{\mathrm{succ}} \right), \end{aligned} \tag{2.3}$$

where $\bar{P}_{\mathrm{succ}} = \sum_k P^k_{\theta} P^k_{\mathrm{succ}}$ is the student's current expected success at this node. This gradient increases logit z_j when token j leads to success more often than average, and decreases it otherwise. This is our reference: the direction in which the student's logits should move to maximally improve its chance of success at this node.

Dr. GRPO recovers this gradient in expectation. A natural question is whether any existing training objective already computes this ideal gradient. Dr. GRPO (Liu et al., 2025) is a variant of GRPO that removes the per-trajectory length normalization $1/|o_i|$. The full GRPO objective includes an importance ratio $\pi_{\theta}/\pi_{\theta_{\mathrm{old}}}$, a KL penalty, and division by trajectory length. After marginalizing the importance ratio, dropping the KL penalty (small β), and removing length normalization, the expected objective at node u reduces to $\sum_k P^k_{\theta} \cdot P^k_{\mathrm{succ}}$ up to constants independent of θ (see Appendix C for the full derivation). The empirical per-sample gradient at node u is:

$$\left.\frac{\partial \hat{L}_{\mathrm{DrGRPO}}}{\partial z_j}\right|_u = \frac{1}{N_u} \sum_{i:\, o_i \ni u} A_i \left(\delta_{r_i,j} - P^j_{\theta} \right), \tag{2.4}$$

where $A_i = (R_i - \bar{R})/\mathrm{std}(R)$ is the normalized advantage, r_i is the token chosen by rollout i, and N_u is the number of rollouts through u. In expectation, this is proportional to the ideal gradient (Equation 2.3):

$$\mathbb{E}\left[\nabla_{z_j} L_{\mathrm{DrGRPO}} \big|_u \right] \;\propto\; P^j_{\theta}(P^j_{\mathrm{succ}} - \bar{P}_{\mathrm{succ}}). \tag{2.5}$$

This connection motivates using Equation equation 2.3 as our oracle reference: it is what reward-based training would converge toward at each node, given sufficient rollouts. Standard GRPO's $1/|o_i|$ factor couples the advantage to trajectory length, preventing this clean per-node decomposition.

In practice, we compute the ideal gradient *directly* from empirical $\hat{P}^k_{\text{succ}}$ estimates at each node, not from per-sample advantage terms. This has lower variance than the finite-sample Dr. GRPO estimator and avoids the noise of individual trajectory rewards. The node's expected success is computed as $\bar{P}_{\text{succ}} = \sum_k P^k_\theta \hat{P}^k_{\text{succ}}$, weighting children by the student's own probabilities rather than by visit counts; pooling raw rollouts at the node would be biased, because targeted rollouts change how often each child is visited.

2.3 Distillation gradients

We now derive the gradient that each distillation algorithm produces at node u.

GKD (Generalized Knowledge Distillation). GKD (Agarwal et al., 2024), in its reverse-KL instance, minimizes the reverse KL divergence between student and teacher at each node:

$$L_{\text{GKD}}(u) = \text{KL}(\pi_\theta \| \pi_{\text{te}}) = \sum_k P^k_\theta \left(\log P^k_\theta - \log P^k_{\text{te}}\right) \tag{2.6}$$

Defining the per-token log-ratio $\ell_k = \log P^k_\theta - \log P^k_{\text{te}}$, the gradient is:

$$\begin{aligned} \frac{\partial L_{\text{GKD}}}{\partial z_j} &= \sum_k \frac{\partial}{\partial z_j}\left[P^k_\theta \, \ell_k\right] \\ &= \sum_k \ell_k \cdot P^k_\theta(\delta_{kj} - P^j_\theta) + \sum_k P^k_\theta \cdot \frac{\partial \ell_k}{\partial z_j} \end{aligned} \tag{2.7}$$

The second sum contributes $\sum_k P^k_\theta \cdot (\delta_{kj} - P^j_\theta) = P^j_\theta - P^j_\theta = 0$ (the softmax Jacobian sums to zero), so:

$$\frac{\partial L_{\text{GKD}}}{\partial z_j} = P^j_\theta(\ell_j - \bar{\ell}) \tag{2.8}$$

where $\bar{\ell} = \sum_k P^k_\theta \ell_k = \text{KL}(\pi_\theta \| \pi_{\text{te}})$. Since we minimize this KL, the distillation gradient (with sign flip) is $-P^j_\theta(\ell_j - \bar{\ell})$, which pushes logits toward tokens where the teacher assigns relatively higher probability.

Other distillation objectives. The single-sample estimator of Lu & Lab (2025) recovers the GKD gradient in expectation with opposite sign (the -1 baseline vanishes), and the REINFORCE-style gradient of MiniLLM (Gu et al., 2024) takes the same local form $P^j_\theta(f_j - \bar{f})$, but with f_k depending on the full future trajectory rather than being purely local. Full derivations are in Appendix C.

Summary. All four methods produce per-node gradients of the form:

$$\frac{\partial L}{\partial z_j} = P^j_\theta(f_j - \bar{f}), \qquad \bar{f} = \sum_k P^k_\theta f_k \tag{2.9}$$

with $f_k = P^k_{\text{succ}}$ for Dr. GRPO, $f_k = \pm(\log P^k_\theta - \log P^k_{\text{te}})$ for GKD (and its single-sample estimator), and a trajectory-dependent reward-to-go for MiniLLM. Because they share this structure, we can compare their *directions* via cosine similarity. A consequence of the shared P^j_θ factor is that the gradient magnitude for any token is gated by the student's current probability: even if the teacher identifies a high-success token, the update is small when P^j_θ is small. The teacher can amplify tokens the student already partially believes in, but has limited ability to inject entirely new continuations in a single step.

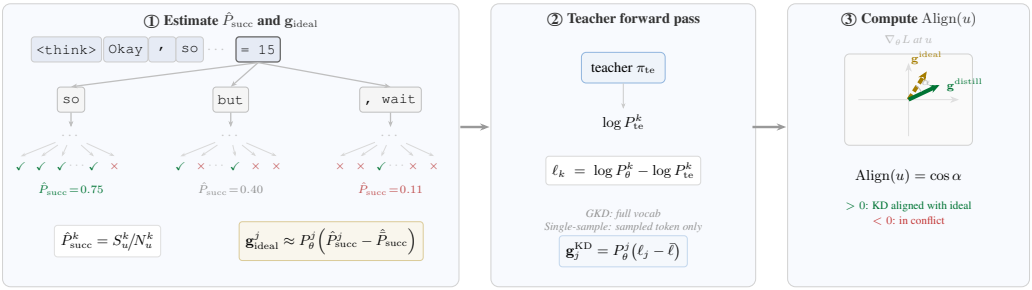


Figure 2: Computing the gradient alignment score at a branching node u. (1) Student rollouts yield empirical $\hat{P}^k_{\text{succ}}$ per branch and the ideal gradient. (2) A teacher forward pass produces the distillation gradient. (3) Their cosine similarity measures whether the teacher pushes toward success (> 0) or against it (< 0).

2.4 The Gradient Alignment Score

We define the **gradient alignment score** at node u as the cosine similarity between the ideal gradient (Equation 2.3) and the distillation gradient (e.g., GKD):

$$\text{Align}(u) = \cos\big(\mathbf{g}^{\text{ideal}}_u, \mathbf{g}^{\text{D}}_u\big) = \frac{\sum_{j \in \mathcal{S}_u} g^{\text{ideal}}_j \, g^{\text{D}}_j}{\|\mathbf{g}^{\text{ideal}}\|_{\mathcal{S}_u} \, \|\mathbf{g}^{\text{D}}\|_{\mathcal{S}_u}} \tag{2.10}$$

where $\mathbf{g}^{\text{ideal}}_u$ is the ideal gradient computed directly from empirical $\hat{P}^k_{\text{succ}}$ and $\mathbf{g}^{\text{D}}_u$ is the distillation gradient vector, both restricted to $\mathcal{S}_u$ (the set of tokens with sufficient visit counts at node u). The restriction is necessary because P^k_{succ} is only reliably estimated for tokens that have been sampled enough times. Both gradients are closed-form in logit space at a single token position, so computing $\text{Align}(u)$ never materializes a parameter-space gradient: given one teacher forward pass, its cost is $O(|\mathcal{S}_u|)$ per node, independent of model size.

The score ranges from -1 to $+1$: positive values mean the teacher and algorithm push the student toward tokens that lead to success at this node, negative values mean they push toward tokens that lead to failure, and values near zero mean the guidance is nearly orthogonal to the reward signal, as we would expect for purely stylistic disagreements.

We aggregate node scores by their mean along each path, and also report a success-rate-weighted (SR-weighted) cosine $\sum_u w_u \text{Align}(u) / \sum_u w_u$, where w_u is the range of $\hat{P}^k_{\text{succ}}$ across the children in $\mathcal{S}_u$, so that nodes where the outcome depends more on the choice count more. Mean cosine is our primary metric. Because $\mathcal{S}_u$ typically contains only two or three tokens, individual node scores often lie near ± 1, so a large per-node spread is expected by construction; our conclusions rest on averages over many nodes and questions, which we report with confidence intervals.

