SIMPLE SELF-DISTILLATION THROUGH ERROR VERBALIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement Learning with Verifiable Rewards (RLVR) has become the de facto approach to enhancing model performance on tasks such as coding and mathematical reasoning. A common feature among popular RL algorithms, such as GRPO (Shao et al., 2024), is to assign every token of a failed rollout the same negative advantage, penalizing sound and unsound reasoning steps alike. *Self-distillation* prompts a teacher model with access to privileged information, such as a correct solution drawn from the model's own successful rollouts, to produce token-level learning signals. Inspired by this, we propose *verbalized self-distillation* (VSD), a novel method for reweighting negative advantages in GRPO. Rather than conditioning the teacher on a raw correct solution, a *writer model* is given a successful rollout and the failed attempt, and is tasked with identifying the first point of divergence in the attempt and distilling this diagnosis into hindsight instructions. The teacher is then conditioned on these instructions instead of the raw rollout. The per-token divergence between the teacher and the current policy is computed over the policy's failed rollout, and subsequently used to reweight the advantages. Across four mathematical reasoning benchmarks, VSD consistently outperforms GRPO and other self-distillation baselines.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has emerged as the standard paradigm for post-training language models on tasks requiring precise, verifiable reasoning, including mathematical problem-solving, code generation, and long-horizon agentic tasks. Simpler supervised approaches such as supervised finetuning (SFT) rely on off-policy targets, which have been shown to cause model drift and catastrophic forgetting (Shenfeld et al., 2026; Yi et al., 2026). GRPO (Shao et al., 2024) addresses this by optimizing on-policy rollouts, using outcome-based rewards as the sole supervision signal. Its success in DeepSeek-R1 (DeepSeek-AI, 2025) has motivated a family of refinements that includes DAPO (Yu et al., 2025) and Dr. GRPO (Liu et al., 2025), tuning its sampling, normalization, and regularization. Despite this progress, a fundamental limitation persists: GRPO assigns the same advantage to every token in a rollout, providing no token-level signal about which reasoning steps were sound and which were erroneous. On-policy distillation (OPD) (Agarwal et al., 2024) addresses this by training a student to match a stronger teacher's next-token distributions along trajectories sampled from the student itself, yielding dense per-token supervision.

Self-distillation methods extend this idea to settings without a stronger external teacher. SDPO (Hübotter et al., 2026) performs self-distillation by conditioning a teacher copy of the model on privileged information, such as a correct solution from the same sampled group, and distilling its token distributions along the student's rollout into the unprivileged policy. SDPO therefore does not require an external teacher: the teacher is derived from the student itself. However, recent work by Kim et al. (2026b) showed that self-distillation methods can degrade out-of-distribution (OOD) generalization on mathematical reasoning tasks, even when training performance improves in-distribution, pointing to two interrelated design weaknesses. The first concerns *how privileged information is constructed*: conditioning the teacher on a raw correct solution suppresses its verbal expressions of uncertainty, what Kim et al. (2026b) call *epistemic verbalization*, which propagates to the student through distillation and limits its capacity for exploration and self-correction on unseen problems. The second concerns *how the teacher signal is used*: prior methods minimize an imitation loss that uniformly drives the student's token distributions toward the teacher's, conflating tokens

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.

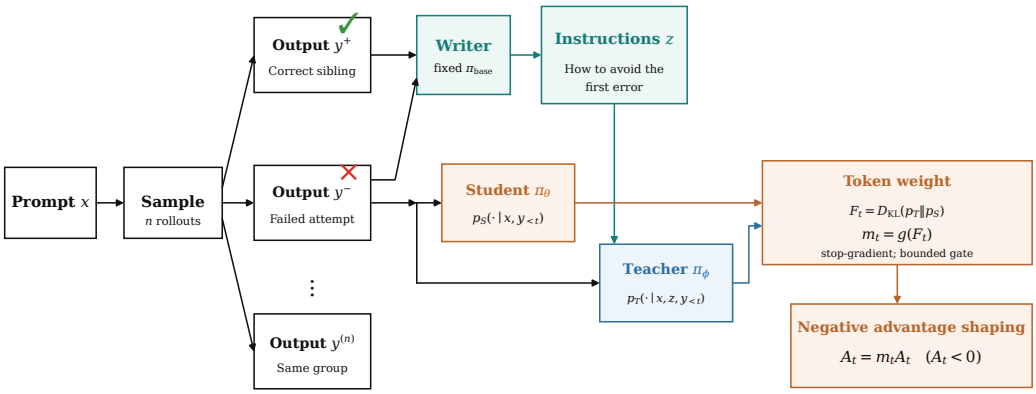

Figure 1: Overview of VSD. For each failed rollout, a fixed writer compares the attempt with a correct sibling and produces instructions describing the error. A teacher receives these instructions, while the student does not, and both produce next-token distributions at each position of the failed rollout. The per-token divergence between these distributions is used as signal to rescale the rollout's negative advantage: large divergence strengthens the penalty, but smaller divergence weakens it.

with real errors and tokens that were only stylistically different from the reference solution (Hübotter et al., 2026; Zhao et al., 2026).

These issues motivate our focus on transferring training gains to out-of-distribution mathematical reasoning benchmarks, rather than improving training-set reward alone. In this work, we present **verbalized self-distillation (VSD)**, a method that directly addresses both of the said weaknesses. Firstly, rather than conditioning the teacher on a raw correct solution, we express the privileged information as additional *instructions to follow* (Figure 1), generated by a fixed writer model. The writer receives the failed rollout and a correct sibling, reasons through the failure, and produces targeted instructions on how to avoid the mistake. The writer's analysis is discarded and only the instructions reach the teacher. This presents the privileged information in a form instruction-tuned models are familiar with: explicit instructions to follow, rather than a reference solution. We hypothesize that this alignment with instruction-following training makes teacher-student disagreement more informative about where the errors actually occur in the failed rollout. Secondly, instead of using the student-teacher disagreement as an imitation loss, VSD uses it as a signal to reweight negative advantages token by token: tokens where the *error-aware* teacher most strongly disagrees receive amplified penalties, while those with small disagreement receive attenuated ones.

To verify that the writer's instructions carry useful privileged information, we conduct an inference-only experiment with a frozen Qwen3-4B (§4.3). On a subset of DAPO-Math problems, supplying the instructions raises pass@1 from 33.9% to 50.4%, confirming that instructions contain actionable information beyond what the model can produce unaided. A qualitative example of the writer's analysis and extracted instructions is provided in Appendix H.

We demonstrate the effectiveness of VSD on four challenging out-of-distribution mathematical reasoning benchmarks: AIME 2024, 2025, 2026, and HMMT February 2025 (Dekoninck et al., 2026). VSD consistently improves the four-benchmark mean pass@1 over GRPO on both Qwen3-4B and Qwen3-1.7B (§4.1). At step 250, VSD reaches mean pass@1 of 0.4755 versus GRPO's 0.4396 on Qwen3-4B ($+3.59$ pp), and 0.2255 versus 0.2094 on Qwen3-1.7B ($+1.61$ pp). On Qwen3-4B, VSD outperforms all baselines throughout training; notably, SDPO collapses to pass@1 between 0.10 and 0.11, consistent with the generalization failures documented by Kim et al. (2026b). Further ablations (§4.2) confirm the value of each design choice used in VSD: at step 250, replacing verbalized instructions with a raw correct sibling solution reduces mean pass@1 by 3.35 points on Qwen3-4B and 1.40 points on Qwen3-1.7B, while removing the privileged context costs 5.55 and 3.38 points, respectively. On Qwen3-1.7B, a simple teacher-free uniform attenuation of negative advantages falls 2.39 points short of VSD, isolating the benefit of adaptive, instruction-guided reweighting.

2 METHOD

Setup Given a prompt x, the student policy π_θ, with parameters θ, generates a rollout $y = (y_1, \ldots, y_T) \sim \pi_\theta(\cdot \mid x)$, where y_t is the token at position t and T is the rollout length. A binary verifier $r(x, y)$ returns 1 for a correct rollout and 0 otherwise. We sample n rollouts per prompt, and write $r^{(i)} = r(x, y^{(i)})$ for the reward of rollout i. The token advantage $\widehat{A}_t^{(i)}$ is the group-centered reward, without standard deviation normalization, following Dr. GRPO (Liu et al., 2025):

$$\widehat{A}_t^{(i)} = r^{(i)} - \frac{1}{n}\sum_{j=1}^{n} r^{(j)}. \tag{1}$$

This advantage is shared by all tokens of a rollout before applying our per-token shaping.

2.1 VERBALIZATION

VSD separates two roles: a *writer* generates privileged information in the form of failure-specific advice, and a *teacher* π_ϕ, with parameters ϕ, uses that advice to assign token probabilities to the student's rollout. Comparing these probabilities with the student's yields the per-token divergence signal (see §2.2). Figure 1 summarizes the process.

Consider a failed rollout $y^{(i)}$ ($r^{(i)}$=0) in a group with at least one correct sibling $y^{(j)}$. The writer receives the problem x, the correct sibling, $y^{(j)}$ as a reference, and the failed attempt $y^{(i)}$, and is prompted to produce two sections. In ANALYSIS, it re-derives a correct solution and examines the attempt step by step to identify the *first error* in the failed rollout, rather than any departure from the reference derivation. In INSTRUCTIONS, it translates that diagnosis into concise, second-person advice on how to avoid the error, with explicit instructions not to reproduce the reference solution or reveal the final answer. Figure 2 provides the full prompt. A complete example is provided in Appendix H.

The writer's analysis section gives it space to check the diagnosis before summarizing it into actionable guidance. We retain only the instructions $z^{(i)}$: neither the analysis nor the reference solution enters the teacher's context. This verbal bottleneck replaces a full alternative solution with targeted advice specific to the failed attempt (see the teacher prompt in Figure 2). The teacher prompt is used only to score the failed rollout, not to generate a replacement solution.

