THE WISDOM OF WRONG CROWDS: LEARNING FROM DISAGREEMENT AMONG FAILED ROLLOUTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) trains language models from verifiable outcomes, but with binary rewards, the outcome-based advantage in group-relative methods vanishes when every rollout for a problem is incorrect. Many recent self-distillation methods provide token-level supervision by modulating verifier-derived advantages, introducing external correct information, or constructing reference-free signals for failed rollouts. However, relations among failed rollouts within an all-wrong group remain underexplored as a source of token-level supervision. We observe that all-wrong groups need not be information-free: failed rollouts can make different mistakes, and disagreement in their answers provides a coarse cue to differences in their reasoning. We introduce **SibCredit**, which converts these answer relations into selective token-level negative credit by re-scoring a target's sampled tokens under contexts constructed from other failed rollouts in the same group. Different-answer rollouts provide candidate negative evidence, but their opposition can also reflect shared conditioning effects, such as shifts in reasoning style. Same-answer rollouts serve as a control, retaining penalties only when different-answer opposition sufficiently exceeds same-answer opposition. Because every conditioning rollout is incorrect, SibCredit uses only induced probability decreases as candidate negative evidence and discards probability increases. The resulting token-level signal is constructed entirely from the current all-wrong group, while other groups use standard GRPO. Experiments on Qwen3-1.7B and Qwen3-4B across mathematical reasoning and code generation show that SibCredit consistently improves over GRPO and recent self-distillation baselines. Token-level analysis further shows that same-answer calibration retains a larger fraction of opposition strength on annotated error tokens than on epistemic expressions.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become a standard approach for improving the reasoning capabilities of language models. Many widely used RLVR algorithms, including GRPO (Shao et al., 2024), RLOO (Ahmadian et al., 2024), DAPO (Yu et al., 2025), and GSPO (Zheng et al., 2025), estimate advantages from reward differences among rollouts sampled for the same problem. Such outcome-level supervision is coarse: it indicates whether a response succeeds, but not which intermediate decisions cause it to fail. With binary verifier rewards, when every rollout in a group fails, all rollouts receive the same zero reward. Such *all-wrong groups* are common in practice and important. We find that, in our mathematical reasoning runs with Qwen3-1.7B and Qwen3-4B (Yang et al., 2025a), all-wrong groups account for roughly 25% and 15% of sampled groups, respectively. Recovering supervision from these groups is valuable because it can provide learning signals for problems that are challenging for the current policy.

Recent work provides signals for failed rollouts mainly through on-policy self-distillation (OPSD; Zhao et al., 2026). Some methods use this signal to modulate verifier-derived advantages, but such modulation alone cannot recover learning signals when those advantages vanish (Yang et al., 2026; Pan et al., 2026). Reference-guided methods introduce external correct information, such as reference solutions, reference hints, or previously successful trajectories, to provide token-level super-

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.

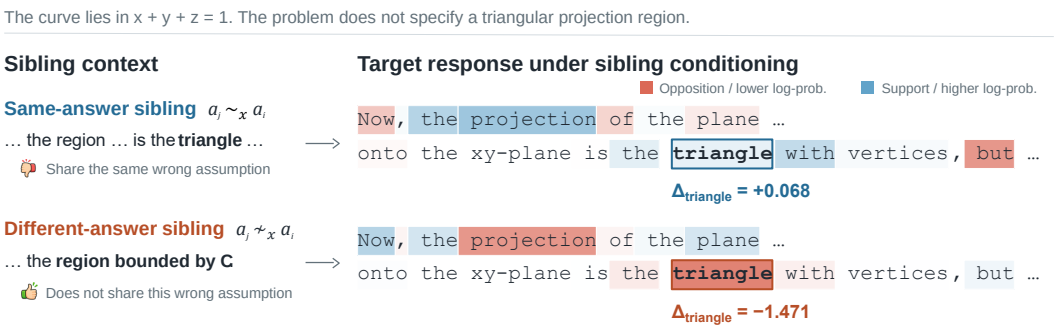

Figure 1: Local opposition under two sibling contexts. The same-answer sibling shares the target's unsupported triangular-region assumption, while the different-answer sibling opposes the token "triangle." The displayed Δ values are signed teacher-minus-base log-probability gaps. This example illustrates the intuition.

vision (Xu et al., 2026; Zhang et al., 2026; Cai et al., 2026; Luo et al., 2026); such information, however, can be costly to obtain or unavailable, particularly for hard problems. Reference-free methods instead assign nonzero advantages to failed rollouts or construct token-level supervision without correct privileged context (Li et al., 2026b; Pei et al., 2026), but do not exploit relations among multiple failed rollouts within the group. We therefore ask: **can the failed rollouts in an all-wrong group, by themselves, provide useful supervision?**

We find that the answer lies in how failures differ. Rollouts for the same problem often share part of their reasoning before diverging (Sullivan & Koller, 2026), and failed rollouts frequently end at different wrong answers. Across our runs, over 85% of all-wrong groups contain at least two distinct answers. Two failed rollouts that reach different answers must have taken different decisions somewhere. Comparing a failed rollout with its alternatives can therefore indicate not only that it fails, but where it may have gone wrong, which is precisely the token-level information that outcome rewards lack. We refer to other rollouts for the same problem as *siblings*. Conditioning the model on a different-answer sibling and re-scoring the target rollout reveals which target tokens lose support given a different failed reasoning path. In Figure 1, a different-answer sibling sharply lowers the probability of the target's unsupported assumption "triangle" ($\Delta = -1.47$), whereas a same-answer sibling that shares this assumption does not ($\Delta = +0.07$).