2.5 Computing the Score at Scale

The alignment score (Equation 2.10) requires reliable estimates of P^k_{succ} at each branching node (Figure 2 summarizes the three-step computation). Naïvely, this would require thousands of rollouts through every possible next token at every node, clearly infeasible for sequences of hundreds of tokens with vocabularies of 150K.

The core challenge is sparsity: given G initial rollouts, most tokens at most nodes receive zero visits, and deep branching points may have only 1–2 rollouts passing through them. To address this, we generate *targeted rollouts*: given a node u and a token k that needs more visits, we construct a prefix (prompt + path to u + token k) and sample completions from the student to the end of the response. Each targeted rollout enriches not only the target node but all ancestors and descendants along its path, so a single rollout launched at depth d simultaneously improves P_{succ} estimates at every node it passes through. This cascading effect means the total budget required grows sublinearly with sequence length.

Exponential depth windows. Rather than allocating rollouts uniformly, we partition the generation into exponentially growing depth windows (e.g., tokens 1 to 50, 51 to 150, 151 to 350, ...),

sampling early windows densely, where branching is frequent, and later windows more sparsely. Within each window, a fixed budget of tokens is ranked by how strongly the teacher disagrees with the student (Appendix D).

Budget and scalability. We target tokens where $P_\theta^k > \tau$ or $P_{\text{te}}^k > \tau$ ($\tau = 0.02$), i.e., those that contribute meaningfully to the gradient, and enrich each to $N_{\min} = 100$ visits. Nodes with $\geq N_{\text{sig}} = 20$ visits per child are retained for the alignment computation; for longer traces (AIME) where estimates are noisier, we use $N_{\text{sig}} = 40$. Raising N_{sig} to 40 or 60 on BoolQ and MMLU did not meaningfully change our conclusions, so we use 20 to retain more nodes. The total compute scales with the user-chosen budget (number of windows $\times$ per-window budget) rather than with sequence length, making the method applicable to traces ranging from $\sim$200 tokens (BoolQ) to $\sim$30K tokens (AIME) without modification. In practice, each question requires $\sim$45K–200K targeted rollouts depending on trace length.

Teacher-independent tree sharing. The generation tree and $\hat{P}_{\text{succ}}^k$ estimates depend only on the student's rollouts and outcomes, so we share one enriched tree across all 8 teacher configurations; each teacher then needs only one additional forward pass, which reduces total compute by $\sim 7\times$ compared to independent runs (Appendix D).

3 Experimental Setup

Student models. We evaluate two student scales from the Qwen3 family (Yang et al., 2025): Qwen3-0.6B and Qwen3-1.7B.

Teacher configurations. For each student, we evaluate 8 teacher configurations spanning two families:

Self-distillation (teacher = same model with enriched context): SELF-1C (1 correct solution in context), SELF-SUM-1C (correct solution summarized by Qwen3-32B), SELF-1C1W (1 correct + 1 wrong solution), SELF-SUM-1C1W (both summarized), SELF-1C (32B) (correct solution generated by Qwen3-32B shown to student-as-teacher).

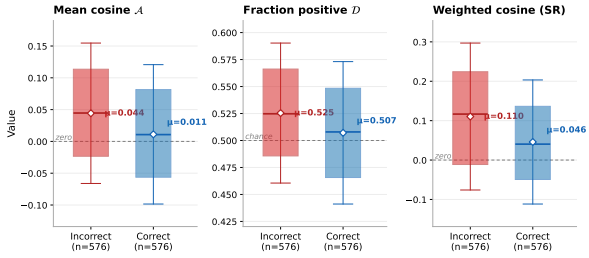
External teachers (larger models, same prompt as student): Qwen3-4B, Qwen3-8B, Qwen3-14B.

Datasets. We evaluate on two benchmarks: **BoolQ** (Clark et al., 2019), a reading comprehension task with True/False answers and short reasoning chains; and **MMLU** (Hendrycks et al., 2020), a multiple-choice knowledge benchmark with medium-length chains. We additionally present case studies on AIME 2025 ($\sim$5K–30K token traces) in Section 4.4. Rollout budgets and total compute are given in Appendix D.

Metrics. At each branching node with $\geq$2 children having $\geq$20 visits and nonzero success-rate range, we compute: gradient alignment (ideal vs. GKD cosine), teacher advantage, and success rate statistics. We aggregate per path (mean cosine along the path), per question (correct/incorrect split), and per teacher (means with 95% CIs across questions). p-values for correct versus incorrect paths come from two-sided Welch t-tests on per-path means, where each question contributes up to four representative paths (two correct, two incorrect), pooled across teacher configurations.

4 Results

We present results across two datasets and two model scales, totaling $\sim$88K decision points for BoolQ (0.6B) and $\sim$81K for BoolQ (1.7B), with MMLU providing $\sim$49K (0.6B) and $\sim$46K (1.7B). Overall, gradient alignment is weakly positive (mean cosine $+0.027$ for 0.6B, $+0.026$ for 1.7B on BoolQ), with a large per-node spread (std 0.83 to 0.91; see Appendix F.1) that is expected for cosines over two or three tokens (Section 2.4). Cheap single-pass features such as the divergence between teacher and student correlate only weakly with alignment (Appendix F.7), so the score cannot be replaced by a forward-pass proxy.



	Correct	Incorrect	p-value
Qwen3-0.6B, BoolQ			
Mean cosine	.011	.044	7×10^{-8}
Weighted (SR)	.046	.110	8×10^{-10}
Qwen3-1.7B, BoolQ			
Mean cosine	.001	.058	2×10^{-9}
Weighted (SR)	.010	.093	7×10^{-9}
Qwen3-0.6B, MMLU			
Mean cosine	.009	.048	.0001
Weighted (SR)	.021	.118	$<10^{-11}$
Qwen3-1.7B, MMLU			
Mean cosine	.012	.028	.123
Weighted (SR)	.034	.098	$<10^{-5}$

Figure 3: Gradient alignment on correct vs. incorrect paths. **Left:** distribution for Qwen3-0.6B on BoolQ; the teacher's gradient is more aligned on paths leading to wrong answers. **Right:** the pattern holds across all settings under both mean cosine and SR-weighted cosine. Full results in Appendix Table 10.

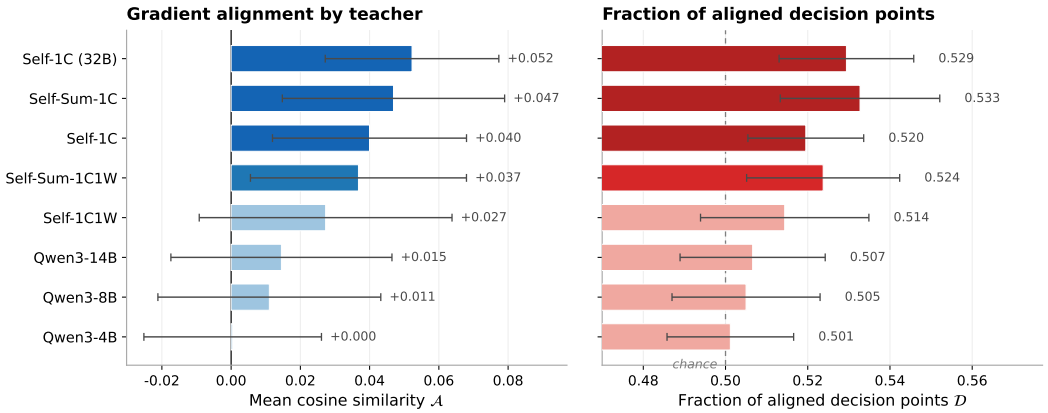


Figure 4: Teacher ranking by gradient alignment (Qwen3-0.6B, MMLU). Self-distillation with correct demonstrations dominates. Additional plots for BoolQ and 1.7B in Figure 11.

4.1 Distillation helps more on incorrect paths

Across all settings, **incorrect paths show higher alignment than correct paths** (Figure 3), significantly so in every setting under the SR-weighted cosine. On incorrect paths, the reward gradient points away from the current (failing) trajectory, and the teacher (which generally prefers tokens leading to success) pushes in the same direction. On correct paths, the student already favors successful tokens, so the teacher's disagreements are more often stylistic. Since the score is a cosine, this gap does not come from smaller ideal gradients on correct paths. The effect is strongest for 1.7B on BoolQ ($\Delta = -0.056$, $p < 10^{-9}$); even on MMLU where the mean cosine gap is not significant ($p = 0.12$), the weighted cosine is highly significant ($p < 10^{-5}$).

4.2 The best teacher depends on student capacity

Teacher rankings change with model scale (Figure 4, Table 1). For the **0.6B student**, self-distillation with correct-only demonstrations (Self-1C, Self-Sum-1C, Self-1C (32B)) aligns better than every external teacher on both datasets, whereas for the **1.7B student on BoolQ**, Qwen3-8B has the highest mean alignment, on par with Self-Sum-1C.

One hypothesis consistent with this pattern is **context comprehensibility**: a small student may not absorb the distribution of a much larger model whose reasoning differs from its own, but can use a correct solution in its own style, while for 1.7B the gap to an 8B teacher is small enough for the teacher's additional knowledge to become usable.

Table 1: Teacher ranking by mean gradient alignment (with 95% CI) across datasets. Self-distillation dominates for 0.6B; external teachers become competitive for 1.7B.