2.2 ADVANTAGE SHAPING

Teacher signal Both teacher and student are teacher-forced over the original failed rollout $y^{(i)}$: at each position, they score the next-token distribution given the same failed prefix, but only the teacher receives the instructions $z^{(i)}$. We denote this prefix by $y^{(i)}_{<t} = (y^{(i)}_1, \ldots, y^{(i)}_{t-1})$ and the student's conditioning context by $s_t = (x, y^{(i)}_{<t})$. Let p_T and p_S denote the teacher's and student's next-token distributions, respectively:

$$\begin{aligned} p_T(\cdot \mid s_t) &= \pi_\phi\left(\cdot \mid x, z^{(i)}, y^{(i)}_{<t}\right), \\ p_S(\cdot \mid s_t) &= \pi_\theta\left(\cdot \mid x, y^{(i)}_{<t}\right). \end{aligned} \tag{2}$$

Here, the subscripts T and S identify the teacher and student. The notation $p_T(\cdot \mid s_t)$ leaves the additional instruction context $z^{(i)}$ implicit. We use a nonnegative approximation F_t to the teacher-to-student Kullback–Leibler (KL) divergence, denoted D_{KL}, as a token-level weighting signal.[1]

$$F_t \approx D_{\mathrm{KL}}(p_T(\cdot \mid s_t) \,\|\, p_S(\cdot \mid s_t)).$$

The approximation is based on the highest-probability student tokens and is specified in §3.1.

[1] We use forward KL since preliminary experiments with reverse KL showed that it assigns low divergence in tokens that are consequential for reasoning.

Writer prompt

```
You are an expert math tutor. Below are a problem, correct reference solution(s), and ONE
SPECIFIC incorrect attempt at it.
# Problem
{prompt}
# Correct reference solution(s)
{references}
# The incorrect attempt to diagnose
{incorrect_attempt}
# Your task
Work in TWO clearly separated sections.
## Analysis
Think it through here, freely. Re-derive the key steps of the problem YOURSELF, then walk
the attempt line by line and find the FIRST place its math or logic genuinely diverges from a
correct path. Actually COMPUTE the value you believe it should be and confirm it leads
toward the correct answer. This section is your scratch work.
## Instructions
Now, grounded in your analysis, write 3-6 sentences of direct, specific advice to whoever
wrote this attempt (address them as 'you'). Name the KEY mistake --- where in the attempt it
happens and WHY it is wrong --- and say concretely what to do differently to correct it. Be
specific to THIS attempt (point at the actual step that went wrong), not generic study tips.
Do NOT restate the whole solution and do NOT reveal the final numerical answer. Output the
'## Analysis' section first, then the '## Instructions' section.
```

Teacher prompt

```
{prompt}
# Instructions from your tutor (follow these):
{instructions}
Correctly solve the original question.
```

Figure 2: Full writer and teacher prompts. The writer performs diagnosis in ANALYSIS and expresses its advice in INSTRUCTIONS; only the latter is inserted into the teacher prompt. Placeholders are filled with the problem, correct reference solutions, failed attempt, and extracted instructions.

Weighting rule We adapt bounded multiplicative advantage shaping for token-level learning (Yang et al., 2026; Kim et al., 2026a), treating F_t as a stop-gradient signal. A rollout is *covered* if it is incorrect, has at least one correct sibling in its group, and yields a nonempty instruction section under the extraction procedure (see Appendix G). Let $c_t = 1$ for every token of a covered rollout and $c_t = 0$ otherwise. For rollout i, let $\widehat{A}_t$ denote the unshaped advantage $\widehat{A}_t^{(i)}$ defined in Eq. 1, with the rollout index suppressed for readability. We first compute a clipped divergence weight w_t, blend it with the neutral weight 1 to obtain the multiplier m_t, and apply it to covered tokens to obtain the shaped advantage $\widehat{A}'_t$:

$$w_t = \mathrm{clamp}\left(\frac{F_t}{\tau},\, w_{\mathrm{lo}},\, w_{\mathrm{hi}}\right), \qquad m_t = \lambda + (1-\lambda)\, w_t, \qquad \widehat{A}'_t = \widehat{A}_t \cdot \big[1 + c_t\,(m_t - 1)\big]. \quad (3)$$

Here clamp clips its first argument to the interval set by the lower and upper bounds w_{lo} and w_{hi}, with $0 < w_{\mathrm{lo}} < 1 < w_{\mathrm{hi}}$. The divergence scale $\tau > 0$ controls how strongly disagreement affects the weight, while $\lambda \in [0, 1]$ interpolates between divergence-based shaping and leaving the advantages unchanged. Larger values of λ pull the multiplier toward 1, reducing the strength of shaping. Setting $\lambda = 1$ recovers the unshaped advantages; $\lambda = 0$ uses the clamped divergence weight alone. For $\lambda < 1$, τ is the amplification threshold: $m_t > 1$ if and only if $F_t > \tau$. Since covered rollouts have negative advantages, factors below one attenuate the failure penalty, while factors above one strengthen it. Uncovered tokens retain their original advantages and require no teacher score.

The multiplier is positive and bounded between $\lambda + (1-\lambda)w_{\mathrm{lo}}$ and $\lambda + (1-\lambda)w_{\mathrm{hi}}$, preserving the advantage's sign and constraining the change in each token's weight. Unlike methods that minimize a distillation loss toward a privileged teacher (Hübotter et al., 2026; Zhao et al., 2026), VSD uses the divergence only to rescale the existing policy-gradient contribution.

3 EXPERIMENTAL SETUP

Model and data We train Qwen3-4B and Qwen3-1.7B (Yang et al., 2025) in non-thinking mode on DAPO-Math-17K (Yu et al., 2025). For all methods, we omit reward-standard-deviation normalization, as in Equation 1, following Dr. GRPO (Liu et al., 2025), and average the policy-gradient loss over all tokens in the batch, following DAPO (Yu et al., 2025). The resulting objective is common to

all baselines and VSD, and is presented in detail in Appendix B. We report results over 250 training steps. The training configuration and hyperparameters are detailed in Appendix B.

3.1 Training and implementation details

Teacher parameterization. The teacher and writer are both frozen copies of the base policy $\pi_{\text{base}} = \pi_{\theta_0}$. The writer generates instructions, while the teacher scores student rollouts conditioned on those instructions to provide the weighting signal. Neither receives gradients from the student objective.

Instruction generation. The writer uses greedy decoding, and extracted instructions are capped at $2{,}048$ tokens unless stated otherwise. It receives the shortest correct siblings in the group as a reference. Extraction failures and empty instructions leave the rollout's advantages unchanged. The exact extraction and fallback rules are given in Appendix G.

KL approximation. For computational efficiency, we approximate the full-vocabulary KL using a reduced set of tokens, so the divergence calculation requires only $k + 1$ terms per token position. At position t, let $\mathcal{K}_t$ be the set of the student's $k = 8$ highest-probability next tokens. We combine all tokens outside $\mathcal{K}_t$ into one additional category, which we call the *tail bucket*. Each model assigns this category the sum of probabilities for all excluded tokens; the same excluded-token set is used for both models. Suppressing the conditioning on s_t, $p_T(v)$ and $p_S(v)$ are the teacher and student probabilities of vocabulary token v. For either distribution, $\bar{p}_\bullet = 1 - \sum_{v \in \mathcal{K}_t} p_\bullet(v)$ is the probability mass outside the retained set, where $\bullet$ stands for T or S. The resulting divergence is

$$F_t = \sum_{v \in \mathcal{K}_t} p_T(v) \log \frac{p_T(v)}{p_S(v)} + \bar{p}_T \log \frac{\bar{p}_T}{\bar{p}_S}, \qquad F_t \geq 0. \tag{4}$$

Shaping hyperparameters. We set λ=0.5, w_{lo}=0.5, w_{hi}=1.5, so $m_t \in [0.75, 1.25]$. We use $\tau = 1$ for Qwen3-4B and $\tau = 0.5$ for Qwen3-1.7B. An initial 1.7B run with $\tau = 1$ kept more covered tokens at the minimum multiplier (Appendix B, Figure 6), motivating a lower threshold so that smaller divergences can affect token weights.

3.2 Baselines and evaluation

Baselines We compare VSD with four alternatives that separate the effects of standard RLVR training, reference-policy regularization, token-level advantage shaping, and privileged-context imitation.

1. **GRPO** : Our primary baseline uses the same model, data, rollout sampling, and group-centered advantages as VSD, but applies no token-level shaping: every token receives the rollout-level advantage in Eq. 1. This isolates the contribution of VSD's verbalized teacher signal.
2. **GRPO + KL:** This baseline augments GRPO with the conventional per-token KL regularizer toward the frozen base policy ($\beta = 10^{-3}$), discouraging the policy from drifting from its initialization. Unlike VSD's divergence, this KL is optimized as a loss over all sampled tokens rather than used to reweight failed rollouts.
3. **RLRT** (Kim et al., 2026a) is a closer token-level shaping baseline. It uses a clipped student–teacher probability ratio to emphasize successful token choices and reweights only correct trajectories, leaving failed ones unchanged. VSD instead uses an instruction-conditioned distributional KL to shape failed trajectories. Reproduction details are in Table 3.
4. **SDPO** (Hübotter et al., 2026) represents privileged-context self-distillation. Its teacher sees a raw correct sibling solution, and the student directly imitates the teacher's token distributions along its on-policy trajectory. VSD instead supplies concise failure-specific instructions and uses the divergence for advantage shaping, not imitation. The raw-privilege ablation in §4.2 uses SDPO-like context with VSD's update rule, separating the effects of context and objective.

We evaluate all baselines on Qwen3-4B and GRPO on Qwen3-1.7B.