Turning this observation into a training signal is, however, not straightforward for two reasons. First, different-answer opposition is not specific to answer disagreement. Conditioning on any putative reference shifts the target distribution for reasons unrelated to correctness, including changes in reasoning style and suppression of epistemic expressions (Kim et al., 2026; Pan et al., 2026). In our offline analysis, different-answer and same-answer siblings oppose epistemic-expression tokens at similar rates (72.5% vs. 68.4%), yet differ markedly on annotated error tokens (55.7% vs. 30.0%). Consistent with this confounding effect, directly using different-answer opposition as a penalty produces only a brief early gain, before performance steadily deteriorates below the base model as responses shrink. Second, every sibling is itself incorrect: its support does not show that a target token is correct, and distilling toward its distribution can import another erroneous line of reasoning. The challenge is therefore to use an incorrect reference to extract negative evidence associated with answer disagreement while filtering shared conditioning effects.

In this work, we introduce **SibCredit**, which uses failed siblings as contrasting conditioning contexts without treating any sibling as a correct target. For each target in an all-wrong group, we split its siblings by whether their final answers agree with the target, and use them only as conditioning contexts to re-score the target's own sampled tokens. Same-answer siblings serve as a natural control: they come from the same problem and all-wrong group, are likewise incorrect, and use the same conditioning format. A token is therefore penalized only when different-answer opposition sufficiently exceeds same-answer opposition, thereby suppressing shifts common to both. Because no sibling is correct, we never imitate sibling distributions; we discard sibling-induced probability increases and retain only decreases. The result is a bounded, nonpositive token-level advantage constructed without external correct information and applied only to eligible targets in all-wrong groups; other targets use standard GRPO in our experiments.

Experiments on Qwen3-1.7B and Qwen3-4B across mathematical reasoning and code generation show that SibCredit achieves the highest domain-average accuracy among the evaluated methods in all four model–domain settings. Compared with GRPO, it improves average accuracy by 2.3 and 1.7 percentage points on mathematical reasoning, and by 0.8 and 0.7 points on code generation, respectively. Ablations on mathematical reasoning show that randomizing sibling answer relations or allowing positive auxiliary credit reduces the gains, shuffling token-level credit eliminates the improvement over GRPO, and removing the same-answer control leads to training degradation.

2 PRELIMINARIES

2.1 GROUP-RELATIVE RLVR AND ALL-WRONG GROUPS

For a problem x, RLVR samples rollouts $\mathcal{Y}(x) = \{y_i\}_{i=1}^{G}$ from a behavior policy π_{old} and assigns binary verifier rewards $r_i = R(x, y_i) \in \{0, 1\}$. Methods using within-group reward baselines include GRPO (Shao et al., 2024), RLOO (Ahmadian et al., 2024), DAPO (Yu et al., 2025), Dr. GRPO (Liu et al., 2025), and GSPO (Zheng et al., 2025). Taking GRPO as our running example, its outcome advantage is

$$A_{i,t}^{\text{GRPO}} = A_i = \frac{r_i - \bar{r}}{\sigma_r + \epsilon_{\text{num}}}, \qquad \bar{r} = \frac{1}{G}\sum_{j=1}^{G} r_j, \tag{1}$$

where σ_r is the within-group reward standard deviation. In an *all-wrong group*, every $r_i = 0$, so $A_i = 0$ and verifier rewards provide no policy-gradient signal. The same holds for advantages formed by subtracting group-mean or leave-one-out baselines from these binary rewards.

2.2 CONTEXT-CONDITIONED SAMPLED-TOKEN SCORING

On-policy self-distillation (OPSD) uses additional context to obtain token-level supervision from the same model (Zhao et al., 2026). RLSD (Yang et al., 2026) and RLCSD (Pan et al., 2026) use context-conditioned versus base log-probability gaps for advantage modulation. For a scoring copy π_{sc} and context z, define

$$\Delta_{i,t}(z) = \log \pi_{\text{sc}}(y_{i,t} \mid x, z, y_{i,<t}) - \log \pi_{\text{sc}}(y_{i,t} \mid x, y_{i,<t}). \tag{2}$$

Both terms score the same sampled token at the same target prefix with identical model parameters, varying only the context. Teacher forcing requires no additional trajectory generation; $\Delta_{i,t}(z) < 0$ indicates reduced support under z.

3 SIBCREDIT: LEARNING FROM FAILED SIBLINGS

3.1 OVERVIEW AND DESIGN PRINCIPLES

SibCredit constructs a nonpositive token-level auxiliary advantage from failed siblings when the group-relative outcome advantage vanishes. Consider an all-wrong rollout group $\mathcal{Y}(x) = \{y_i\}_{i=1}^{G}$, where $r_i = 0$ for all i. For a target rollout y_i, the other rollouts $\{y_j : j \neq i\}$ are its siblings, and $a_i = \text{Ans}(x, y_i)$ denotes its task-specific answer representation. Addressing the two challenges identified in Section 1, SibCredit follows two principles: **(1) control for shared conditioning effects** by using same-answer siblings to calibrate different-answer opposition; and **(2) use incorrect siblings conservatively** only as conditioning contexts for re-scoring the target, retaining negative evidence without imitating sibling distributions. Figure 2 summarizes the pipeline.

3.2 SIBLING-CONDITIONED TOKEN SCORING

We partition siblings by answer relation:

$$\mathcal{P}_i^{\text{same}} = \{j \neq i \mid a_j \sim_x a_i\}, \qquad \mathcal{P}_i^{\text{diff}} = \{j \neq i \mid a_j \not\sim_x a_i\}. \tag{3}$$

For mathematics, $\sim_x$ denotes equivalence between normalized answers; for code, a_i is the vector of test outcomes obtained by executing y_i, and $\sim_x$ denotes exact equality of these vectors.