	Qwen3-0.6B		Qwen3-1.7B	
Teacher	BoolQ	MMLU	BoolQ	MMLU
SELF-1C	$\mathbf{.047 \pm .021}$	$.040 \pm .028$	$.028 \pm .024$	$.010 \pm .024$
SELF-1C (32B)	$.047 \pm .022$	$\mathbf{.052 \pm .025}$	$.020 \pm .034$	$.021 \pm .030$
SELF-SUM-1C	$.041 \pm .025$	$.047 \pm .032$	$\mathbf{.050 \pm .030}$	$\mathbf{.036 \pm .025}$
SELF-SUM-1C1W	$.016 \pm .025$	$.037 \pm .031$	$.008 \pm .029$	$.034 \pm .033$
SELF-1C1W	$.019 \pm .018$	$.027 \pm .037$	$.002 \pm .027$	$.009 \pm .022$
Qwen3-14B	$.016 \pm .021$	$.015 \pm .032$	$.036 \pm .035$	$.017 \pm .035$
Qwen3-8B	$.018 \pm .022$	$.011 \pm .032$	$\mathbf{.053 \pm .028}$	$.014 \pm .037$
Qwen3-4B	$.018 \pm .024$	$.000 \pm .026$	$.040 \pm .024$	$.017 \pm .041$

Table 2: Alignment score versus distillation training gain with external teachers on BoolQ. Δ is the accuracy gain over the untrained student (baseline .750 for 0.6B, .810 for 1.7B); Δ_{mean125} averages evaluations from step 125 onward and Δ_{final} is the last checkpoint. Bold marks the best teacher per column within each student.

			lr 1e-5		lr 1e-6	
Student	**Teacher**	**Align**	Δ_{mean125}	Δ_{final}	Δ_{mean125}	Δ_{final}
Qwen3-0.6B	Qwen3-4B	**.018**	**+.072**	+.080	**+.034**	**+.041**
	Qwen3-8B	**.018**	+.061	**+.089**	+.024	+.034
	Qwen3-14B	.016	+.059	+.066	+.023	+.033
Qwen3-1.7B	Qwen3-4B	.040	+.035	+.033	**+.027**	**+.031**
	Qwen3-8B	**.053**	**+.042**	**+.038**	+.024	+.019
	Qwen3-14B	.036	+.035	+.026	+.019	+.021

Adding a wrong demonstration **lowers alignment**: each 1C1W variant scores below its 1C counterpart (marginally for 1.7B on MMLU). Per-teacher breakdowns of the correct and incorrect gap are in Appendix F.8.

Summarization helps larger students. SELF-SUM-1C substantially raises alignment for 1.7B but has mixed effects for 0.6B, and SELF-1C (32B) works well mainly for 0.6B. Rankings are largely robust to metric choice (Appendix F.4); the exception is 1.7B on BoolQ, where the SR-weighted cosine places SELF-SUM-1C above Qwen3-8B. We treat mean cosine as primary.

4.3 DOES THE SCORE PREDICT TRAINING OUTCOMES?

We distill each student from the three external teachers with on-policy GKD (reverse KL) and full fine-tuning for 250 steps at learning rates 10^{-5} and 10^{-6}, evaluating held-out accuracy every 25 steps; only the teacher varies within a setting. On BoolQ (Table 2), at 10^{-5}, which trains better for both students, the ranking largely agrees with the score: the two teachers tied at the top for 0.6B both beat 14B, and for 1.7B the top-scored teacher (8B) gains the most, with the full ordering matching on final accuracy. At 10^{-6} the 1.7B ranking flips, and the learning-rate effect is two to three times the spread between teachers, so a single run mixes teacher quality with hyperparameters, whereas the score has no training hyperparameters. On MMLU every teacher lowers accuracy, but for 1.7B the lowest-scored teacher is last at both learning rates; on math (OpenThoughts, evaluated on AIME 2025), the 1.7B ordering matches the score exactly (Appendix E). Preliminary self-distillation runs show a similar trend (Appendix E.1).

4.4 CASE STUDIES: MATHEMATICAL REASONING (AIME 2025)

On AIME 2025 case-study questions with much longer reasoning traces (Appendix B), incorrect paths again show higher alignment on every analyzed question. Teacher choice differs from BoolQ and MMLU: SELF-1C1W ranks first on some questions, and summarized contexts lose to raw

demonstrations on the most difficult 0.6B question; we report these as case-study observations. This supports per-task diagnostics over a fixed recipe.

5 Related Work

Knowledge distillation (Hinton et al., 2015) trains a student on teacher soft distributions; sequence-level variants (Kim & Rush, 2016) generate teacher outputs for training but suffer from exposure bias (Bengio et al., 2015). On-policy distillation (OPD) addresses this by supervising the student on its own rollouts: MiniLLM (Gu et al., 2024) uses reverse KL via policy gradient, GKD (Agarwal et al., 2024) interpolates between on- and off-policy data, and Yang et al. (2026) show the teacher's log-ratio acts as an implicit dense reward. OPD is now standard in industry pipelines (Yang et al., 2025; Xiao et al., 2026; Zeng et al., 2026), and has been extended to self-distillation settings where a single model serves as its own teacher under privileged conditioning (Snell et al., 2022; Hübotter et al., 2026; Zhao et al., 2026; Ye et al., 2026; Shenfeld et al., 2026; Penaloza et al., 2026). Our work does not propose a new distillation algorithm but provides a diagnostic that measures, at each token, whether the teacher's signal agrees with the reward objective.

Despite OPD's growing adoption, when and why it fails remains poorly understood. Two concurrent works investigate this question from complementary angles. Li et al. (2026) show that OPD requires thinking-pattern consistency and genuinely new knowledge from the teacher, with success driven by progressive alignment on high-probability overlap tokens (97–99% of mass); they also reveal that reward quality degrades with trajectory depth. Kim et al. (2026) trace self-distillation degradation to the suppression of epistemic verbalization (the model's expression of uncertainty), showing that richer conditioning contexts suppress uncertainty tokens and harm OOD generalization when task coverage is broad. More broadly, distillation can hurt with overly capable teachers (Cho & Hariharan, 2019; Mirzadeh et al., 2020; Busbridge et al., 2025), and small models struggle to learn from strong reasoners (Li et al., 2025). Our gradient alignment score provides a mechanistic explanation for these phenomena: it directly quantifies, at each token position, whether the teacher's signal is exploitable by the student, regardless of the teacher's aggregate performance.

On the reward side, GRPO (Shao et al., 2024; Guo et al., 2025; Liu et al., 2025) and DAPO (Yu et al., 2026) train reasoning models with sparse outcome rewards, while process reward models (Lightman et al., 2024; Uesato et al., 2022) provide step-level feedback but require separate annotation. Our gradient decomposition unifies these perspectives by showing that reward and distillation objectives share the same local structure ($P_\theta^j(f_j - \bar{f})$), enabling direct offline comparison at token granularity without training or additional models.

6 Conclusion

We set out to answer a simple question: at each token in a reasoning chain, does the teacher's distillation signal actually point toward correct answers? We derived an ideal per-node gradient from empirical success probabilities, showed that major distillation objectives share the same local structure, and built a scalable pipeline to compute gradient alignment offline for long sequences.

Our experiments show that distillation guidance is most aligned on failing rollouts, that the best teacher configuration depends on student capacity and task, and that for external teachers the score's ranking largely agrees with training under the better-performing training configuration, whereas the ranking from an individual run can shift with training hyperparameters such as the learning rate. In practice, the score can be used to check that a teacher carries useful signal before spending a training run, computed per domain, and used to weight distillation toward failing rollouts. A natural next step is to combine reinforcement learning with on-policy distillation, using the score to apply the teacher's signal where it agrees with the reward.

AI use statement

In this work, we used generative AI tools to assist with editing and polishing the text of the paper, with writing and debugging parts of the experiment and analysis code, and with LaTeX formatting of tables and figures. We have not used generative AI tools to generate research ideas, derivations,

or experimental results; the proposed method, the gradient derivations, the experimental design, and the analysis of results are the authors' own. We did not use generative AI tools for literature search or for generating bibliography entries, and every reference was verified manually by the authors against the original publication. All AI-edited text was read and revised by the authors, and all AI-assisted code was reviewed and tested by the authors before being used to produce results reported in this paper. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

REFERENCES

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

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

Dan Busbridge, Amitis Shidani, Floris Weers, Jason Ramapuram, Etai Littwin, and Russ Webb. Distillation scaling laws. *arXiv preprint arXiv:2502.08606*, 2025.

Jang Hyun Cho and Bharath Hariharan. On the efficacy of knowledge distillation. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 4793–4801. IEEE, 2019.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 2924–2936, 2019.

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

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

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

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

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

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

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

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

Yuetai Li, Xiang Yue, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Bill Yuchen Lin, Bhaskar Ramasubramanian, and Radha Poovendran. Small models struggle to learn from strong reasoners. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 25366–25394, 2025.

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

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

Kevin Lu and Thinking Machines Lab. On-policy distillation. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20251026. https://thinkingmachines.ai/blog/on-policy-distillation.

Seyed Iman Mirzadeh, Mehrdad Farajtabar, Ang Li, Nir Levine, Akihiro Matsukawa, and Hassan Ghasemzadeh. Improved knowledge distillation via teacher assistant. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 5191–5198, 2020.