Evaluation Kim et al. (2026b) showed that self-distillation leads to lower out-of-distribution performance and so we focus our evaluation on cross-benchmark generalization: after training on

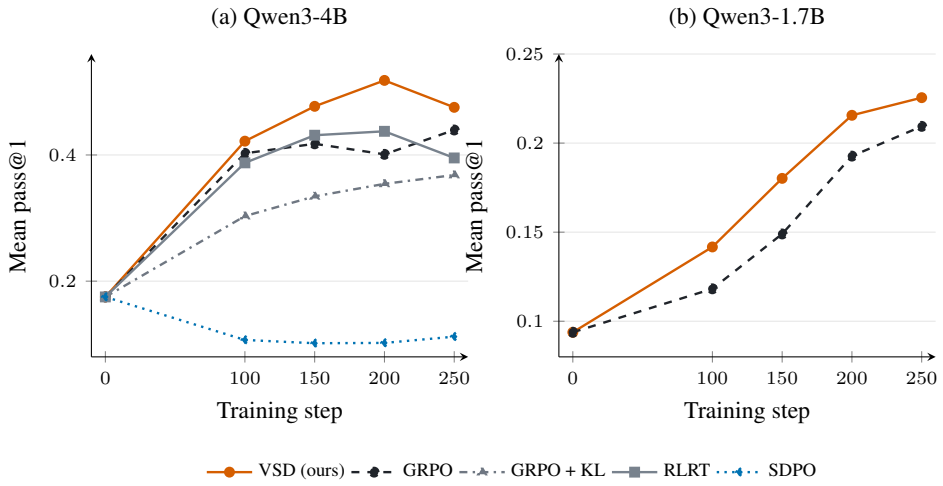

Figure 3: Main results: four-benchmark mean pass@1. **(a)** Qwen3-4B: all methods start from the same initial model before RL (step 0, 0.1750). GRPO + KL uses $\beta = 10^{-3}$; **(b)** Qwen3-1.7B: VSD versus GRPO at matched checkpoints every 50 steps from step 100 through step 250, with the shared base model at step 0 (0.0938). Numerical results are in Tables 1 and 17.

DAPO-Math-17K, we evaluate on AIME 2024, 2025, and 2026 and HMMT February 2025 (Dekoninck et al., 2026). The student solves evaluation problems without the writer's instructions or reference solutions. We report the unweighted mean pass@1 and pass@16 over the four benchmarks, using 16 rollouts per problem at temperature 0.7, top-p 0.95, and a 32k-token budget. The Qwen3-4B runs and the Qwen3-1.7B GRPO baseline are evaluated every 50 steps; Qwen3-1.7B VSD is additionally evaluated every 25 steps for fine-grained ablation comparisons (§4.2).

4 Experiments

4.1 Main results

We present our main results in Figure 3. Panel (a) compares VSD with all baselines on Qwen3-4B, while panel (b) compares VSD and GRPO on Qwen3-1.7B. Exact aggregate results are reported in Table 1 (Appendix A) and Table 17 (Appendix F); per-benchmark Qwen3-4B results appear in Appendix E.

VSD improves mean pass@1 over GRPO at every matched checkpoint for both model scales, and on Qwen3-4B it outperforms every evaluated baseline throughout training. GRPO + KL consistently underperforms GRPO, while RLRT improves over GRPO at intermediate checkpoints but finishes below it. The SDPO reproduction collapses on Qwen3-4B: pass@1 falls from the base model's 0.1750 to 0.10–0.11 and remains there through step 250, consistent with the generalization failures observed by Kim et al. (2026b) in other model configurations. At the final checkpoint, VSD reaches 0.4755 versus GRPO's 0.4396 on Qwen3-4B and 0.2255 versus 0.2094 on Qwen3-1.7B, gains of 3.59 and 1.61 percentage points, respectively.

4.2 Ablations

To further isolate the contributions of VSD, we test whether verbalized instructions provide a more useful teacher signal than two alternative forms of privileged context:

Raw privilege The teacher receives a correct sibling trajectory directly, without a writer, following prior work (Hübotter et al., 2026; Kim et al., 2026a; Yang et al., 2026).

No privilege The teacher receives only the original prompt, with no additional guidance. Teacher and student score the same failed rollout under identical conditioning: the original prompt and the rollout prefix. The teacher remains frozen at the initial weights, whereas the student's weights

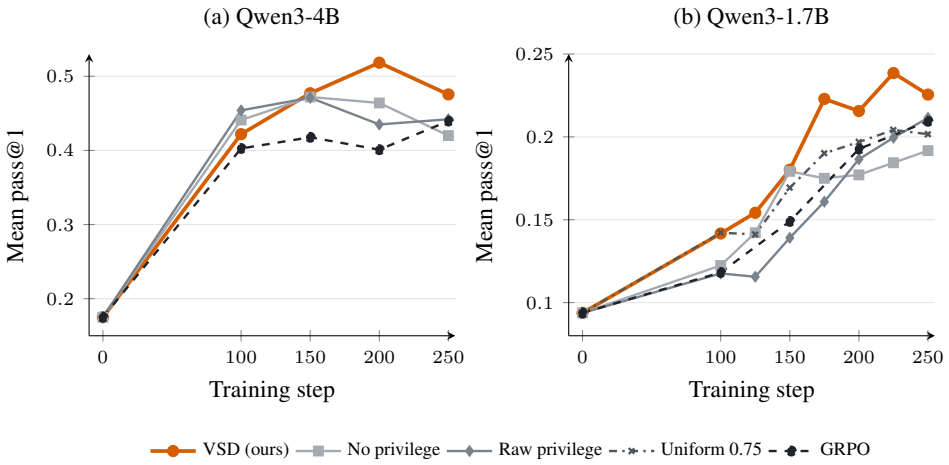

Figure 4: Ablation results: four-benchmark mean pass@1. **(a)** Qwen3-4B context variants and the GRPO baseline, evaluated every 50 steps from step 100 through step 250; **(b)** Qwen3-1.7B ablations. VSD and the three ablation controls are evaluated every 25 steps and GRPO every 50 steps. Numerical results are in Tables 4 and 7.

change through RL updates. Their divergence therefore reflects changes in the student's parameters relative to the fixed teacher, rather than access to privileged information.

Uniform attenuation. Analyzing the results (see Figure 5), we observe that most covered tokens receive VSD's minimum multiplier, $m_t = 0.75$, motivating a simple uniform control. *Uniform 0.75* multiplies every negative advantage by 0.75, leaves nonnegative advantages unchanged, and uses no teacher or token-level distinctions. If VSD helps mainly by weakening failed-rollout updates overall, this control should match it; their difference isolates the value of adaptive, teacher-guided weighting. We run this control only for Qwen3-1.7B.

Overall, VSD, Raw privilege and No privilege are identical except for the context that the teacher observes.

Results. Figure 4 and Tables 4 and 7 show the results of the ablations. On Qwen3-4B, the context controls lead at step 100, but subsequently plateau or decline, while VSD rises from 0.4219 at step 100 to 0.5183 at step 200. VSD achieves the highest mean pass@1 at steps 150, 200, and 250. At step 250, it reaches 0.4755, exceeding raw privilege and no privilege by 3.35 and 5.55 percentage points, respectively. On Qwen3-1.7B, VSD exceeds both context controls at every checkpoint and achieves four-benchmark mean pass@1 of 0.2255, exceeding raw privilege, no privilege, and Uniform 0.75 by 1.40, 3.38, and 2.39 percentage points, respectively. This highlights the importance of adaptive weighting with instructions for empirical performance.

4.3 Analysis

VSD attenuates negative advantages Figure 5 compares the divergence and induced weight regimes for both model sizes. In both cases, the distribution of F_t is right-skewed: the median remains well below the linear clamp band, while the upper percentiles rise during training. For Qwen3-4B ($\tau = 1$), the 99th percentile crosses the upper boundary around step 100 and the 90th percentile reaches it late in training. For Qwen3-1.7B, the lower threshold makes smaller divergences enter the linear regime, while the 99th percentile lies near or above the upper boundary for much of training. The corresponding regime splits initially place nearly all covered tokens at the minimum multiplier $m_t = 0.75$. Late in training, roughly 80% of covered 4B tokens and 85% of covered 1.7B tokens remain at this floor; most of the remainder lie in the linear regime, and only a small minority reach the maximum $m_t = 1.25$. Thus, VSD evolves from near-uniform attenuation to heterogeneous weighting that strengthens a limited subset of failure penalties.