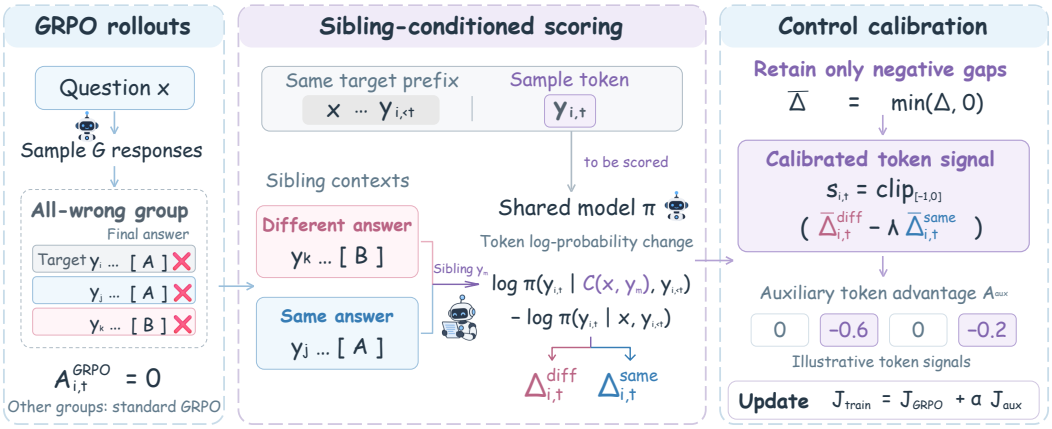

Figure 2: Overview of SibCredit. For an all-wrong rollout group, siblings of a target response are partitioned by answer relation. Each sibling is used only as a conditioning context to re-score the target's sampled tokens under the same target prefix. Different-answer opposition provides candidate negative credit, while same-answer opposition calibrates shifts shared across sibling contexts. The resulting bounded nonpositive signal is used as an auxiliary token-level advantage; all other targets receive only the standard GRPO objective.

The auxiliary signal is activated for a target rollout only when the sampled group is all wrong and both sibling types are available:

$$E_i = \mathbf{1}\left[\left(\sum_{j=1}^{G} r_j = 0\right) \wedge \left(|\mathcal{P}_i^{\text{same}}| > 0\right) \wedge \left(|\mathcal{P}_i^{\text{diff}}| > 0\right)\right]. \tag{4}$$

Across our mathematical reasoning training runs, approximately 75% of all-wrong groups contain at least one eligible target rollout. For each target with $E_i = 1$, we uniformly sample without replacement up to K siblings from each set, yielding $\widetilde{\mathcal{P}}_i^{\text{same}} \subseteq \mathcal{P}_i^{\text{same}}$ and $\widetilde{\mathcal{P}}_i^{\text{diff}} \subseteq \mathcal{P}_i^{\text{diff}}$.

Each sampled sibling supplies context $z_j = (y_j, a_j)$ in Equation 2. Although every sibling is incorrect, we deliberately present it as a correct reference so that the scoring model treats its reasoning as a serious alternative: the resulting support change probes whether the target token remains plausible if that reasoning were accepted. The sibling therefore serves only as a probe, not a supervised target; reduced support is candidate opposition rather than evidence that the target token is necessarily incorrect. Crucially, the same- and different-answer branches use the same reference framing, target prefix, scoring model, and scoring procedure, varying only the sibling content. Thus, generic shifts induced by the framing are shared across branches and are controlled by the same-answer comparison (Appendix C).

The scoring model π_{sc} is a snapshot of the policy. We average the sampled-token log-probability gaps within each branch to reduce sensitivity to individual siblings:

$$\Delta_{i,t}^{b} = \frac{1}{|\widetilde{\mathcal{P}}_i^{b}|} \sum_{j \in \widetilde{\mathcal{P}}_i^{b}} \Delta_{i,t}(z_j), \qquad b \in \{\text{same}, \text{diff}\}. \tag{5}$$

Because both branches share the same base log-probability, subtracting the two branch gaps cancels this term exactly (Appendix A.1). Hence, $\Delta_{i,t}^{\text{diff}} - \Delta_{i,t}^{\text{same}}$ directly compares the average sibling-conditioned support for the same sampled token under different-answer and same-answer contexts. This branch comparison underlies SibCredit's control mechanism; Section 3.3 converts the two gaps into a conservative negative training signal using same-answer opposition as calibration.

3.3 Control-Calibrated Token Signal

The signed gaps in Equation 5 measure how sibling conditioning changes the model's support for a sampled token. Let $d_{i,t}$ and $c_{i,t}$ denote the magnitudes of different-answer and same-answer

opposition, respectively. For an eligible target, we define

$$d_{i,t} = \left(-\Delta^{\text{diff}}_{i,t}\right)_+, \qquad c_{i,t} = \left(-\Delta^{\text{same}}_{i,t}\right)_+,$$
$$s_{i,t} = -\min\left\{\left(d_{i,t} - \lambda c_{i,t}\right)_+, 1\right\}, \qquad \lambda \geq 0, \tag{6}$$

where $(u)_+ = \max\{u, 0\}$. A token receives negative credit if and only if $d_{i,t} > \lambda c_{i,t}$: different-answer opposition must exceed the weighted same-answer control. The cap of 1 bounds the penalty magnitude, ensuring $s_{i,t} \in [-1, 0]$ even for unusually large log-probability gaps. We set $s_{i,t} = 0$ for ineligible rollouts.

Control for shared conditioning effects (Principle 1). Different-answer opposition can reflect both answer disagreement and generic effects of sibling conditioning. We therefore use same-answer opposition as a control for opposition that is not specific to answer disagreement. Section 4.4 examines this distinction using annotated error tokens and epistemic expressions.

The same-answer control can only weaken or remove a candidate penalty; it cannot create or strengthen one. We set $\lambda = 10$, so same-answer opposition only one-tenth as large as different-answer opposition is sufficient to eliminate the penalty. This deliberately conservative choice requires different-answer opposition to substantially exceed the same-answer control before negative credit is assigned. Appendix A.2 gives the corresponding algebraic properties, and Appendix D.4 studies sensitivity to λ.