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.

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

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

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

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.

Bangjun Xiao, Bingquan Xia, Bo Yang, Bofei Gao, Bowen Shen, Chen Zhang, Chenhong He, Chiheng Lou, Fuli Luo, Gang Wang, et al. Mimo-v2-flash technical report. *arXiv preprint arXiv:2601.02780*, 2026.

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

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

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

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

Aohan Zeng, Xin Lv, Zhenyu Hou, Zhengxiao Du, Qinkai Zheng, Bin Chen, Da Yin, Chendi Ge, Chenghua Huang, Chengxing Xie, et al. Glm-5: from vibe coding to agentic engineering. *arXiv preprint arXiv:2602.15763*, 2026.

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

Table 3: Notation used throughout the paper.

Symbol	Meaning
q, G	Prompt; number of initial student rollouts
π_θ, π_{te}	Student policy; teacher policy
u	Node in the generation tree (a prompt plus a partial response)
j, k	Token indices in the vocabulary
z_j	Student logit for token j at node u
P_θ^k, P_{te}^k	Student and teacher probability of token k at node u
δ_{kj}	Kronecker delta (1 if $k = j$, 0 otherwise)
N_u^k, S_u^k	Number of rollouts through u that choose token k; number of those that reach a correct answer
P_{succ}^k, $\hat{P}_{\text{succ}}^k$	True probability of success after choosing token k at u; its estimate S_u^k/N_u^k
$\bar{P}_{\text{succ}}$	$\sum_k P_\theta^k \hat{P}_{\text{succ}}^k$, the student's expected success at u
$\mathcal{S}_u$	Tokens at u with at least N_{sig} visits
N_{sig}, $N_{\min}$	Significance threshold (20; 40 for AIME); target visits per enriched token (100)
ℓ_k, $\bar{\ell}$	Log ratio $\log P_\theta^k - \log P_{\text{te}}^k$; its student-weighted mean $\text{KL}(\pi_\theta \| \pi_{\text{te}})$
f_k, $\bar{f}$	Per-token signal in the shared gradient form $P_\theta^j(f_j - \bar{f})$; its student-weighted mean
$\mathbf{g}_u^{\text{ideal}}$, $\mathbf{g}_u^{\text{D}}$	Ideal and distillation gradient vectors over $\mathcal{S}_u$
$\text{Align}(u)$, $\text{Adv}(u)$	Gradient alignment score; teacher advantage

A Notation

B Full AIME 2025 Case Studies

B.1 Overview

To test whether our findings extend to mathematical reasoning (where thinking traces are substantially longer, ∼5K–30K tokens, and reasoning steps more complex), we analyze AIME 2025 questions on which the student's success rate is neither near zero nor near one, so that the generation tree contains informative branching. Each question uses the same 8 teacher configurations as BoolQ/MMLU. We use sig $= 40$ (requiring $\geq$40 visits per child for significance) given the longer chains and sparser branching.

Teacher rankings on AIME. Figure 7 shows teacher alignment on two representative questions. Both external and self-distillation teachers achieve positive alignment across all analyzed questions. Self-Sum-1C (summarized correct demonstration) is the strongest teacher on Q0 ($+0.043$) and Q7 ($+0.055$), while the raw demo variants lead on Q3 (Self-1C1W: $+0.042$, Self-1C: $+0.040$) and Q28 (Self-1C1W: $+0.053$). Notably, Self-1C1W (which includes a wrong demonstration) is the best teacher on the harder questions (Q3 and Q28), contrasting sharply with BoolQ/MMLU where it consistently underperforms.

Correct vs. incorrect on AIME. Among teachers with positive alignment, the incorrect $>$ correct pattern holds on every analyzed question when focusing on high-stakes decision points. On Q0, all positive teachers show higher alignment on incorrect paths (best teacher Self-Sum-1C: incorrect $+0.097$ vs. correct -0.011, $\Delta = -0.108$). On Q3, the top-ranked teachers show the same pattern (Self-1C1W: incorrect $+0.056$ vs. correct $+0.027$, $\Delta = -0.029$). On Q28, 7 of 8 positive teachers show incorrect $>$ correct (Self-1C1W: incorrect $+0.097$ vs. correct $+0.008$, $\Delta = -0.089$). On Q7, the pattern emerges clearly at consequential nodes (best teacher Self-Sum-1C: incorrect $+0.093$ vs. correct $+0.017$, $\Delta = -0.076$ at high success-rate-range nodes), though the gap is smaller at low-stakes nodes where the signal is noisier. This confirms that even on long mathematical reasoning chains, the teacher's gradient is most useful on the student's failing rollouts.

What differs from BoolQ/MMLU. While the correct/incorrect finding transfers, teacher-choice conclusions do not, further supporting the comprehensibility hypothesis. On hard AIME math (Q3),

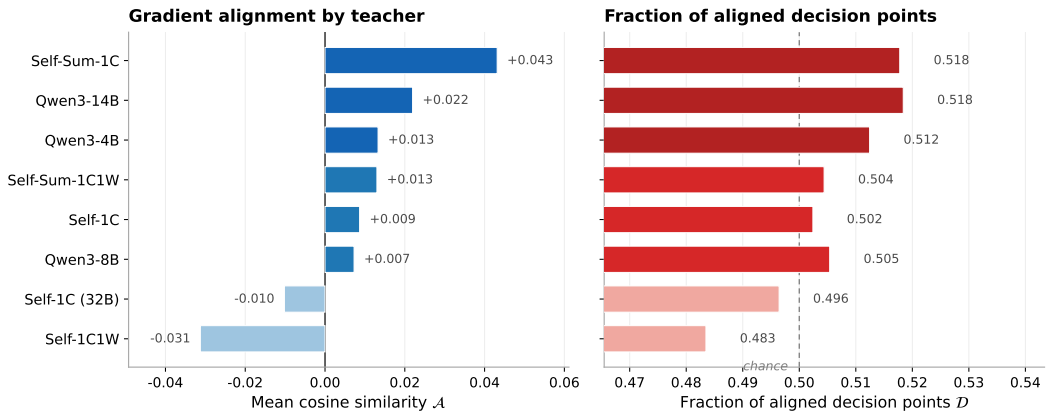


Figure 5: AIME Q0, Qwen3-0.6B

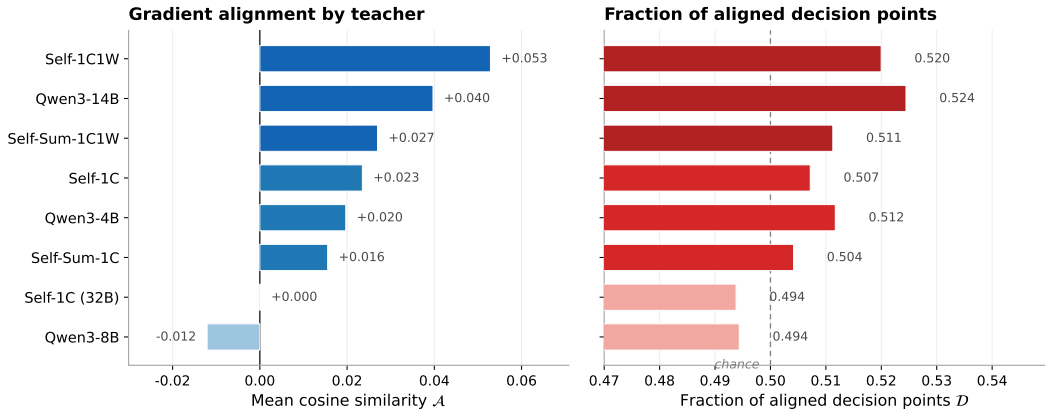


Figure 6: AIME Q28, Qwen3-1.7B

Figure 7: Teacher ranking by gradient alignment on two AIME 2025 questions. Both external teachers and self-distillation variants achieve positive alignment, with SELF-SUM-1C and SELF-1C1W leading.

SELF-1C1W (which includes a wrong demonstration alongside the correct one) is the *best* teacher (+0.042), directly contradicting BoolQ/MMLU where wrong demonstrations consistently hurt. On hard math, seeing a common mistake may provide useful contrastive signal ("avoid this error") that the student can comprehend and act on, whereas on simpler tasks the wrong solution is merely distracting noise. Meanwhile, SELF-SUM-1C (summarized) loses to raw SELF-1C on Q3: the 0.6B student cannot decipher a compressed summary of a hard mathematical argument and needs the full step-by-step trace. On easier/medium questions (Q0, Q7), summarized contexts perform well because the reasoning is simple enough to compress without losing comprehensibility.

Additionally, teacher rankings shift with filter stringency: on Q7, external teachers lead at lenient filters but SELF-SUM-1C dominates at strict filters that focus on high-stakes nodes. These observations reinforce that no universal distillation recipe exists: the optimal teacher configuration depends on the task, the question difficulty, the student's capacity to comprehend the context, and which decision points one prioritizes.