Do instructions increase the probability of success? We directly test whether the instructions generated by the writer improve solutions without updating model weights. We use a frozen Qwen3-

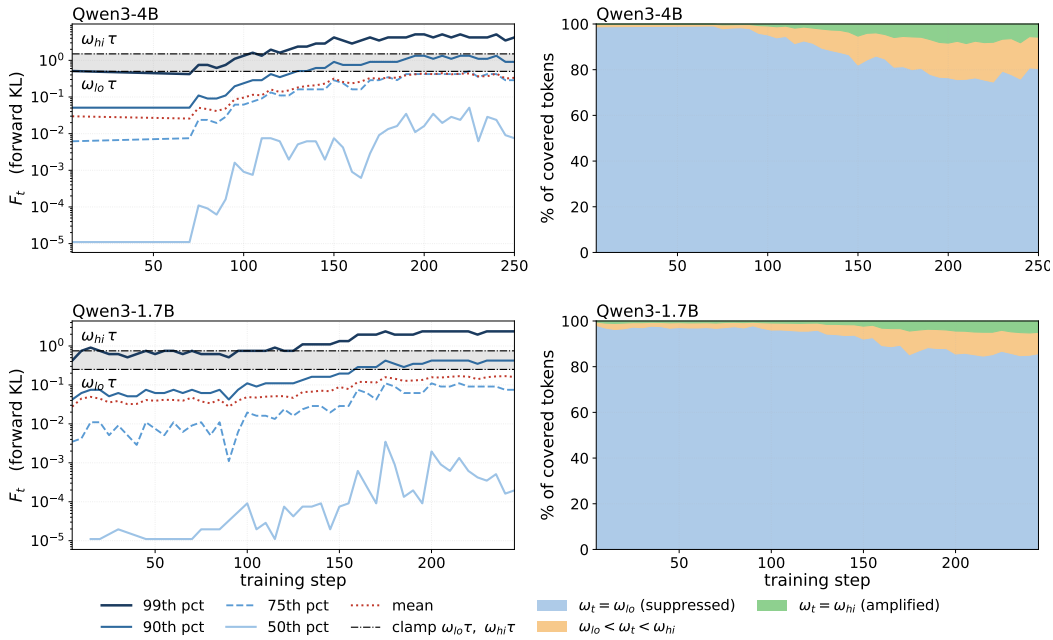

Figure 5: VSD statistics. The top row shows Qwen3-4B with $\tau = 1$; the bottom row shows Qwen3-1.7B with $\tau = 0.5$. **Left:** percentiles of the per-token forward-KL divergence F_t across covered tokens on a log scale; the dotted red curve is the mean and the shaded region is the linear clamp band between $w_{\text{lo}}\tau$ and $w_{\text{hi}}\tau$. **Right:** the fraction of covered tokens whose w_t is clamped up to w_{lo}, lies in the linear band, or is clamped down to w_{hi}. With $\lambda = 0.5$, the two clamped regimes correspond to multipliers $m_t = 0.75$ and $m_t = 1.25$ respectively.

4B, the same base model used as writer and teacher during training, on DAPO-Math cases with both correct and incorrect sibling rollouts. Instructions are generated with the training writer template, greedy decoding, and a 2,048-token cap, retaining only the instruction section under the training extraction rule. To conservatively exclude potential answer disclosure, we discard every case in which the gold answer appears as a substring of the instructions after removing commas and whitespace. Filtering leaves 35 of the 81 analyzed cases. On these problems, we compare the bare prompt with the training teacher-pass template containing the instructions, sampling eight solutions per case per condition at temperature 0.7. Instructions increase pass@1 score from 0.339 to 0.504, a paired lift of 16.4 percentage points with bootstrap interval $[+6.8, +26.4]$ from 4,000 resamples over cases.

Qualitative example Appendix H provides a full example of the analysis and instructions produced by a writer.

5 Related Work

Privileged-context self-distillation Conditioning a model on privileged information and distilling the conditioned distribution into the unconditioned model is the shared design of SDPO (Hübotter et al., 2026), subsequent analyses of self-distillation (Kim et al., 2026b), and context distillation more broadly (Askell et al., 2021). SDPO conditions on a correct sibling solution drawn from the model's own group and the reasoning self-distillation methods in these works generally transfer the teacher signal via a distillation loss while conditioning on the raw solution, though teacher parameterizations and update rules vary. On-policy self-distillation (OPSD) (Zhao et al., 2026) similarly matches the privileged teacher's token distributions along the student's own rollouts, but draws its privileged context from externally supplied reference solutions (including expert reasoning traces), which are not always available. Skill-SD (Wang et al., 2026) conditions its teacher on retrieved summaries of prior trajectories and adds an auxiliary reverse-KL distillation loss. VSD instead generates failure-specific instructions for each incorrect rollout by comparing it with a correct sibling, and uses

the resulting teacher-to-student forward-KL divergence to reweight negative advantages rather than minimize an imitation loss.

Advantage shaping RLSD (Yang et al., 2026) reweights token-level advantages using bounded, sign-preserving multipliers derived from a direction-aware sampled-token probability ratio with a privileged teacher. RLRT (Kim et al., 2026a) reverses this ratio and reweights only correct trajectories, emphasizing successful student choices that the teacher assigns lower probability to; incorrect trajectories retain their original advantages. A3PO (Tang et al., 2026) adaptively and asymmetrically shapes token-level advantages for both positive and negative samples. VSD uses the self-distillation framework with a different signal on a different subset of rollouts. We shape only failed trajectories, using a teacher-to-student distributional KL conditioned on attempt-specific instructions, rather than a probability ratio for the sampled token (§2). Deng et al. (2025) identify *lazy likelihood displacement*, in which negative gradients from incorrect responses hinder increases in the likelihood of correct responses, and propose NTHR, which uses correct group responses as anchors to selectively reduce penalties on tokens contributing to this effect. VSD likewise modifies negative-token penalties, but obtains its signal from an instruction-conditioned teacher and permits both attenuation and amplification rather than only selective attenuation.

Credit assignment in RLVR Process reward models supervise intermediate steps (Uesato et al., 2022; Lightman et al., 2024) but need step annotation or a learned verifier; in contrast, our signal derives from the policy's own response to added context. Sequence-level GRPO variants (Yu et al., 2025) modify sampling or normalization; the shaping here acts below the sequence level. Token-selective methods weight or mask tokens by entropy or policy statistics (Wang et al., 2025).

Hindsight, critique, and KL regularization Madaan et al. (2023) introduced Self-Refine for which the model is prompted to improve its first attempt. VSD on the contrary, presents a writer model with two independently sampled rollouts (one correct and one incorrect) and asks it to spot the first error and uses the output for training. RLHF regularizes toward a reference policy with a KL divergence penalty (Ziegler et al., 2019); in RLVR this term is typically dropped (Yu et al., 2025; Liu et al., 2025; Hu et al., 2025). Although our advantage shaping method also uses a KL term between the teacher and the student, there are multiple structural differences. In our method, the divergence is computed under stop-gradient, so no gradient ever flows through it towards the frozen reference; it acts as a bounded multiplier rather than an unbounded additive penalty; and it influences only the failed trajectories.

6 Conclusion

In this work, we introduce VSD, a self-distillation method that relies on self-generation of privileged information. Specifically, we use a separate frozen writer model, which generates instructions specific to an incorrect rollout by comparing it to a correct solution and summarizing where the model made a mistake. These instructions are used as privileged information for computing a detached teacher-student divergence signal used for reweighting negative advantages. Empirically, we show VSD improves generalization to out-of-distribution setups, improving mean pass@1 over GRPO and other baselines on both Qwen3-4B and Qwen3-1.7B across multiple mathematical reasoning benchmarks. Ablations further substantiate the advantage of using instructions as privileged information compared to raw solutions. Overall, VSD provides a simple self-distillation recipe with strong generalization properties.

AI use statement

In this work, we used generative AI tools to refine hypotheses, provide feedback on research methodology and experimental design, implement methods, support qualitative data analysis, run experiments, retrieve and interpret results, find related works, and aid writing.

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 *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=3zKtaqxLhW.

Amanda Askell, Yuntao Bai, Anna Chen, Dawn Drain, Deep Ganguli, Tom Henighan, Andy Jones, Nicholas Joseph, Ben Mann, Nova DasSarma, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Jackson Kernion, Kamal Ndousse, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, and Jared Kaplan. A general language assistant as a laboratory for alignment, 2021. URL https://arxiv.org/abs/2112.00861.

DeepSeek-AI. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645:633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL https://www.nature.com/articles/s41586-025-09422-z.

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

Wenlong Deng, Yi Ren, Muchen Li, Danica J. Sutherland, Xiaoxiao Li, and Christos Thrampoulidis. On the effect of negative gradient in group relative deep reinforcement optimization. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/6adaf0cbeba11705d4ea67a62044f63d-Abstract-Conference.html.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/ed873d79e7c268c020c4b4db13a2812a-Abstract-Conference.html.

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 *Forty-Third International Conference on Machine Learning (ICML 2026)*, 2026. URL https://arxiv.org/abs/2601.20802.

Jeonghye Kim, Jiwon Jeon, Dongsheng Li, and Yuqing Yang. Rebellious student: Reversing teacher signals for reasoning exploration with self-distilled RLVR, 2026a. URL https://arxiv.org/abs/2605.10781.

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)*, 2026b. URL https://arxiv.org/abs/2603.24472.

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 *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

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. In *Second Conference on Language Modeling (COLM 2025)*, 2025. URL https://openreview.net/forum?id=5PAF7PAY2Y.

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 Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/91edff07232fb1b55a505a9e9f6c0ff3-Abstract-Conference.html.

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

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=qA6FgH0nnZ.

Xinyu Tang, Yuliang Zhan, Zhixun Li, Xin Zhao, Zhenduo Zhang, Zujie Wen, Zhiqiang Zhang, and Jun Zhou. Rethinking sample polarity in reinforcement learning with verifiable rewards. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2928–2954, San Diego, California, United States, 2026. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.134. URL https://aclanthology.org/2026.acl-long.134/.

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, 2022. URL https://arxiv.org/abs/2211.14275.

Hao Wang, Guozhi Wang, Han Xiao, Yufeng Zhou, Yue Pan, Jichao Wang, Ke Xu, Yafei Wen, Xiaohu Ruan, Xiaoxin Chen, and Honggang Qi. Skill-SD: Skill-conditioned self-distillation for multi-turn LLM agents, 2026. URL https://arxiv.org/abs/2604.10674.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui 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 Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/a797c2d2e0c1fdabf4d1ab8cd0b465c6-Abstract-Conference.html.

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

Chenxu Yang, Chuanyu Qin, Qingyi Si, Minghui Chen, Naibin Gu, Dingyu Yao, Zheng Lin, Weiping Wang, Jiaqi Wang, and Nan Duan. Self-distilled RLVR, 2026. URL https://arxiv.org/abs/2604.03128.