Use incorrect siblings conservatively. Because every sibling in an all-wrong group is incorrect, SibCredit uses it only as a conditioning context to re-score the target's own sampled token, rather than distilling the full sibling-conditioned vocabulary distribution. Neither increased nor decreased support under an incorrect sibling context alone establishes whether the sampled token is correct. We therefore use these shifts asymmetrically: probability decreases provide candidate opposition, whereas probability increases are discarded rather than converted into positive credit. This asymmetric design is also consistent with recent observations that negative self-distillation signals are more reliable than positive ones (Ding & Zhang, 2026; Li et al., 2026b).

3.4 Training Objective

SibCredit retains the GRPO objective and adds a token-level auxiliary surrogate for eligible rollouts from all-wrong groups. Let $\mathcal{B}$ denote a batch of rollouts sampled from π_{old}, with eligibility determined within each rollout's original problem group. We suppress the problem index for readability. For token $y_{i,t}$, define the importance ratio and auxiliary advantage as

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\text{old}}(y_{i,t} \mid x, y_{i,<t})}, \qquad A^{\text{aux}}_{i,t} = \text{sg}[s_{i,t}], \tag{7}$$

where sg denotes the stop-gradient operator and $A^{\text{aux}}_{i,t} \in [-1, 0]$. Thus, the auxiliary signal provides only nonpositive credit and can remain nonzero even when the GRPO advantage vanishes.

We optimize the auxiliary advantage with the same PPO-style clipped surrogate as GRPO. Both objectives use token-mean normalization: J_{GRPO} averages over all valid response tokens in the batch, whereas J_{aux} averages over valid response tokens in eligible rollouts. Let $N_{\text{aux}} = \sum_{i \in \mathcal{B}} E_i |y_i|$ denote the latter token count. The auxiliary objective is

$$J_{\text{aux}}(\theta) = \mathbb{E}_{\mathcal{B}}\left[\frac{1}{\max\{N_{\text{aux}}, 1\}} \sum_{i \in \mathcal{B}} E_i \sum_{t=1}^{|y_i|} \min\Big\{\rho_{i,t}(\theta) A^{\text{aux}}_{i,t}, \right.$$
$$\left. \text{clip}\left(\rho_{i,t}(\theta), 1 - \epsilon_{\text{clip}}, 1 + \epsilon_{\text{clip}}\right) A^{\text{aux}}_{i,t}\Big\}\right], \tag{8}$$

where $\epsilon_{\text{clip}} > 0$ is the clipping parameter. Normalization over eligible tokens prevents ineligible rollouts from diluting the auxiliary signal. The auxiliary term is zero when no rollout is eligible. Tokens with zero auxiliary advantage remain in the denominator, so the auxiliary contribution reflects both the magnitude and frequency of retained negative signals.

At $\theta = \theta_{\text{old}}$, each token with negative auxiliary advantage contributes a gradient opposing its sampled-token log-probability; Appendix A.3 derives this local effect.

Finally, we jointly maximize

$$J_{\text{train}}(\theta) = J_{\text{GRPO}}(\theta) + \alpha J_{\text{aux}}(\theta), \qquad \alpha \geq 0, \tag{9}$$

where α controls the strength of the auxiliary objective.

For each eligible target, computing the SibCredit signal requires one base scoring pass and up to K sibling-conditioned passes from each branch, for at most $2K + 1$ teacher-forced scoring passes (five with $K = 2$). These passes are required only for eligible targets and introduce no additional trajectory generation.

4 Experiments

We organize our evaluation around three questions: **RQ1:** Does SibCredit improve reasoning performance across model scales and domains? **RQ2:** Which design choices are needed to recover useful supervision from failed siblings? **RQ3:** Does the sibling contrast distinguish erroneous content from shared conditioning effects?

4.1 Experimental Setup

Models and data. We evaluate Qwen3-1.7B and Qwen3-4B (Yang et al., 2025a) on mathematical reasoning and code generation. Following Pan et al. (2026), we train on DeepMath (He et al., 2025) difficulty levels 5–7 and 6–8, respectively. For code, we use TACO-Verified (Li et al., 2023; Li, 2024), retaining problems with at least five test cases following DeepCoder (Luo et al., 2025).

Baselines. We compare with the base model, GRPO (Shao et al., 2024), and token-level self-distillation baselines. RLCSD (Pan et al., 2026) uses correct and incorrect in-group rollouts to construct a token-level modulation of advantages computed from verifier rewards. OPSD (Zhao et al., 2026) and CPO (Xu et al., 2026) use an external correct solution to provide privileged token-level supervision. Thus, in an all-wrong group, RLCSD produces no update, while OPSD and CPO require correct information outside the failed group.

Training. All methods use the same training data and shared optimizer settings. OPSD, RLCSD, CPO, and SibCredit sample $G = 8$ rollouts per problem at temperature 1.0; GRPO uses $G = 12$ to account for the additional scoring cost of self-teacher methods. For Qwen3-1.7B mathematical reasoning, SibCredit uses $0.86\times$ GRPO's model FLOPs per step (Appendix C.6). SibCredit uses up to $K = 2$ siblings per answer-relation set, control weight $\lambda = 10$, and auxiliary weight $\alpha = 0.5$ for mathematics and 0.2 for code. Appendix C provides the complete configurations.

Evaluation. We evaluate mathematical reasoning on AIME24, AIME25 (Zhang & Team Math-AI, 2024; 2025), HMMT25 (Balunovic et al., 2025), and OlympiadBench (He et al., 2024), and code generation on LiveCodeBench v5 (Jain et al., 2025), MBPP+, and HumanEval+ (Liu et al., 2023). We report mean@N, averaging correctness over N independent responses per problem and then over problems. To reduce sampling variance on the small benchmarks, we use $N = 32$ for AIME24, AIME25, and HMMT25, and $N = 2$ for the remaining benchmarks. All methods share the same evaluation protocol, detailed in Appendix C.