C Full Gradient Derivations

C.1 Softmax Jacobian

The student's transition probability at node u is $P_\theta^k = e^{z_k} / \sum_{k'} e^{z_{k'}}$, with Jacobian:

$$\frac{\partial P_\theta^k}{\partial z_j} = P_\theta^k(\delta_{kj} - P_\theta^j) \tag{C.1}$$

For any objective $L(u) = \sum_k P_\theta^k f_k$ where f_k does not depend on θ:

$$\frac{\partial L}{\partial z_j} = \sum_k f_k \cdot P_\theta^k(\delta_{kj} - P_\theta^j) = P_\theta^j f_j - P_\theta^j \sum_k P_\theta^k f_k = P_\theta^j(f_j - \bar{f}) \tag{C.2}$$

C.2 Dr. GRPO

The full GRPO objective in the limit $G \to \infty$ is:

$$J_{\text{GRPO}} = \mathbb{E}_{o_i \sim \pi_{\theta_{\text{old}}}} \left(\frac{1}{|o_i|} \sum_t \frac{\pi_\theta(o_{i,t} \mid q,\, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q,\, o_{i,<t})} \cdot A_i - \beta\, D_{KL}(\pi_\theta \| \pi_{\text{ref}}) \right) \tag{C.3}$$

We make the following simplifications to obtain a per-node decomposition:

1. Drop the KL penalty. Since β is typically small, we set $\beta = 0$.

2. Marginalize the importance ratio. For trajectories passing through node u and choosing token k, the importance weight $\pi_\theta(o_{i,t} \mid \cdot)/\pi_{\theta_{\text{old}}}(o_{i,t} \mid \cdot) = P_\theta^k / P_{\theta_{\text{old}}}^k$ at step t cancels with the sampling measure $P_{\theta_{\text{old}}}^k$ after marginalization, leaving a net factor of P_θ^k.

3. Remove length normalization (Dr. GRPO). Standard GRPO divides by $|o_i|$, coupling the advantage to trajectory length:

$$J(u) \propto \sum_k P_\theta^k \cdot \mathbb{E}\left[\frac{A_i}{|o_i|} \;\middle|\; o_i \text{ passes } u \to v_k \right] \tag{C.4}$$

Since A_i and $|o_i|$ are correlated (e.g., successful trajectories may be shorter), this expectation does not factor. Dr. GRPO removes the $1/|o_i|$ factor, giving:

$$J_{\text{DrGRPO}}(u) \propto \sum_k P_\theta^k \cdot \mathbb{E}[A_i \mid o_i \text{ passes } u \to v_k] \tag{C.5}$$

4. Evaluate the conditional expectation. With $A_i = (R_i - \bar{R})/\text{std}(R)$ and binary rewards $R_i \in \{0, 1\}$:

$$\mathbb{E}[A_i \mid o_i \text{ passes } u \to v_k] = \frac{P_{\text{succ}}^k - \bar{R}}{\text{std}(R)} = \alpha \cdot P_{\text{succ}}^k - C \tag{C.6}$$

where $\alpha = 1/\text{std}(R)$ and $C = \bar{R}/\text{std}(R)$ are batch-level constants (each trajectory shifts them by $\pm 1/G$, negligible for large G).

5. Remove the constant. Since $\sum_k P_\theta^k \cdot (-C) = -C$ is independent of θ, the gradient-relevant objective at node u reduces to:

$$L_{\text{DrGRPO}}(u) = \sum_k P_\theta^k \cdot P_{\text{succ}}^k \tag{C.7}$$

which is the ideal objective (Equation 2.2), giving gradient $P_\theta^j(P_{\text{succ}}^j - \bar{P}_{\text{succ}})$.

Empirical estimator. In practice with finite samples, the per-node gradient is estimated as:

$$\left.\frac{\partial \hat{L}_{\mathrm{DrGRPO}}}{\partial z_j}\right|_u = \frac{1}{N_u} \sum_{i:\, o_i \ni u} A_i \left(\delta_{r_i,j} - P_\theta^j\right) \tag{C.8}$$

where r_i is the token chosen by rollout i at node u and N_u is the number of rollouts passing through u. In our framework, we instead compute the ideal gradient directly from empirical $\hat{P}^k_{\mathrm{succ}}$ estimates, which is equivalent in expectation but lower variance.

C.3 GKD

The GKD loss is $L_{\mathrm{GKD}} = \mathrm{KL}(\pi_\theta \| \pi_{\mathrm{te}}) = \sum_k P_\theta^k (\log P_\theta^k - \log P_{\mathrm{te}}^k)$. Defining $\ell_k = \log P_\theta^k - \log P_{\mathrm{te}}^k$, the gradient is:

$$\frac{\partial L_{\mathrm{GKD}}}{\partial z_j} = P_\theta^j (\ell_j - \bar{\ell}) \quad \text{where} \quad \bar{\ell} = \mathrm{KL}(\pi_\theta \| \pi_{\mathrm{te}}) \tag{C.9}$$

The additional term from $\partial \log P_\theta^k / \partial z_j$ produces $+P_\theta^j(1-1) = 0$ after applying the Jacobian sum-to-zero property.

C.4 Single-sample GKD estimator

The single-sample estimator of Lu & Lab (2025) uses importance weighting at the sampled token only. For rollout i choosing token r_i at node u, the weight is:

$$w_i = \log P_{\mathrm{te}}^{r_i} - \log P_\theta^{r_i} - 1 \tag{C.10}$$

and the per-sample gradient contribution is:

$$\nabla_{z_j} \ell_i = w_i \cdot (\delta_{r_i,j} - P_\theta^j) \tag{C.11}$$

Taking the expectation over the student's sampling distribution:

$$\begin{aligned} \mathbb{E}[\nabla_{z_j} \ell] &= \sum_k P_\theta^k \left(\log P_{\mathrm{te}}^k - \log P_\theta^k - 1\right) (\delta_{kj} - P_\theta^j) \\ &= P_\theta^j (-\ell_j - 1) - P_\theta^j \sum_k P_\theta^k (-\ell_k - 1) \\ &= -P_\theta^j (\ell_j - \bar{\ell}) \end{aligned} \tag{C.12}$$

The constant -1 cancels because $\sum_k P_\theta^k (\delta_{kj} - P_\theta^j)(-1) = 0$. This recovers the GKD gradient (Equation 2.8) with opposite sign, confirming that the single-sample importance-weighted estimator and the full-vocabulary KL minimization produce the same gradient direction in expectation.

C.5 MiniLLM

MiniLLM (Gu et al., 2024) uses a REINFORCE-style gradient where the reward-to-go couples each node's gradient to all downstream nodes. At node u (step t, sampled token r_i), the per-sample gradient is:

$$\nabla_{z_j} \ell_i = -(\delta_{r_i,j} - P_\theta^j) \cdot \left(\sum_{t' \geq t} R_{t'} - 1 \right) \tag{C.13}$$

where $R_{t'} = \log P_{\mathrm{te}}^{o_{i,t'}} - \log P_\theta^{o_{i,t'}}$ is the per-step reward. Unlike GKD and its single-sample variant, here f_k in the unified form $P_\theta^j (f_j - \bar{f})$ depends on the full future trajectory rather than being purely local, making the per-node contribution path-dependent.

D COMPUTATION DETAILS

Rollout prioritization. Not all nodes are equally informative for targeted enrichment. We prioritize using two criteria: (i) GKD gradient magnitude $|P_\theta^j(\ell_j - \bar{\ell})|$, identifying tokens where the teacher disagrees most strongly; and (ii) probability difference $|P_{\text{te}}^j - P_\theta^j|$, identifying where the two distributions diverge most. We allocate a budget per depth window (smaller, denser windows early; larger, coarser windows later) and rank candidate tokens by both criteria. Within each window, we select a fixed number k of tokens with the largest GKD gradient magnitude and r tokens with the largest probability difference, prioritizing tokens where the teacher disagrees most strongly.

Multi-teacher pipeline. Phase 1 (shared): generate G initial rollouts, build the generation tree, and run targeted rollouts to enrich branching points. Each teacher's logits are computed to identify important tokens; rollouts accumulate in the shared tree so later teachers benefit from prior enrichment. Phase 2 (per-teacher): compute P_{succ}^k once from the enriched tree (teacher-independent), then for each teacher run one forward pass to obtain P_{te}^k and evaluate the alignment score.

Compute budget. Each question requires substantial compute: G=200 initial rollouts, 4 representative paths (2 correct, 2 incorrect), and ∼45K–200K targeted rollouts at undersampled branching points (totaling ∼72 A100-days for the full experiment suite).

E TRAINING VALIDATION DETAILS

All runs distill the student from one external teacher with on-policy GKD (reverse KL) and full fine-tuning for 250 steps, at learning rates 10^{-5} and 10^{-6}, evaluating held-out accuracy every 25 steps. Only the teacher varies within a setting. Δ is the accuracy gain over the shared untrained baseline. Δ_{mean}, Δ_{mean50}, and Δ_{mean125} average the evaluations from step 25, 50, and 125 onward; Δ_{max} is the best evaluation and Δ_{final} the last checkpoint. For math, students are trained on OpenThoughts and evaluated on AIME 2025, and alignment values are the per-question AIME scores of Appendix B.

Table 4: All training metrics on BoolQ (baseline .750 for Qwen3-0.6B, .810 for Qwen3-1.7B).