Junkeun Yi, Damon Mosk-Aoyama, Baihe Huang, Ritu Gala, Charles Wang, Sugam Devare, Khushi Bhardwaj, Abhibha Gupta, Oleksii Kuchaiev, Jiantao Jiao, Jian Zhang, and Venkat Krishna Srinivasan. PivotRL: High accuracy agentic post-training at low compute cost. In *Third Conference on Language Modeling*, 2026. URL https://openreview.net/forum?id=Ra73M7XgbC.

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 Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html.

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, 2026. URL https://arxiv.org/abs/2601.18734.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. Fine-tuning language models from human preferences, 2019. URL https://arxiv.org/abs/1909.08593.

Table 1: Qwen3-4B, mean pass@1 over the four benchmarks; best value per step is in bold. Per-benchmark results for VSD, GRPO, RLRT, and SDPO are in Appendix E.

step	100	150	200	250
GRPO	0.4026	0.4177	0.4010	0.4396
GRPO + KL	0.3031	0.3344	0.3542	0.3682
VSD	**0.4219**	**0.4771**	**0.5182**	**0.4755**
RLRT (Kim et al., 2026a)	0.3875	0.4313	0.4375	0.3953
SDPO (Hübotter et al., 2026)	0.1068	0.1016	0.1021	0.1120

Table 2: Training configuration for the Qwen3 experiments.

learning rate	10^{-6} (10 warmup steps)
prompts per batch / mini-batch	32 / 32 (one gradient step per batch)
rollouts per prompt (n)	8
rollout temperature	1.0
max response length	8,192 tokens
training steps	250
KL loss / entropy bonus	none / none
divergence support	top-k, $k = 8$, plus tail bucket
gate $(\lambda,\ w_{\mathrm{lo}},\ w_{\mathrm{hi}})$	$(0.5,\ 0.5,\ 1.5)$
teacher policy	fixed base policy π_{base}
writer policy	fixed base policy π_{base}
writer decoding	greedy
instruction cap / reference cap	2,048 / 8,192 tokens

Table 3: RLRT-specific settings for the Qwen3-4B reproduction. We use the thinking-model configuration; these settings are distinct from VSD's gate hyperparameters.

Setting	Value
Initial GRPO/reverse-weight blend ($\lambda_{\mathrm{RLRT},0}$)	0.5
Weight-clipping half-width (ϵ_w)	0.2
Blend decay duration (N_{decay})	30 training steps
Reweighted trajectories	Correct only ($r = 1$)

A Aggregate results (Qwen3-4B)

Table 1 reports the aggregate Qwen3-4B results across the four evaluation benchmarks. The corresponding per-benchmark results are provided in Appendix E.

B Hyperparameters

Table 2 lists the shared training configuration, and Table 3 gives the additional settings used for the RLRT baseline. Figure 6 summarizes the weight regimes that motivate the gate-temperature choice.

B.1 Optimization objective

For a rollout batch $\mathcal{B}$, let T_i be the response length and let $\widehat{A}_t'^{(i)}$ denote the shaped advantage from Eq. 3 (or the corresponding unshaped advantage for a baseline). We minimize the token-mean on-policy objective

$$\mathcal{L}_{\mathrm{PG}}(\theta) = -\frac{1}{\sum_{i\in\mathcal{B}} T_i} \sum_{i\in\mathcal{B}} \sum_{t=1}^{T_i} \widehat{A}_t'^{(i)} \log \pi_\theta\Big(y_t^{(i)} \mid x^{(i)}, y_{<t}^{(i)}\Big). \tag{5}$$

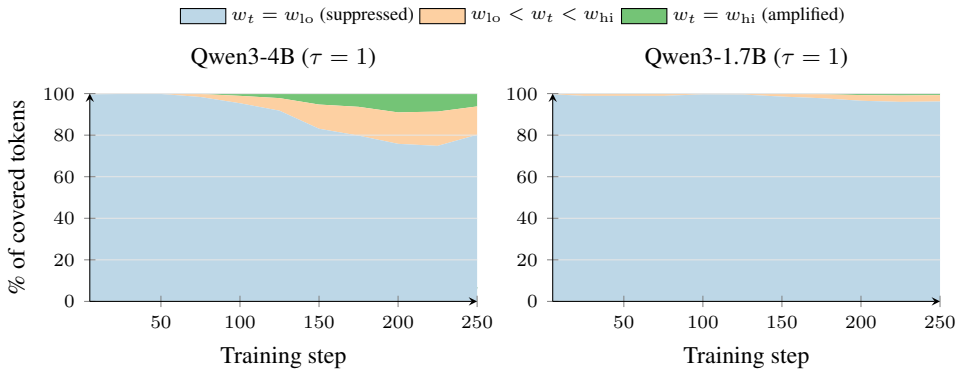

Figure 6: Weight regimes with $\tau = 1$ on both model sizes through step 250. Blue denotes floor-clamped weights ($m_t = 0.75$), orange the linear band, and green ceiling-clamped weights ($m_t = 1.25$). Qwen3-1.7B keeps more covered tokens at the floor, motivating $\tau = 0.5$ to make the gate responsive to smaller divergences.

C Context ablation results (Qwen3-4B)

Table 4: Qwen3-4B context ablations: unweighted four-benchmark mean pass@1 / pass@16.

step	VSD	Raw privilege	No privilege
100	0.4219 / 0.7083	0.4536 / 0.7500	0.4406 / 0.6833
150	0.4771 / 0.7500	0.4713 / 0.7250	0.4724 / 0.7583
200	0.5183 / 0.7418	0.4354 / 0.7250	0.4641 / 0.7333
250	0.4755 / 0.7000	0.4417 / 0.7334	0.4198 / 0.7000

VSD has the highest mean pass@1 at steps 150, 200, and 250, although both context controls lead at step 100. The pass@16 comparison is more mixed: each context variant leads at one or more checkpoints, and raw privilege leads at step 250 with 0.733, ahead of VSD's 0.700.

Table 12 gives the per-benchmark VSD results, while Tables 5 and 6 give the corresponding context controls, with each cell reported as pass@1 / pass@16.

Table 5: Raw-privilege ablation (Qwen3-4B), per benchmark, pass@1 / pass@16.

step	AIME 2024	AIME 2025	AIME 2026	HMMT 2025
100	0.5729 / 0.8000	0.4562 / 0.7667	0.4771 / 0.8333	0.3083 / 0.6000
150	0.6000 / 0.8000	0.4792 / 0.8000	0.5062 / 0.7333	0.3000 / 0.5667
200	0.5583 / 0.8333	0.4437 / 0.8667	0.4417 / 0.7333	0.2979 / 0.4667
250	0.5542 / 0.8333	0.4292 / 0.7667	0.4854 / 0.7667	0.2979 / 0.5667

Table 6: No-privilege ablation (Qwen3-4B), per benchmark, pass@1 / pass@16.

step	AIME 2024	AIME 2025	AIME 2026	HMMT 2025
100	0.5458 / 0.8000	0.4708 / 0.7333	0.4750 / 0.7000	0.2708 / 0.5000
150	0.5875 / 0.8333	0.4979 / 0.8000	0.4896 / 0.8000	0.3146 / 0.6000
200	0.5813 / 0.8000	0.5042 / 0.8000	0.4896 / 0.7667	0.2812 / 0.5667
250	0.5437 / 0.7667	0.4458 / 0.7333	0.4396 / 0.7333	0.2500 / 0.5667

Table 7: Qwen3-1.7B ablations: unweighted four-benchmark mean pass@1 / pass@16.

step	VSD	No privilege	Raw privilege	Uniform 0.75
100	0.1417 / 0.4000	0.1224 / 0.3417	0.1176 / 0.3284	0.1422 / 0.4167
125	0.1542 / 0.3917	0.1422 / 0.4167	0.1156 / 0.3333	0.1411 / 0.3583
150	0.1802 / 0.4167	0.1792 / 0.4417	0.1391 / 0.3833	0.1693 / 0.4167
175	0.2229 / 0.4833	0.1750 / 0.4417	0.1609 / 0.4167	0.1901 / 0.4250
200	0.2156 / 0.5083	0.1771 / 0.4583	0.1865 / 0.4833	0.1969 / 0.4750
225	0.2385 / 0.5250	0.1844 / 0.4500	0.1995 / 0.4667	0.2042 / 0.4833
250	0.2255 / 0.4583	0.1917 / 0.4333	0.2115 / 0.4917	0.2016 / 0.4750

Table 8: VSD (Qwen3-1.7B), per benchmark, pass@1 / pass@16.

step	AIME 2024	AIME 2025	AIME 2026	HMMT 2025
100	0.1833 / 0.4667	0.1375 / 0.4333	0.1354 / 0.3667	0.1104 / 0.3333
125	0.1833 / 0.5000	0.1688 / 0.4333	0.1521 / 0.3667	0.1125 / 0.2667
150	0.2313 / 0.5000	0.1896 / 0.4667	0.1729 / 0.4333	0.1271 / 0.2667
175	0.3042 / 0.6000	0.2250 / 0.4333	0.1938 / 0.4667	0.1688 / 0.4333
200	0.3000 / 0.7000	0.2167 / 0.5333	0.1979 / 0.4667	0.1479 / 0.3333
225	0.3271 / 0.6667	0.2437 / 0.5000	0.2208 / 0.5667	0.1625 / 0.3667
250	0.3271 / 0.6333	0.2271 / 0.4000	0.1979 / 0.4667	0.1500 / 0.3333

Table 9: No-privilege ablation (Qwen3-1.7B), per benchmark, pass@1 / pass@16.