4.2 Main Results

Table 1 shows that SibCredit achieves the highest domain-average accuracy in all four model–domain settings. For mathematical reasoning, it improves over GRPO by 2.3 and 1.7 percentage points on Qwen3-1.7B and Qwen3-4B, respectively; for code generation, the corresponding gains are 0.8 and 0.7 points. Notably, despite using no correct reference solution, SibCredit also outperforms both reference-guided baselines, OPSD and CPO, on the domain average in every setting. At the individual-benchmark level, OPSD remains best on MBPP+ for Qwen3-1.7B and LiveCodeBench for Qwen3-4B. Figure 4 in the appendix further compares the training dynamics of

Table 1: Main results on mathematical reasoning and code generation (%). Correct ref. denotes the source of correct privileged context used for token-level supervision: *External* indicates a dataset-provided solution, *In-group* indicates a correct rollout from the sampled group, and *None* indicates that no correct reference is used; "–" indicates not applicable. Avg. denotes the mean across benchmarks within each domain. **Bold** marks the best result for each model, including ties. Olym.: OlympiadBench; LCB: LiveCodeBench.

		Mathematical Reasoning Accuracy					**Code Generation Accuracy**			
Method	Correct ref.	AIME24	AIME25	HMMT25	Olym.	Avg.	LCB v5	MBPP+	HumanEval+	Avg.
Qwen3-1.7B										
Base	–	12.7	10.3	4.9	40.2	17.0	11.4	49.5	61.6	40.8
GRPO	–	13.8	11.3	5.5	42.3	18.2	13.5	49.7	61.9	41.7
OPSD	External	11.0	10.0	4.7	40.7	16.6	11.7	**50.9**	61.0	41.2
RLCSD	In-group	13.2	12.6	5.0	41.7	18.1	13.2	48.9	61.9	41.3
CPO	External	11.4	10.3	4.8	41.3	16.9	13.8	49.7	62.5	42.0
SibCredit	None	**16.7**	**13.9**	**7.1**	**44.3**	**20.5**	**14.1**	49.6	**63.7**	**42.5**
Qwen3-4B										
Base	–	22.5	20.7	11.9	52.0	26.8	20.7	59.4	76.8	52.3
GRPO	–	23.2	21.9	12.1	53.4	27.7	21.9	59.4	77.7	53.0
OPSD	External	22.7	20.8	12.0	52.4	27.0	**24.3**	57.7	77.1	53.0
RLCSD	In-group	23.3	21.8	**13.0**	54.4	28.1	22.8	59.3	77.4	53.1
CPO	External	22.9	22.0	12.0	54.3	27.8	23.1	59.5	77.7	53.4
SibCredit	None	**25.0**	**23.5**	**13.0**	**56.1**	**29.4**	23.4	**59.8**	**78.0**	**53.7**

SibCredit, GRPO, OPSD, and the variant without same-answer control. Overall, these results show that useful supervision can be recovered directly from all-wrong groups without external correct information, with consistent gains across the evaluated model scales and domains.

4.3 Ablation Studies

Table 2 evaluates four design choices on Qwen3-1.7B for mathematical reasoning under the same training and evaluation setup. Variant definitions are provided in Appendix D.2.

Answer-relation structure. Randomizing sibling assignments while preserving branch sizes and target eligibility reduces average accuracy from 20.5 to 18.9. The remaining gain over GRPO (18.2) suggests that failed-sibling conditioning with calibration remains useful, while grouping by answer relation provides additional benefit.

Same-answer control. Removing the same-answer control reduces average accuracy to 14.6, below the base model (17.0), even after approximately matching the effective auxiliary signal strength. The training curves in Appendix D.3 show that this variant initially improves, but the improvement is not sustained: accuracy subsequently declines as responses become shorter. In contrast, full SibCredit maintains longer responses and achieves higher accuracy. Together with the token analysis in Section 4.4, these observations support the role of same-answer calibration in selectively retaining negative evidence, beyond simply reducing penalty magnitude.

Negative-only auxiliary credit. Allowing positive credit from failed siblings reduces average accuracy from 20.5 to 18.8. Because the conditioning siblings are themselves incorrect, increased support for a target token may reflect a shared mistake rather than correctness. This result supports retaining only calibrated negative evidence in this setting.

Token-level localization. Shuffling auxiliary advantages across valid token positions within each rollout while preserving their values reduces average accuracy to 18.0, eliminating the gain over GRPO (18.2). This shows that the gains depend on where penalties are applied, beyond their aggregate strength.

Table 2: Ablation results on Qwen3-1.7B for mathematical reasoning (%). Δ denotes the change in average accuracy relative to full SibCredit.

Method	AIME24	AIME25	HMMT25	Olym.	Avg.	Δ
GRPO	13.8	11.3	5.5	42.3	18.2	−2.3
SibCredit (full)	16.7	13.9	7.1	44.3	**20.5**	–
w/ randomized answer relation	15.2	11.8	6.3	42.3	18.9	−1.6
w/o same-answer control	10.9	8.0	4.2	35.4	14.6	−5.9
w/ positive credit	14.3	11.8	7.2	41.8	18.8	−1.7
w/ shuffled token positions	13.7	11.7	5.2	41.3	18.0	−2.5

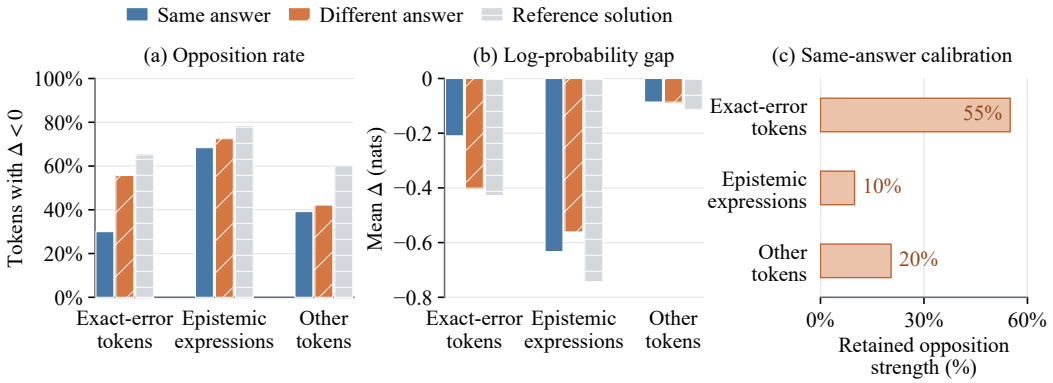

Figure 3: Token-level analysis and same-answer calibration. (a) Fraction of tokens with a negative branch-averaged log-probability gap. (b) Mean signed log-probability gap under each conditioning context. (c) Fraction of different-answer opposition strength retained after same-answer calibration. The reference-solution context in (a–b) is used only for offline comparison. Annotation details are in Appendix B.