Student	lr	Teacher	Align	Δ_{mean}	Δ_{mean50}	Δ_{mean125}	Δ_{max}	Δ_{final}
0.6B	1e-5	4B	.018	+.063	+.066	+.072	+.087	+.080
		8B	.018	+.056	+.055	+.061	+.089	+.089
		14B	.016	+.051	+.054	+.059	+.066	+.066
	1e-6	4B	.018	+.029	+.031	+.034	+.041	+.041
		8B	.018	+.019	+.020	+.024	+.034	+.034
		14B	.016	+.017	+.020	+.023	+.033	+.033
1.7B	1e-5	4B	.040	+.030	+.031	+.035	+.046	+.033
		8B	.053	+.037	+.039	+.042	+.045	+.038
		14B	.036	+.031	+.032	+.035	+.054	+.026
	1e-6	4B	.040	+.026	+.028	+.027	+.031	+.031
		8B	.053	+.020	+.021	+.024	+.029	+.019
		14B	.036	+.019	+.020	+.019	+.026	+.021

E.1 SELF-DISTILLATION TRAINING RUNS (PRELIMINARY)

We also train each student with the five self-distillation contexts, using the on-policy self-distillation setup of Zhao et al. (2026) with LoRA, learning rate 5×10^{-6}, and the same schedule and evaluation protocol as the external-teacher runs. Table 7 reports pre-training alignment (Table 1) next to Δ_{mean125} for single-seed runs. The Spearman correlation between alignment and training gain ranges from 0.67 to 1.00 across settings, and contexts with only correct demonstrations train better than their counterparts with a wrong demonstration in every comparison. On MMLU, as with external teachers, every context lowers accuracy.

Table 5: Training on MMLU: Δ_{mean125} / Δ_{final} per teacher (baseline .514 for Qwen3-0.6B, .678 for Qwen3-1.7B). Distillation lowers accuracy for every teacher.

Student	lr	Qwen3-4B	Qwen3-8B	Qwen3-14B
0.6B alignment		.000	.011	.015
0.6B	1e-5	$-.030$ / $-.004$	$-.029$ / $-.034$	$-.049$ / $-.052$
0.6B	1e-6	$-.033$ / $-.036$	$-.046$ / $-.056$	$-.026$ / $-.028$
1.7B alignment		.017	.014	.017
1.7B	1e-5	$-.033$ / $-.020$	$-.039$ / $-.040$	$-.022$ / $-.014$
1.7B	1e-6	$-.017$ / $-.032$	$-.034$ / $-.038$	$-.022$ / $-.044$

Table 6: Training on OpenThoughts, evaluated on AIME 2025 (baseline .333 for Qwen3-1.7B, .183 for Qwen3-0.6B).

Student	Teacher	Align	Best acc.	$\Delta_{\max}$
1.7B	Qwen3-14B	$+.040$	.367	$+.034$
	Qwen3-4B	$+.020$	.333	$+.000$
	Qwen3-8B	$-.012$	.300	$-.033$
0.6B	Qwen3-14B	$+.022$	.200	$+.017$
	Qwen3-4B	$+.013$	.233	$+.050$
	Qwen3-8B	$+.007$	.183	$+.000$

F Additional Figures and Analysis

F.1 Alignment score distribution

The distribution of per-node gradient alignment scores across all decision points and 8 teachers on BoolQ spans $[-1, +1]$ with a slight positive mean ($+0.027$ for 0.6B, $+0.026$ for 1.7B) and a per-node standard deviation of 0.83 to 0.91, a spread expected for cosines over two or three tokens (Section 2.4).

F.2 Per-path cosine oscillation

Along individual reasoning paths, the gradient alignment score oscillates token-by-token between positive and negative values, confirming that the teacher's helpfulness is highly local: even on a single path, consecutive tokens can alternate between beneficial and harmful distillation signal.

F.3 Selective distillation (oracle analysis)

Given the high per-token heterogeneity, a natural question is: what if we could apply the teacher's gradient *only* at tokens where alignment is positive? Table 8 compares the mean importance-weighted signal under selective strategies that retain only decision points with alignment above a threshold t.

Simply filtering to tokens with positive alignment ($t = 0$) yields 10–15$\times$ improvement in mean signal using only $\sim$52% of tokens, and *every path* benefits. This is an oracle analysis; at training time, one does not know the true alignment. However, it establishes a strong upper bound and motivates using the divergence-based predictors identified in Appendix F.7 as approximate filters for alignment-aware training.

F.4 Teacher ranking by alternative metrics

Table 9 reports teacher rankings using fraction of positively aligned tokens and success-rate-range-weighted cosine. For 0.6B, both metrics preserve the same hierarchy as mean cosine (Table 1). For 1.7B, the weighted cosine reverses the top ranking: Self-Sum-1C leads over Qwen3-8B, as discussed in Section 4.2.

Table 7: Self-distillation training (single seed): alignment / Δ_{mean125} per context. Baselines are .750 and .810 on BoolQ and .514 and .678 on MMLU for Qwen3-0.6B and Qwen3-1.7B. The last row gives the Spearman correlation between alignment and training gain within each setting.

	BoolQ		MMLU	
Context	0.6B	1.7B	0.6B	1.7B
SELF-1C	.047 / +.045	.028 / +.032	.040 / −.015	.010 / −.031
SELF-1C (32B)	.047 / +.041	.020 / +.035	.052 / −.005	.021 / −.017
SELF-SUM-1C	.041 / +.054	.050 / +.037	.047 / −.029	.036 / −.001
SELF-1C1W	.019 / −.014	.002 / +.027	.027 / −.037	.009 / −.033
SELF-SUM-1C1W	.016 / −.023	.008 / +.011	.037 / −.045	.034 / −.012
Spearman ρ	0.67	0.80	0.80	1.00

Table 8: Selective distillation (oracle): mean signal (SR range $\times$ cosine), fraction of tokens retained, and fraction of paths that beat full GKD. Filtering to aligned tokens yields 10–15$\times$ signal improvement using $\sim$50% of tokens.

Setting	**Strategy**	**Mean signal**	**% tokens**	**% beats GKD**
0.6B BoolQ	Full GKD	0.007 ± 0.001	100%	—
	Selective (t=0)	0.070 ± 0.002	51.6%	100%
	Selective (t=0.3)	0.076 ± 0.002	46.6%	100%
1.7B BoolQ	Full GKD	0.006 ± 0.001	100%	—
	Selective (t=0)	0.093 ± 0.003	51.6%	100%
	Selective (t=0.3)	0.096 ± 0.003	49.7%	100%
0.6B MMLU	Full GKD	0.005 ± 0.001	100%	—
	Selective (t=0)	0.057 ± 0.002	51.7%	100%
	Selective (t=0.3)	0.062 ± 0.002	46.1%	100%
1.7B MMLU	Full GKD	0.006 ± 0.001	100%	—
	Selective (t=0)	0.077 ± 0.003	51.2%	100%
	Selective (t=0.3)	0.081 ± 0.003	47.9%	100%

The weighted cosine (SR range) amplifies the differences seen in mean cosine: self-distillation teachers for 0.6B achieve weighted cosine 0.108–0.120 vs. 0.053–0.062 for external baselines ($2\times$ ratio). For 1.7B, SELF-SUM-1C leads on weighted cosine (0.088) followed by Qwen3-8B (0.072), showing that at high-stakes decision points the summarized self-distillation context remains competitive even when mean cosine favors the external teacher. Fraction positive shows smaller but consistent differences in the same direction.

F.5 ADDITIONAL TEACHER RANKING PLOTS

F.6 FULL CORRECT VS. INCORRECT BREAKDOWN

Table 10 extends the main-text results (Figure 3) with teacher advantage. The pattern holds across all metrics, with the weighted cosine showing the largest effect sizes.

F.7 WITHIN-PATH CORRELATION DETAILS

To understand *where* alignment is positive within a reasoning chain, we compute within-path Spearman correlations between the alignment score and single-rollout features (Table 11). Teacher–student divergence (KL, JS, L_2) correlates positively with alignment, while distributional similarity correlates negatively: the useful signal lives where the teacher disagrees with the student. However, correlations are uniformly weak ($|\rho| \approx 0.02$–0.04): no single feature reliably predicts whether a disagreement is helpful or harmful, since that depends on downstream success probabilities not observable from a single forward pass. The consistent sign nonetheless suggests divergence could

Table 9: Teacher ranking by fraction positive and weighted cosine (SR range) on BoolQ.

	Frac. positive		Weighted cosine (SR)	
Teacher	0.6B	1.7B	0.6B	1.7B
SELF-1C	.526	.515	**.120**	.046
SELF-1C (32B)	**.528**	.512	.108	.039
SELF-SUM-1C	.522	.527	.114	**.088**
SELF-SUM-1C1W	.509	.505	.065	.038
SELF-1C1W	.511	.501	.043	.017
Qwen3-14B	.512	.518	.060	.053
Qwen3-8B	.511	**.530**	.062	.072
Qwen3-4B	.512	.520	.053	.060

Table 10: Full correct vs. incorrect path alignment across all metrics.