step	AIME 2024	AIME 2025	AIME 2026	HMMT 2025
100	0.1667 / 0.4333	0.1354 / 0.4000	0.1083 / 0.3000	0.0792 / 0.2333
125	0.1938 / 0.5333	0.1604 / 0.4333	0.1271 / 0.3667	0.0875 / 0.3333
150	0.2479 / 0.6000	0.1812 / 0.4667	0.1771 / 0.4333	0.1104 / 0.2667
175	0.2417 / 0.5333	0.1958 / 0.5000	0.1479 / 0.4667	0.1146 / 0.2667
200	0.2417 / 0.6000	0.1938 / 0.4333	0.1604 / 0.4333	0.1125 / 0.3667
225	0.2333 / 0.5333	0.1938 / 0.4000	0.1792 / 0.5333	0.1313 / 0.3333
250	0.2542 / 0.5667	0.2292 / 0.4667	0.1667 / 0.4000	0.1167 / 0.3000

Table 10: Raw-privilege ablation (Qwen3-1.7B), per benchmark, pass@1 / pass@16.

step	AIME 2024	AIME 2025	AIME 2026	HMMT 2025
100	0.1724 / 0.4138	0.1208 / 0.3333	0.1062 / 0.3000	0.0708 / 0.2667
125	0.1708 / 0.4000	0.1062 / 0.3000	0.1083 / 0.3667	0.0771 / 0.2667
150	0.1792 / 0.5000	0.1646 / 0.4667	0.1229 / 0.3000	0.0896 / 0.2667
175	0.2083 / 0.5667	0.1667 / 0.4333	0.1625 / 0.4000	0.1062 / 0.2667
200	0.2583 / 0.6333	0.2000 / 0.5000	0.1667 / 0.4667	0.1208 / 0.3333
225	0.2750 / 0.6000	0.2062 / 0.5333	0.1792 / 0.4333	0.1375 / 0.3000
250	0.2771 / 0.6667	0.2396 / 0.5000	0.1917 / 0.4333	0.1375 / 0.3667

D Ablation results (Qwen3-1.7B)

Pass@16 ablations. The pass@16 comparison is more mixed (Table 7). VSD leads all controls at steps 175, 200, and 225, reaching a peak of 0.5250 at step 225. However, no privilege leads at steps 125 and 150, and uniform attenuation leads at step 100. At step 250, raw privilege leads with 0.4917, ahead of VSD's 0.4583. Thus, VSD's consistent pass@1 advantage over the context controls does not translate into uniformly higher pass@16.

Tables 8–11 give the per-benchmark breakdown for each ablation arm, with each cell reported as pass@1 / pass@16.

Table 11: Uniform-0.75 ablation (Qwen3-1.7B), per benchmark, pass@1 / pass@16.

step	AIME 2024	AIME 2025	AIME 2026	HMMT 2025
100	0.2021 / 0.5333	0.1417 / 0.4667	0.1354 / 0.3667	0.0896 / 0.3000
125	0.2083 / 0.5000	0.1354 / 0.4000	0.1313 / 0.3333	0.0896 / 0.2000
150	0.2479 / 0.5333	0.1750 / 0.4000	0.1521 / 0.4333	0.1021 / 0.3000
175	0.2708 / 0.6000	0.1833 / 0.4333	0.1667 / 0.3667	0.1396 / 0.3000
200	0.2771 / 0.6000	0.1917 / 0.5333	0.1979 / 0.4333	0.1208 / 0.3333
225	0.3021 / 0.6333	0.1812 / 0.4667	0.2062 / 0.5333	0.1271 / 0.3000
250	0.2583 / 0.6000	0.2042 / 0.5333	0.2000 / 0.4667	0.1437 / 0.3000

Table 12: VSD (Qwen3-4B), per benchmark, pass@1 / pass@16.

step	AIME 24	AIME 25	AIME 26	HMMT 25
0	0.2229 / 0.5333	0.1958 / 0.4667	0.1688 / 0.4667	0.1125 / 0.2667
100	0.5375 / 0.800	0.4500 / 0.767	0.4396 / 0.733	0.2604 / 0.533
150	0.5917 / 0.833	0.5083 / 0.767	0.4813 / 0.733	0.3271 / 0.667
200	0.6438 / 0.833	0.5438 / 0.800	0.5375 / 0.767	0.3479 / 0.567
250	0.6146 / 0.833	0.5000 / 0.733	0.4875 / 0.767	0.3000 / 0.467

Table 13: GRPO + KL ($\beta = 10^{-3}$; Qwen3-4B), per benchmark, pass@1 / pass@16.

step	AIME 24	AIME 25	AIME 26	HMMT 25
0	0.2229 / 0.5333	0.1958 / 0.4667	0.1688 / 0.4667	0.1125 / 0.2667
100	0.3937 / 0.7333	0.3063 / 0.6000	0.3187 / 0.7000	0.1938 / 0.5000
150	0.4354 / 0.7667	0.3354 / 0.7333	0.3479 / 0.7333	0.2188 / 0.4667
200	0.4771 / 0.8000	0.3542 / 0.7333	0.3625 / 0.6000	0.2229 / 0.5667
250	0.4750 / 0.8000	0.4083 / 0.7333	0.3771 / 0.7333	0.2125 / 0.5000

Table 14: RLRT baseline (Qwen3-4B), per benchmark, pass@1 / pass@16.

step	AIME 24	AIME 25	AIME 26	HMMT 25
0	0.2229 / 0.5333	0.1958 / 0.4667	0.1688 / 0.4667	0.1125 / 0.2667
100	0.4792 / 0.833	0.4063 / 0.800	0.3875 / 0.700	0.2771 / 0.467
150	0.5500 / 0.800	0.4521 / 0.767	0.4625 / 0.733	0.2604 / 0.433
200	0.5438 / 0.800	0.4479 / 0.767	0.4813 / 0.800	0.2771 / 0.533
250	0.4917 / 0.800	0.4229 / 0.733	0.4229 / 0.733	0.2438 / 0.433

E PER-BENCHMARK RESULTS (QWEN3-4B)

Table 1 reports the four-benchmark mean; Tables 12–16 give the per-benchmark breakdown behind it, each cell as pass@1 / pass@16. Figure 7 plots the pass@1 trajectories separately for each benchmark.

F VSD VERSUS GRPO (QWEN3-1.7B)

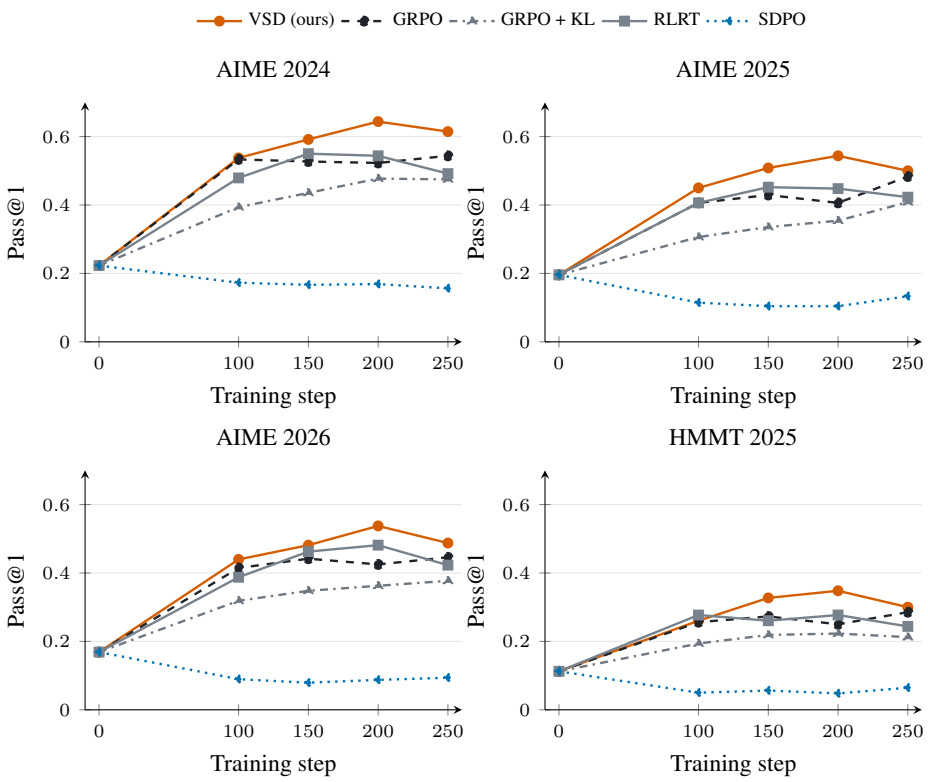

Figure 7: Qwen3-4B pass@1 by benchmark from the shared base model at step 0 through step 250. All panels use the same vertical scale. Curves show VSD, GRPO, GRPO + KL ($\beta = 10^{-3}$), RLRT, and SDPO from Tables 12–16.