4.4 What Does the Signal Capture?

We analyze whether different-answer siblings provide error-related negative evidence, and whether same-answer calibration suppresses conditioning effects shared across sibling contexts. We distinguish annotated exact-error tokens from epistemic-expression tokens identified by a fixed lexicon; the latter provide a lexical proxy for reasoning expression. The analysis covers 60 all-wrong groups and 209 target rollouts. Appendix B describes the sampling and annotation protocol.

Raw sibling opposition contains both error-related and shared effects. Panels (a–b) of Figure 3 compare the signed self-teacher-minus-base gaps under different-answer, same-answer, and reference-solution contexts. On exact-error tokens, different-answer siblings oppose the target more often than same-answer siblings (55.7% vs. 30.0%) and produce a more negative mean gap (-0.401 vs. -0.210). Notably, in this offline sample, different-answer opposition on exact-error tokens is close to that induced by the reference-solution context, despite requiring no correct trajectory. In contrast, epistemic expressions are frequently opposed under both different-answer and same-answer conditioning (72.5% vs. 68.4%). This pattern indicates that different-answer conditioning contains error-related information, but its raw opposition also includes effects shared across sibling contexts, consistent with prior observations that privileged conditioning systematically shifts epistemic expression (Kim et al., 2026; Pan et al., 2026).

Same-answer calibration selectively suppresses shared opposition. Panel (c) of Figure 3 measures how much different-answer opposition strength remains after applying the same-answer control. For each token category, we measure the retained-strength ratio $\mathbb{E}[(d_{i,t} - \lambda c_{i,t})_+]/\mathbb{E}[d_{i,t}]$. Calibration preserves a larger fraction of opposition strength on exact-error tokens than on epistemic expressions or other tokens, supporting selective suppression of shared conditioning effects rather than uniform rescaling.

Sensitivity of signal construction. We further examine how the calibration strength λ and sibling count K affect downstream performance. On Qwen3-1.7B for mathematical reasoning, increasing λ from 1 to 3 to 10 raises average accuracy from 15.8% to 18.8% to 20.5%, compared with 18.2% for GRPO, supporting sufficiently strong same-answer calibration in this setting. Increasing K from 1 to 2 improves average accuracy from 19.8% to 20.5%, while $K = 4$ provides no further gain at the reported precision. Full results are provided in Appendix D.4.

5 Related Work

Learning signals for all-wrong groups. Advantage-modulation methods use self-distillation to redistribute token credit while preserving the verifier-derived update direction; this alone cannot recover updates when the advantage is zero (Yang et al., 2026; Pan et al., 2026). Other methods obtain supervision beyond the current failed group through correct references (Xu et al., 2026; Cai et al., 2026), external teachers (Han et al., 2026), or previously successful trajectories (Luo et al., 2026). Reference-free alternatives assign nonzero advantages to zero-variance groups (Li et al., 2026b) or construct negative token supervision without correct privileged context (Pei et al., 2026; Ding & Zhang, 2026). SibCredit instead derives token-level credit from answer relations among failed siblings within the current group.

Relations among multiple rollouts. Consensus methods use majority answers as pseudo-labels for rewards or distillation (Zuo et al., 2025; Li et al., 2026c), but the majority pseudo-label is necessarily incorrect in an all-wrong voting pool. Shared-prefix analyses identify implicit process credit in GRPO (Sullivan & Koller, 2026), while tree-based methods estimate intermediate values from sampled continuations (Hou et al., 2025; Yang et al., 2025b); their outcome-derived credit provides no contrast when all evaluated continuations receive zero binary reward. Cross-rollout self-distillation conditions on correct siblings, sometimes contrasting them with incorrect siblings (Li et al., 2026a; Pan et al., 2026; Yu et al., 2026). SibCredit uses disagreement among failed siblings as contrastive evidence about sampled tokens, without treating sibling answers as pseudo-labels.

Privileged-conditioning bias and calibration. Privileged context can suppress epistemic expression and degrade reasoning (Kim et al., 2026), and self-distillation gaps can concentrate on style tokens (Pan et al., 2026). Such context-induced likelihood shifts need not reflect token-level action values (Nguyen et al., 2026). RLCSD contrasts correct and incorrect siblings to suppress shared style shifts (Pan et al., 2026); CPO contrasts reference-guided and vanilla distributions for correctness-aware advantage shaping (Xu et al., 2026). Purified OPSD subtracts a reference-only component to form a PMI-based target (Shen et al., 2026), while CSCR downweights tokens sensitive to opposing outcome conditions, preserving the verifier-determined credit direction and budget (He et al., 2026). SibCredit contrasts two types of incorrect siblings: different-answer siblings supply candidate opposition, and same-answer siblings control for shared conditioning effects. It requires no correct context and retains only negative credit.