Setting	**Metric**	**Correct**	**Incorrect**	Δ	p-value
0.6B BoolQ	Mean cosine	0.011	0.044	-0.033	7×10^{-8}
	Weighted cosine (SR range)	0.046	0.110	-0.065	8×10^{-10}
	Teacher advantage	0.002	0.008	-0.006	$< 10^{-19}$
1.7B BoolQ	Mean cosine	0.001	0.058	-0.056	2×10^{-9}
	Weighted cosine (SR range)	0.010	0.093	-0.083	7×10^{-9}
	Teacher advantage	-0.002	0.011	-0.013	$< 10^{-33}$
0.6B MMLU	Mean cosine	0.009	0.048	-0.039	0.0001
	Weighted cosine (SR range)	0.021	0.118	-0.097	$< 10^{-11}$
	Teacher advantage	0.002	0.005	-0.004	$< 10^{-14}$
1.7B MMLU	Mean cosine	0.012	0.028	-0.016	0.123
	Weighted cosine (SR range)	0.034	0.098	-0.063	$< 10^{-5}$
	Teacher advantage	0.001	0.010	-0.009	$< 10^{-14}$

serve as a cheap necessary-condition filter for alignment-aware training. Alignment also trends slightly positive with depth (early tokens are templatic; later tokens involve actual reasoning steps).

Table 11: Within-path Spearman correlations between single-rollout features and the gradient alignment score. Top: averaged over all 8 teachers. Bottom: restricted to a teacher per setting.

	Qwen3-0.6B		Qwen3-1.7B	
Feature	BoolQ	MMLU	BoolQ	MMLU
All teachers combined				
Depth (normalized)	$+.042$	$+.035$	$-.007$	$+.010$
KL($\pi_\theta \Vert \pi_{\text{te}}$)	$+.029$	$+.028$	$+.009$	$+.030$
JS divergence	$+.030$	$+.028$	$+.006$	$+.022$
L_2 distance	$+.031$	$+.034$	$+.006$	$+.023$
Cosine($\pi_\theta, \pi_{\text{te}}$)	$-.029$	$-.026$	$-.003$	$-.020$
With a teacher only				
	SELF-1C	SELF-1C (32B)	SELF-SUM-1C	SELF-SUM-1C
Depth (normalized)	$+.042$	$+.026$	$+.000$	$+.028$
KL($\pi_\theta \Vert \pi_{\text{te}}$)	$+.044$	$+.000$	$+.011$	$+.028$
JS divergence	$+.041$	$-.002$	$+.009$	$+.020$
L_2 distance	$+.042$	$+.013$	$+.009$	$+.020$
Cosine($\pi_\theta, \pi_{\text{te}}$)	$-.034$	$-.005$	$-.005$	$-.019$

The divergence→alignment pattern is consistent for 0.6B and strengthens when restricting to the best teacher, but weakens for 1.7B BoolQ regardless of teacher choice. The positive sign means alignment is higher where the teacher disagrees with the student: low-divergence tokens offer little useful signal. However, high divergence is necessary but not sufficient: many high-divergence

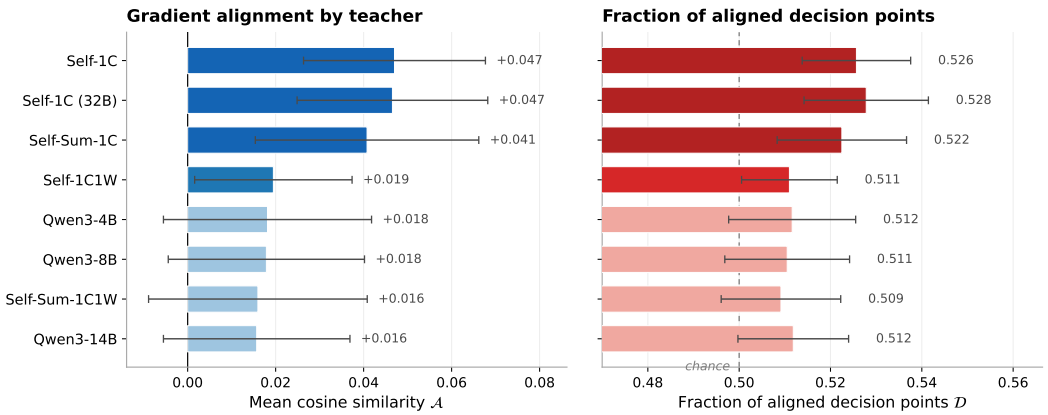


Figure 8: Qwen3-0.6B, BoolQ

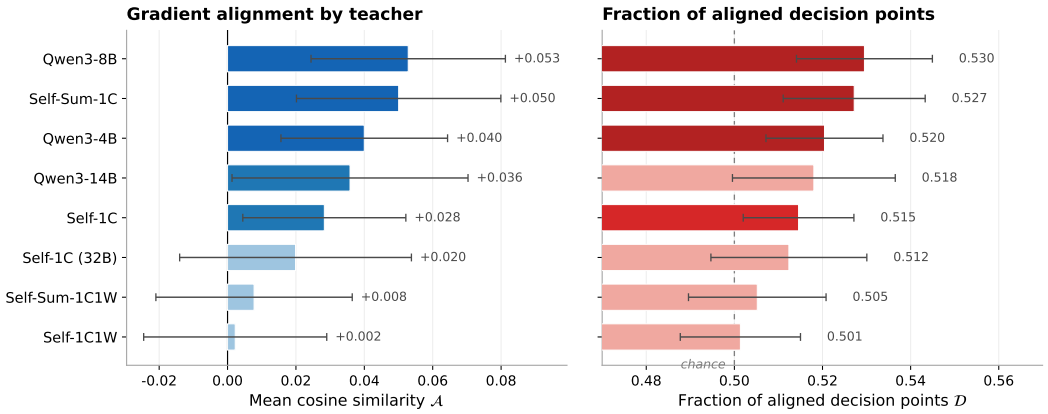


Figure 9: Qwen3-1.7B, BoolQ

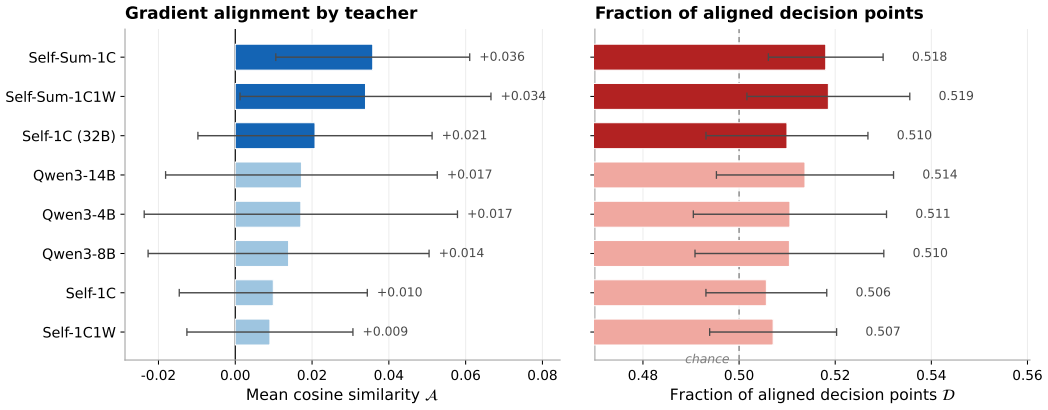


Figure 10: Qwen3-1.7B, MMLU

Figure 11: Teacher ranking by gradient alignment (additional settings). For 1.7B on BoolQ, external teachers (Qwen3-8B) achieve the highest alignment. On MMLU, self-distillation retains an edge for both model scales.

tokens still have negative alignment, since the teacher's confidence can point toward failure as easily as success. Depth correlates positively for 0.6B ($\rho \approx +0.04$), reflecting that early tokens are templatic while later tokens involve reasoning where the teacher's contextual advantage emerges;

this effect vanishes for 1.7B. On AIME (longer traces), the divergence pattern persists but depth becomes weakly negative ($\rho \approx -0.03$ on hard questions), suggesting that on complex math the teacher's advantage does not grow with depth.

F.8 PER-TEACHER CORRECT VS. INCORRECT BREAKDOWN

Table 12 breaks down the correct-vs-incorrect gap by teacher on BoolQ. For 0.6B, SELF-1C is unique: it achieves nearly equal alignment on both correct and incorrect paths ($\Delta \approx 0$), providing a uniformly helpful gradient regardless of path outcome. All other teachers show the typical pattern of higher alignment on incorrect paths; for 1.7B this holds without exception.

Table 12: Per-teacher correct vs. incorrect mean cosine on BoolQ.

	Qwen3-0.6B			Qwen3-1.7B		
Teacher	**Corr.**	**Incorr.**	Δ	**Corr.**	**Incorr.**	Δ
SELF-1C	0.047	0.047	≈ 0	-0.006	0.062	-0.068
SELF-1C (32B)	0.030	0.063	-0.034	-0.027	0.067	-0.094
SELF-SUM-1C	0.023	0.059	-0.036	0.021	0.079	-0.057
SELF-1C1W	0.010	0.029	-0.018	-0.013	0.017	-0.030
SELF-SUM-1C1W	-0.005	0.037	-0.042	-0.026	0.042	-0.068
Qwen3-14B	-0.006	0.037	-0.043	0.017	0.055	-0.039
Qwen3-8B	-0.012	0.048	-0.059	0.026	0.080	-0.055
Qwen3-4B	0.001	0.035	-0.035	0.020	0.060	-0.040

F.9 TEACHER CONTEXT GENERATION AND SCREENING DETAILS

To investigate the impact of in-context demonstrations on performance, we design a screening pipeline that measures the impact of various forms of context on pass rates across difficulty levels. For sourced context in summarized demonstrations, we use Qwen3-32B to process demonstrations. For the rest, we use the same model as the student. All models use thinking mode enabled. We screen questions from two benchmarks: MMLU (Hendrycks et al., 2020) (500 questions, multiple-choice) and BoolQ (Clark et al., 2019) (500 questions, yes/no). For each question we sample G=32 rollouts at temperature τ=1.0 and compute the pass rate as the fraction of rollouts producing a correct answer.