Table 15: SDPO baseline (Qwen3-4B), per benchmark, pass@1 / pass@16.

step	AIME 24	AIME 25	AIME 26	HMMT 25
0	0.2229 / 0.5333	0.1958 / 0.4667	0.1688 / 0.4667	0.1125 / 0.2667
100	0.1729 / 0.467	0.1146 / 0.367	0.0896 / 0.267	0.0500 / 0.167
150	0.1667 / 0.500	0.1042 / 0.367	0.0792 / 0.200	0.0563 / 0.167
200	0.1688 / 0.467	0.1042 / 0.400	0.0875 / 0.200	0.0479 / 0.167
250	0.1563 / 0.467	0.1333 / 0.367	0.0938 / 0.267	0.0646 / 0.200

Table 16: GRPO baseline (Qwen3-4B), per benchmark, pass@1 / pass@16.

step	AIME 24	AIME 25	AIME 26	HMMT 25
0	0.2229 / 0.5333	0.1958 / 0.4667	0.1688 / 0.4667	0.1125 / 0.2667
100	0.5333 / 0.833	0.4062 / 0.767	0.4146 / 0.800	0.2562 / 0.500
150	0.5271 / 0.867	0.4292 / 0.733	0.4417 / 0.733	0.2729 / 0.600
200	0.5229 / 0.800	0.4062 / 0.767	0.4250 / 0.733	0.2500 / 0.467
250	0.5437 / 0.833	0.4833 / 0.767	0.4458 / 0.800	0.2854 / 0.500

Table 17: Qwen3-1.7B main results: unweighted four-benchmark means. VSD and GRPO are reported at matched 50-step checkpoints from step 100 through step 250. Step 0 reports the shared base model before RL training.

step	VSD pass@1	GRPO pass@1	VSD pass@16	GRPO pass@16
0	0.0938	0.0938	0.2167	0.2167
100	0.1417	0.1182	0.4000	0.2917
150	0.1802	0.1490	0.4167	0.3833
200	0.2156	0.1927	0.5083	0.4667
250	0.2255	0.2094	0.4583	0.4917

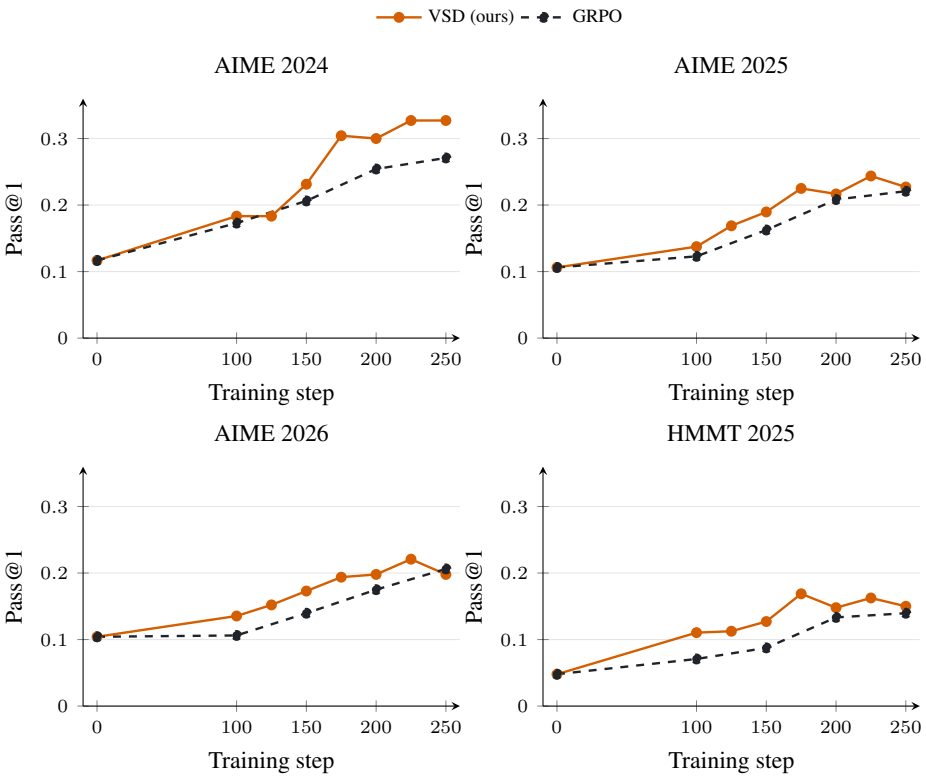

Figure 8: Qwen3-1.7B per-benchmark pass@1 for VSD and GRPO. Both runs start from the shared base model at step 0. VSD is evaluated every 25 steps from step 100 through 250, while GRPO is evaluated every 50 steps. All panels use the same vertical scale.

Table 18: Qwen3-1.7B base model before RL training: per-benchmark evaluation results. The final row gives the reported unweighted four-benchmark means (hard-4).

Benchmark	pass@1	pass@16	Problems
AIME 2024	0.1167	0.3000	30
AIME 2025	0.1062	0.1667	30
AIME 2026	0.1042	0.3000	30
HMMT 2025	0.0479	0.1000	30
Mean (hard-4)	0.0938	0.2167	—

G INSTRUCTION EXTRACTION

The full writer and teacher prompts appear in Figure 2.

Extraction rule. The extracted instructions $z^{(i)}$ are the text following a header line matching the regular expression `(?:^|\n)[ \t]*#*[ \t]*Instructions[ \t]*:?[ \t]*\n` (case-insensitive), stripped of leading/trailing whitespace. If that header is absent, we fall back to text starting at a line matching `\n[-*\s]*Step\s*[:*]`, to tolerate a writer that skipped the INSTRUCTIONS header. If neither pattern matches, extraction fails and the rollout receives no shaping (§2).