6 Conclusion

In this work, we show that all-wrong rollout groups can provide useful token-level supervision even when their group-relative outcome advantage vanishes. Our method, SibCredit, uses different-answer siblings to obtain candidate negative evidence and same-answer siblings to control for shared conditioning effects, constructing an auxiliary training signal entirely from the current failed group. Across mathematical reasoning and code generation with Qwen3-1.7B and Qwen3-4B, SibCredit consistently outperforms GRPO and the evaluated self-distillation baselines in domain-average accuracy. Token-level analysis shows that calibration retains a larger fraction of opposition strength on annotated error tokens than on epistemic expressions. These results suggest that the internal structure of failed rollout groups can provide learning signals beyond what is captured by their shared outcome reward. SibCredit currently requires a target rollout to have both same-answer and different-answer siblings within an all-wrong group; when either sibling type is unavailable, no auxiliary signal can be constructed for that target. Extending the method to such groups is a natural direction for future work.

AI USE STATEMENT

In this work, we used generative AI tools to help develop conceptual frameworks, provide feedback on research methodology and experimental design, implement methods, assist with translation, formulate and check mathematical claims and derivations, help interpret experimental results, and support qualitative annotation, including identifying candidate error-token spans for author review. We have not used generative AI tools to propose or refine hypotheses or to clean or reformat datasets; generating synthetic datasets is not applicable to this work. The authors made the final scientific judgments and conclusions based on the experimental results. Additionally, we used generative AI tools to search for and summarize existing literature, edit parts of the manuscript, and improve its structure and readability. The authors verified and tested AI-assisted implementations against the methods described in this paper, manually audited a random subset of AI-assisted error-token annotations, and checked AI-assisted translations and literature summaries against the original sources and our own records. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies SibCredit, a reinforcement learning method that derives token-level supervision from failed rollouts. Although the method aims to improve task performance, higher benchmark accuracy does not establish the safety or reliability of model outputs. Models trained with SibCredit may retain biases and generate incorrect or harmful content, and improved reasoning and code generation capabilities may also be misused. Deployment therefore requires application-specific safety evaluation, appropriate safeguards, and human oversight.

REPRODUCIBILITY STATEMENT

We provide the experimental and implementation details needed to reproduce our results in the appendices. Appendix C documents the training data, verification procedures, optimization and method-specific configurations, evaluation protocol, and compute accounting. Appendix B details the token-level annotation protocol, and Appendix D provides additional ablations, training dynamics, and hyperparameter analyses. The complete SibCredit configuration and sibling-conditioning prompt are also included.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.662.

Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Lewis Tunstall, Carlos Miguel Patiño, Edward Beeching, Aymeric Roucher, Aksel Joonas Reedi, Quentin Gallouédec, Kashif Rasul, Nathan Habib, Clémentine Fourrier, Hynek Kydlicek, Guilherme Penedo, Hugo Larcher, Mathieu Morlon, Vaibhav Srivastav, Joshua Lochner, Xuan-Son Nguyen, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM3: smol, multilingual, long-context reasoner. `https://huggingface.co/blog/smollm3`, 2025.

Mislav Balunovic, Jasper Dekoninck, Ivo Petrov, Nikola Jovanovic, and Martin Vechev. MathArena: Evaluating LLMs on uncontaminated math competitions. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Qiye Cai, Yichuan Ma, Peiji Li, Yongkang Chen, Qipeng Guo, Yicheng Zou, Linyang Li, Xiaocheng Feng, and Bing Qin. H^2SD: Hybrid Hindsight Self-Distillation. *arXiv preprint arXiv:2607.18955*, 2026. URL `https://arxiv.org/abs/2607.18955`.

Guirong Chen, Shuqi Ye, Wenkai Yang, Shiqi Shen, Guangyao Shen, and Yankai Lin. CURE: Critique-driven unified reinforcement learning for test-time self-improvement. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long*

Papers), pp. 28632–28653. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1321. URL https://aclanthology.org/2026.acl-long.1321/.

Yi Ding and Ruqi Zhang. Does on-policy distillation really distill? from noisy teacher to self-improvement. arXiv preprint arXiv:2608.31046, 2026. URL https://arxiv.org/abs/2608.31046.

Zhuowen Han, Jinwei Xiao, Zhengxi Lu, Renren Jin, Zhiyuan Yao, Yuxin Liu, Hongyan Hao, Yueqing Sun, Yu Yang, Qi GU, Xunliang Cai, and Deyi Xiong. Distill Where You Fail: Recovering Learning Signals of Negative RL-Groups from Adaptive Teacher Guidance. *arXiv preprint arXiv:2608.00782*, 2026. URL https://arxiv.org/abs/2608.00782.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.acl-long.211.

Qiangqiang He, Zhongheng Wu, and ZiJian Wang. Not all tokens deserve equal credit: Counterfactual sensitivity credit reallocation for long-CoT reasoning. *arXiv preprint arXiv:2607.27888*, 2026. URL https://arxiv.org/abs/2607.27888.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. DeepMath-103K: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*, 2025.

Zhenyu Hou, Ziniu Hu, Yujiang Li, Rui Lu, Jie Tang, and Yuxiao Dong. TreeRL: LLM reinforcement learning with on-policy tree search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12355–12369, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.acl-long.604. URL https://aclanthology.org/2025.acl-long.604/.

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

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*, 2026.

Hynek Kydlíček. Math-Verify: Math verification library, 2025. URL https://github.com/huggingface/Math-Verify.

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

Kaixin Li. Verified TACO problems. Hugging Face dataset, 2024. URL https://huggingface.co/datasets/likaixin/TACO-verified.

Rongao Li, Jie Fu, Bo-Wen Zhang, Tao Huang, Zhihong Sun, Chen Lyu, Guang Liu, Zhi Jin, and Ge Li. TACO: Topics in algorithmic COde generation dataset. *arXiv preprint arXiv:2312.14852*, 2023.

Yang Li, Gongle Xue, Yijia Guo, Yuheng Yuan, Liwen Hu, and Lei Ma. CAST: Non-privileged clipped asymmetric self-teaching with advantage flipping for GRPO. *arXiv preprint arXiv:2606.00172*, 2026b.