Context Generation. Before screening, for each question, we generate demonstration responses by sampling from the context source model at temperature τ_{demo}=0.7. Each response is verified against the ground-truth answer and classified as correct or incorrect. Generation retries up to 60 trials per question to collect the required number of demonstrations.

We evaluate seven context configurations, organized into three families:

1. **Raw demonstrations.** Correct and/or incorrect responses are prepended verbatim to the prompt. Variants: 1 correct (SELF-1C), 1 correct + 1 wrong (SELF-1C1W), and 3 correct (SELF-3C). Additionally, SELF-1C (32B) uses a single correct demonstration generated by Qwen3-32B rather than the student.
2. **Summarized demonstrations.** Qwen3-32B condenses each response to its key reasoning steps and final answer. Variants: 1 summarized correct (SELF-SUM-1C) and 1 summarized correct + 1 wrong (SELF-SUM-1C1W).

Context Injection. All context is injected by prepending it to the user message before applying the chat template. Figures 12–14 show the exact prompt structure for each context mode.

Difficulty Binning and Filtering. We assign each question to a difficulty bin based on its *baseline* (no-context) pass rate: *easy* ($p \geq 0.8$), *medium* ($0.25 \leq p < 0.8$), *hard* ($0 < p < 0.25$), and

SYSTEM

```
You are a helpful assistant.
```

USER

```
Answer the following multiple choice question.  The last line
of your response should be of the following format:  'Answer:
$LETTER' (without quotes) where LETTER is one of ABCD. Think
step by step before answering.

{question}

A) {choice_A}
B) {choice_B}
C) {choice_C}
D) {choice_D}
```

Figure 12: Baseline prompt (no context). Shown for MMLU (multiple-choice).

SYSTEM

```
You are a helpful assistant.
```

USER

```
{question with instruction + choices}
```

Demonstration

```
This is a correct response to the question:
"""
{correct_response_1}
"""
```

```
Now answer with a response of your own, including the thinking
process:
```

Figure 13: SELF-1C prompt. A single correct demonstration is prepended. For SELF-1C (32B) the demonstration is generated by Qwen3-32B instead; the prompt format is identical.

extremely hard ($p = 0$). Extremely hard questions are excluded from the primary analysis as a zero baseline provides no signal for measuring improvement during offline gradient analysis.

To ensure fair comparison across context variants, we restrict the analysis to questions for which the context generation process successfully produced at least one correct demonstration under *both* the self-context and 32B-context conditions. Table 13 reports the number of questions retained after this filter. It is worth to emphasize that the **questions where no correct response could be generated for either source are excluded**, ensuring that observed differences reflect the *quality* of injected context rather than its *availability*.

Several patterns emerge across all models and benchmarks. First, even a single correct demonstration (SELF-1C) as teacher context produces dramatic improvements for teacher accuracy. Second, including wrong demonstrations alongside correct ones (SELF-1C1W) consistently hurts performance relative to correct-only variants, and sometimes even degrades below the no-context baseline. Third, the gap between self-generated and 32B-generated demonstrations (SELF-1C vs. SELF-1C (32B)) is small. Lastly, we also provide difficulty breakdown for accuracy changes in screening questions based on context variations in Tables 15. We observe that the accuracy improvements for medium and easy questions (for baseline) are more significant compared to the improvements for

SYSTEM

```
You are a helpful assistant.
```

USER

```
{question with instruction + choices}
```

Demonstrations

```
Below are example responses to the question.  Some are
CORRECT and some are WRONG -- the wrong responses contain
mistakes and should NOT be imitated.
Correct:
"""
{correct_response_1}
"""
WRONG (do NOT imitate):
"""
{wrong_response_1}
"""
```

```
Now answer with a response of your own, including the thinking
process:
```

Figure 14: SELF-1C1W prompt. One correct and one wrong demonstration with warning header.

SYSTEM

```
You are a helpful assistant.
```

USER

```
{question with instruction + choices}
```

Summarized Demonstration

```
This is a correct response to the question:
"""
{summarized_correct_response_1}
"""
```

```
Now answer with a response of your own, including the thinking
process:
```

Figure 15: SELF-SUM-1C prompt. The demonstration is first condensed by Qwen3-32B.

hard questions, which suggests that especially the smaller models may still tend to generate wrong responses even when correct demonstrations are provided in its context.

Table 13: Number of questions with at least one correct demonstration available among randomly selected questions for screening, and the intersection used for analysis.

Model	Benchmark	Self	32B	Both
Qwen3-0.6B	MMLU	381	459	364
Qwen3-0.6B	BoolQ	473	482	458
Qwen3-1.7B	MMLU	357	459	350
Qwen3-1.7B	BoolQ	474	482	463

Table 14: Screening results. Δ is the absolute improvement over baseline.

	MMLU				BoolQ			
	Qwen3-0.6B		Qwen3-1.7B		Qwen3-0.6B		Qwen3-1.7B	
Context Variant	Acc	Δ	Acc	Δ	Acc	Δ	Acc	Δ
Baseline	63.8	—	85.2	—	78.5	—	88.9	—
SELF-1C	98.6	+35.3	99.7	+14.5	98.5	+19.9	98.3	+10.0
SELF-1C1W	68.2	+4.7	94.0	+8.8	77.6	−0.9	87.7	−0.8
SELF-3C	99.4	+36.1	99.8	+14.6	99.3	+20.7	99.3	+11.0
SELF-SUM-1C	96.4	+32.7	99.3	+14.1	97.4	+18.8	98.2	+9.5
SELF-SUM-1C1W	91.5	+27.8	98.2	+13.0	92.5	+14.0	94.5	+5.8
SELF-1C (32B)	98.8	+35.5	99.7	+14.5	98.4	+19.9	98.7	+10.4

Table 15: Detailed difficulty breakdown for **Qwen3-0.6B** (left) and **Qwen3-1.7B** (right). Each cell contains accuracy (%) and number of questions (n).

Qwen3-0.6B

	Hard		Med.		Easy		All		
Variant	Acc	n	Acc	n	Acc	n	Acc	n	Δ
MMLU									
Baseline	12.6	50	52.9	168	94.6	143	63.8	361	—
SELF-1C	97.4	50	98.1	168	99.6	143	98.6	361	+35.3
SELF-1C1W	41.3	50	59.4	168	88.0	143	68.2	361	+4.7
SELF-3C	98.2	50	99.4	168	100.0	143	99.4	361	+36.1
SELF-SUM-1C	90.5	50	95.3	168	99.7	143	96.4	361	+32.7
SELF-SUM-1C1W	80.0	50	89.0	168	98.3	143	91.5	361	+27.8
SELF-1C (32B)	96.0	50	99.1	168	99.5	143	98.8	361	+35.5
BoolQ									
Baseline	12.9	29	56.2	130	94.6	299	78.5	458	—
SELF-1C	92.6	29	96.9	130	99.7	299	98.5	458	+19.9
SELF-1C1W	35.2	29	57.1	130	90.7	299	77.6	458	−0.9
SELF-3C	96.0	29	98.7	130	99.9	299	99.3	458	+20.7
SELF-SUM-1C	84.1	29	94.8	130	99.7	299	97.4	458	+18.8
SELF-SUM-1C1W	68.6	29	85.1	130	98.1	299	92.5	458	+14.0
SELF-1C (32B)	94.4	29	96.3	130	99.7	299	98.4	458	+19.9

Qwen3-1.7B

	Hard		Med.		Easy		All		
Variant	Acc	n	Acc	n	Acc	n	Acc	n	Δ
MMLU									
Baseline	12.5	15	53.1	70	97.8	265	85.2	350	—
SELF-1C	99.0	15	98.8	70	100.0	265	99.7	350	+14.5
SELF-1C1W	71.9	15	83.0	70	98.1	265	94.0	350	+8.8
SELF-3C	98.8	15	99.4	70	100.0	265	99.8	350	+14.6
SELF-SUM-1C	96.5	15	97.1	70	100.0	265	99.3	350	+14.1
SELF-SUM-1C1W	85.0	15	94.3	70	100.0	265	98.2	350	+13.0
SELF-1C (32B)	100.0	15	98.5	70	100.0	265	99.7	350	+14.5
BoolQ									
Baseline	12.9	16	53.2	66	98.3	378	88.9	460	—
SELF-1C	74.6	16	95.0	66	99.9	378	98.3	460	+10.0
SELF-1C1W	31.6	16	58.2	66	95.3	378	87.7	460	−0.8
SELF-3C	91.0	16	97.5	66	100.0	378	99.3	460	+11.0
SELF-SUM-1C	70.3	16	95.3	66	99.8	378	98.2	460	+9.5
SELF-SUM-1C1W	55.7	16	76.9	66	99.2	378	94.5	460	+5.8
SELF-1C (32B)	85.9	16	96.1	66	99.7	378	98.7	460	+10.4