H QUALITATIVE VERBALIZATION EXAMPLE

The following example, taken from the Qwen3-4B model, shows the complete input to the writer and its output: the problem, a correct sibling rollout, the failed rollout, the writer's analysis, and the extracted failure-specific instructions. We preserve the model-generated text verbatim. For visibility, the **correct reference** is shown in green, the **incorrect attempt** in red, and the **writer output** in blue.

```
Let $a, b$, and $c$ be the roots of the cubic polynomial $2x^3 - 3x^2 + 165x - 4$. Compute
↪  \[(a+b-1)^3 + (b+c-1)^3 + (c+a-1)^3.\]
Please reason step by step, and put your final answer within \boxed{}.
```

```
We are given a cubic polynomial:

$$
2x^3 - 3x^2 + 165x - 4
$$

Let $ a, b, c $ be the roots of this polynomial. We are to compute:

$$
(a + b - 1)^3 + (b + c - 1)^3 + (c + a - 1)^3
$$

---

### Step 1: Use Vieta's formulas

For a cubic polynomial $ 2x^3 - 3x^2 + 165x - 4 = 0 $, the roots $ a, b, c $ satisfy:

- $ a + b + c = \frac{3}{2} $ (since the coefficient of $ x^2 $ is -3, and Vieta's formula
↪  says sum of roots is $ -(-3)/2 = 3/2 $)
- $ ab + bc + ca = \frac{165}{2} $
- $ abc = \frac{4}{2} = 2 $

Let's denote:

- $ S = a + b + c = \frac{3}{2} $
- $ P = ab + bc + ca = \frac{165}{2} $
- $ Q = abc = 2 $

---

### Step 2: Simplify the expression

We need to compute:

$$
(a + b - 1)^3 + (b + c - 1)^3 + (c + a - 1)^3
$$

Note that:

- $ a + b = S - c $
- $ b + c = S - a $
- $ c + a = S - b $

So, substituting:

$$
```

```
(a + b - 1)^3 + (b + c - 1)^3 + (c + a - 1)^3 = (S - c - 1)^3 + (S - a - 1)^3 + (S - b -
↪  1)^3
$$

Let's denote $ S - 1 = \frac{3}{2} - 1 = \frac{1}{2} $

So the expression becomes:

$$
\left( \frac{1}{2} - c \right)^3 + \left( \frac{1}{2} - a \right)^3 + \left( \frac{1}{2} -
↪  b \right)^3
$$

So we're looking for:

$$
\sum_{\text{cyc}} \left( \frac{1}{2} - a \right)^3
$$

Let's expand one of these terms:

$$
\left( \frac{1}{2} - a \right)^3 = \left( \frac{1}{2} \right)^3 - 3 \left( \frac{1}{2}
↪  \right)^2 a + 3 \left( \frac{1}{2} \right) a^2 - a^3
$$

So the total sum becomes:

$$
3 \left( \frac{1}{2} \right)^3 - 3 \left( \frac{1}{2} \right)^2 (a + b + c) + 3 \left(
↪  \frac{1}{2} \right) (a^2 + b^2 + c^2) - (a^3 + b^3 + c^3)
$$

Let's compute each part step by step.

---

### Step 3: Compute each component

#### (a) First term: $ 3 \cdot \left( \frac{1}{2} \right)^3 = 3 \cdot \frac{1}{8} =
↪  \frac{3}{8} $

#### (b) Second term: $ -3 \cdot \left( \frac{1}{4} \right) \cdot (a + b + c) =
↪  -\frac{3}{4} \cdot \frac{3}{2} = -\frac{9}{8} $

#### (c) Third term: $ 3 \cdot \frac{1}{2} \cdot (a^2 + b^2 + c^2) = \frac{3}{2} (a^2 + b^2
↪  + c^2) $

We need to compute $ a^2 + b^2 + c^2 $. Recall:

$$
a^2 + b^2 + c^2 = (a + b + c)^2 - 2(ab + bc + ca)
$$

Plug in the known values:

$$
\left( \frac{3}{2} \right)^2 - 2 \cdot \frac{165}{2} = \frac{9}{4} - 165 = \frac{9}{4} -
↪  \frac{660}{4} = -\frac{651}{4}
$$

So third term is:

$$
\frac{3}{2} \cdot \left( -\frac{651}{4} \right) = -\frac{1953}{8}
$$

#### (d) Fourth term: $ - (a^3 + b^3 + c^3) $

We need to compute $ a^3 + b^3 + c^3 $. There is a formula:

$$
a^3 + b^3 + c^3 = (a + b + c)^3 - 3(a + b + c)(ab + bc + ca) + 3abc
$$

Plug in:

- $ a + b + c = \frac{3}{2} $
- $ ab + bc + ca = \frac{165}{2} $
- $ abc = 2 $

Compute step by step:
```

```
First term: $ \left( \frac{3}{2} \right)^3 = \frac{27}{8} $

Second term: $ -3 \cdot \frac{3}{2} \cdot \frac{165}{2} = -3 \cdot \frac{495}{4} =
↪  -\frac{1485}{4} $

Third term: $ +3 \cdot 2 = 6 $

So total:

$$
a^3 + b^3 + c^3 = \frac{27}{8} - \frac{1485}{4} + 6 = \frac{27}{8} - \frac{2970}{8} +
↪  \frac{48}{8} = \frac{27 - 2970 + 48}{8} = \frac{-2895}{8}
$$

Thus, the fourth term is:

$$
- (a^3 + b^3 + c^3) = \frac{2895}{8}
$$

---

### Step 4: Combine all terms

Now add up all four parts:

- First term: $ \frac{3}{8} $
- Second term: $ -\frac{9}{8} $
- Third term: $ -\frac{1953}{8} $
- Fourth term: $ \frac{2895}{8} $

Total:

$$
\frac{3 - 9 - 1953 + 2895}{8} = \frac{(3 - 9) + (-1953 + 2895)}{8} = \frac{-6 + 942}{8} =
↪  \frac{936}{8} = 117
$$

---

### Final Answer:

$$
\boxed{117}
$$
```

```
We are given a cubic polynomial:

$$
2x^3 - 3x^2 + 165x - 4
$$

Let $ a, b, c $ be its roots. We are to compute:

$$
(a + b - 1)^3 + (b + c - 1)^3 + (c + a - 1)^3
$$

---

### Step 1: Use Vieta's formulas

For a cubic polynomial $ 2x^3 - 3x^2 + 165x - 4 $, the roots $ a, b, c $ satisfy:

- $ a + b + c = \frac{3}{2} $
- $ ab + bc + ca = \frac{165}{2} $
- $ abc = \frac{4}{2} = 2 $

This is because for a general cubic polynomial $ px^3 + qx^2 + rx + s $, the sum of roots
↪  is $ -q/p $, the sum of products of roots two at a time is $ r/p $, and the product of
↪  roots is $ -s/p $.

---

### Step 2: Simplify the expression

We are to compute:

$$
(a + b - 1)^3 + (b + c - 1)^3 + (c + a - 1)^3
```

```
$$

Note that:

- $ a + b = (a + b + c) - c = \frac{3}{2} - c $
- Similarly for the others.

So:

$$
(a + b - 1) = \frac{3}{2} - c - 1 = \frac{1}{2} - c
$$
$$
(b + c - 1) = \frac{1}{2} - a
$$
$$
(c + a - 1) = \frac{1}{2} - b
$$

Therefore, the expression becomes:

$$
\left( \frac{1}{2} - c \right)^3 + \left( \frac{1}{2} - a \right)^3 + \left( \frac{1}{2} -
↪  b \right)^3
$$

Let's denote $ x = \frac{1}{2} $, so the expression is:

$$
(x - c)^3 + (x - a)^3 + (x - b)^3
$$

Now, expand each term:

$$
(x - c)^3 = x^3 - 3x^2c + 3xc^2 - c^3
$$

Similarly for the others:

$$
(x - a)^3 = x^3 - 3x^2a + 3xa^2 - a^3
$$
$$
(x - b)^3 = x^3 - 3x^2b + 3xb^2 - b^3
$$

Adding these three together:

$$
3x^3 - 3x^2(a + b + c) + 3x(a^2 + b^2 + c^2) - (a^3 + b^3 + c^3)
$$

So:

$$
\text{Expression} = 3x^3 - 3x^2(a + b + c) + 3x(a^2 + b^2 + c^2) - (a^3 + b^3 + c^3)
$$

Now, substitute $ x = \frac{1}{2} $, and we already know $ a + b + c = \frac{3}{2} $.

We need values of $ a^2 + b^2 + c^2 $ and $ a^3 + b^3 + c^3 $.

---

### Step 3: Compute $ a^2 + b^2 + c^2 $

We use the identity:

$$
a^2 + b^2 + c^2 = (a + b + c)^2 - 2(ab + bc + ca)
$$

Plug in known values:

$$
a^2 + b^2 + c^2 = \left( \frac{3}{2} \right)^2 - 2 \cdot \frac{165}{2} = \frac{9}{4} - 165
↪  = \frac{9}{4} - \frac{660}{4} = -\frac{651}{4}
$$

Wait, that can't be right. Wait, the polynomial is $ 2x^3 - 3x^2 + 165x - 4 $, so the
↪  coefficient of $ x $ is 165, meaning:
```

```
$$
ab + bc + ca = \frac{165}{2}
$$

So:

$$
a^2 + b^2 + c^2 = \left( \frac{3}{2} \right)^2 - 2 \cdot \frac{165}{2} = \frac{9}{4} - 165
↪  = \frac{9 - 660}{4} = -\frac{651}{4}
$$

Yes, that's correct. The sum of squares is negative. That is possible if the roots are
↪  complex.

---

### Step 4: Compute $ a^3 + b^3 + c^3 $

We use the identity:

$$
a^3 + b^3 + c^3 = (a + b + c)^3 - 3(a + b + c)(ab + bc + ca) + 3abc
$$

Substitute:

- $ a + b + c = \frac{3}{2} $
- $ ab + bc + ca = \frac{165}{2} $
- $ abc = 2 $

Then:

$$
a^3 + b^3 + c^3 = \left( \frac{3}{2} \right)^3 - 3 \cdot \frac{3}{2} \cdot \frac{165}{2} +
↪  3 \cdot 2
$$

Calculate each term:

- $ \left( \frac{3}{2} \right)^3 = \frac{27}{8} $
- $ 3 \cdot \frac{3}{2} \cdot \frac{165}{2} = \frac{9 \cdot 165}{4} = \frac{1485}{4} $
- $ 3 \cdot 2 = 6 $

So:

$$
a^3 + b^3 + c^3 = \frac{27}{8} - \frac{1485}{4} + 6
$$

Convert all to eighths:

- $ \frac{27}{8} $
- $ \frac{1485}{4} = \frac{2970}{8} $
- $ 6 = \frac{48}{8} $

So:

$$
a^3 + b^3 + c^3 = \frac{27}{8} - \frac{2970}{8} + \frac{48}{8} = \frac{27 + 48 - 2970}{8} =
↪  \frac{-2995}{8}
$$

---

### Step 5: Plug into the expression

Recall the expression:

$$
3x^3 - 3x^2(a + b + c) + 3x(a^2 + b^2 + c^2) - (a^3 + b^3 + c^3)
$$

We have:

- $ x = \frac{1}{2} $
- $ x^3 = \frac{1}{8} $
- $ x^2 = \frac{1}{4} $
- $ a + b + c = \frac{3}{2} $
- $ a^2 + b^2 + c^2 = -\frac{651}{4} $
- $ a^3 + b^3 + c^3 = -\frac{2995}{8} $

Now compute each term:
```

```
- $ 3x^3 = 3 \cdot \frac{1}{8} = \frac{3}{8} $
- $ -3x^2(a + b + c) = -3 \cdot \frac{1}{4} \cdot \frac{3}{2} = -\frac{9}{8} $
- $ 3x(a^2 + b^2 + c^2) = 3 \cdot \frac{1}{2} \cdot \left( -\frac{651}{4} \right) =
↪  -\frac{1953}{8} $
- $ -(a^3 + b^3 + c^3) = -\left( -\frac{2995}{8} \right) = \frac{2995}{8} $

Now sum them all:

$$
\frac{3}{8} - \frac{9}{8} - \frac{1953}{8} + \frac{2995}{8}
$$

Combine numerators:

$$
(3 - 9 - 1953 + 2995) / 8 = (2995 - 1953 - 6) / 8 = (1036) / 8 = 129.5
$$

Wait, let me double-check:

3 - 9 = -6

-6 - 1953 = -1959

-1959 + 2995 = 1036

Yes.

So:

$$
\frac{1036}{8} = 129.5
$$

But this is a fractional result. However, the original expression is a sum of cubes, which
↪  might be a fraction. Let's write it as:

$$
\frac{1036}{8} = \frac{518}{4} = \frac{259}{2}
$$

Wait:

Wait 1036 ÷ 2 = 518, then ÷ 2 again is 259. So yes:

$$
\frac{1036}{8} = \frac{259}{2}
$$

Thus, the final answer is:

$$
\boxed{\dfrac{259}{2}}
$$

---

### Final Answer:

$$
\boxed{\dfrac{259}{2}}
$$
```

```
## Analysis
The correct approach involves expanding the expression and carefully computing each term
↪  using Vieta's formulas and identities. The incorrect attempt correctly identifies the
↪  expression as $ \left( \frac{1}{2} - a \right)^3 + \left( \frac{1}{2} - b \right)^3 +
↪  \left( \frac{1}{2} - c \right)^3 $, and expands it properly. However, the critical
↪  mistake occurs in the computation of $ a^3 + b^3 + c^3 $. In the incorrect attempt, the
↪  value is calculated as $ -\frac{2995}{8} $, but the correct value should be $
↪  -\frac{2895}{8} $. This error stems from a miscalculation in the arithmetic when
↪  combining the terms: $ \frac{27}{8} - \frac{1485}{4} + 6 $ was incorrectly evaluated as
↪  $ \frac{27}{8} - \frac{2970}{8} + \frac{48}{8} $, but the correct value of $
↪  \frac{1485}{4} $ is $ \frac{2970}{8} $, and the correct result of the expression is $
↪  -\frac{2895}{8} $, not $ -\frac{2995}{8} $. This small arithmetic error propagates
↪  through the rest of the calculation, leading to an incorrect final result.

## Instructions
```

```
You made a mistake in calculating $ a^3 + b^3 + c^3 $. Specifically, when you computed $
↪  \frac{27}{8} - \frac{1485}{4} + 6 $, you incorrectly converted $ \frac{1485}{4} $ to $
↪  \frac{2970}{8} $, which is correct, but then you incorrectly added $ 6 $ as $
↪  \frac{48}{8} $, which is also correct. However, when you combined the terms, you should
↪  have gotten $ \frac{27 - 2970 + 48}{8} = \frac{-2895}{8} $, not $ \frac{-2995}{8} $.
↪  This error led to an incorrect final result. To fix this, double-check your arithmetic
↪  when combining the terms in the expression for $ a^3 + b^3 + c^3 $, and ensure that you
↪  correctly compute each part before combining them.
```