Yijiang Li, Bingyang Wang, Yijun Liang, Yunjie Tian, Di Fu, and Nuno Vasconcelos. On-policy self-distillation without any supervision. *arXiv preprint arXiv:2608.06296*, 2026c.

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

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

Haisen Luo, Yiwei Liu, Haoning Wang, Dan Liu, Junxi Yin, Haotian Wang, Lei Zhang, Xiaoyu Tian, Shuaiting Chen, Yuansheng Song, Baoyan Guo, Xiongfei Yan, Bolan Yang, Chengwei Liu, Ming Cui, and Jiong Chen. DRIFT: Difficulty Routing Self-DIstillation with Rhythm-Gated Exploration and Success BuFfer Training. *arXiv preprint arXiv:2606.30345*, 2026. URL https://arxiv.org/abs/2606.30345.

Michael Luo, Sijun Tan, Roy Huang, Ameen Patel, Alpay Ariyak, Qingyang Wu, Xiaoxiang Shi, Rachel Xin, Colin Cai, Maurice Weber, Ce Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. DeepCoder: A fully open-source 14B coder at O3-mini level, 2025. URL https://pretty-radio-b75.notion.site/DeepCoder-A-Fully-Open-Source-14B-Coder-at-O3-mini-Level-1cf81902c14680b3bee5eb349a512a51. Notion Blog.

Xuan-Phi Nguyen, Zeyu Leo Liu, Yang Li, Shrey Pandit, Yiran Zhao, Anurag Koul, and Shafiq Joty. Privileged likelihood is not automatically value: Three checks for token credit in on-policy self-distillation. *arXiv preprint arXiv:2608.09263*, 2026. URL https://arxiv.org/abs/2608.09263.

Leyi Pan, Shuchang Tao, Yunpeng Zhai, Lingzhe Zhang, Zhaoyang Liu, Bolin Ding, Aiwei Liu, and Lijie Wen. RLCSD: Reinforcement Learning with Contrastive On-Policy Self-Distillation. *arXiv preprint arXiv:2606.11709*, 2026. URL https://arxiv.org/abs/2606.11709.

Rongcan Pei, Zhepei Wei, Shuyao Xu, Xinyu Zhu, Wei-Lin Chen, and Yu Meng. Negative self-distillation: Learning to reason by avoiding flaws. *arXiv preprint arXiv:2609.11699*, 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.

Zhanming Shen, Jintao Tong, Shaotian Yan, Chen Shen, Hao Chen, Wentao Ye, Xiaomeng Hu, Rui Miao, Haobo Wang, Junbo Zhao, Gang Chen, and Jieping Ye. Purified OPSD: On-policy self-distillation without losing how to think. *arXiv preprint arXiv:2607.02234*, 2026. URL https://arxiv.org/abs/2607.02234.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297. ACM, 2025. doi: 10.1145/3689031.3696075.

Michael Sullivan and Alexander Koller. GRPO is secretly a process reward model. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2509.21154.

Weiwen Xu, Jia Liu, Hou Pong Chan, Long Li, Deng Cai, Min Chen, and Hao Zhang. Beyond Entropy: Correctness-Aware Advantage Shaping via Contrastive Policy Optimization. *arXiv preprint arXiv:2607.14614*, 2026. URL https://arxiv.org/abs/2607.14614.

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. *arXiv preprint arXiv:2505.09388*, 2025a. 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. *arXiv preprint arXiv:2604.03128*, 2026. URL https://arxiv.org/abs/2604.03128.

Zhicheng Yang, Zhijiang Guo, Yinya Huang, Xiaodan Liang, Yiwei Wang, and Jing Tang. TreeRPO: Tree relative policy optimization. *arXiv preprint arXiv:2506.05183*, 2025b. URL https://arxiv.org/abs/2506.05183.

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 D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 113222–113244. Curran Associates, Inc., 2025. doi: 10.52202/085713-3775. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a4277440d50f1f15d2cb4c14f7e0c0d2-Paper-Conference.pdf.

Weichen Yu, Xiaomin Li, Yizhou Zhao, Xiaoze Liu, Ruowang Zhang, Haixin Wang, Yinyi Luo, Chen Henry Wu, Gaurav Mittal, Matt Fredrikson, and Yu Hu. Multi-rollout on-policy distillation via peer successes and failures. *arXiv preprint arXiv:2605.12652*, 2026.

Yifan Zhang and Team Math-AI. American Invitational Mathematics Examination (AIME) 2024, 2024. URL https://huggingface.co/datasets/math-ai/aime24.

Yifan Zhang and Team Math-AI. American Invitational Mathematics Examination (AIME) 2025, 2025. URL https://huggingface.co/datasets/math-ai/aime25.

Yubo Zhang, Xinhong Ma, Zezhong Tan, and Ziqiang Dong. I-SDPO: Instance-Level Adaptive Self-Distillation Policy Optimization. *arXiv preprint arXiv:2608.12957*, 2026. URL https://arxiv.org/abs/2608.12957.

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

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group Sequence Policy Optimization. *arXiv preprint arXiv:2507.18071*, 2025. URL https://arxiv.org/abs/2507.18071.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen Zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. TTRL: Test-time reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 38, 2025. doi: 10.52202/085713-4376. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/be690ea16f005c174f6c4102a5970e67-Abstract-Conference.html.

A Additional Analysis of SibCredit

A.1 Interpretation of the Raw Sibling Contrast

The two branch gaps in Equation 5 share the same base log-probability. Their difference therefore cancels the base score exactly:

$$\begin{aligned}\Delta_{i,t}^{\mathrm{diff}} - \Delta_{i,t}^{\mathrm{same}} &= \frac{1}{|\widetilde{\mathcal{P}}_i^{\mathrm{diff}}|} \sum_{j \in \widetilde{\mathcal{P}}_i^{\mathrm{diff}}} \log \pi_{\mathrm{sc}}(y_{i,t} \mid x, z_j, y_{i,<t}) \\ &\quad - \frac{1}{|\widetilde{\mathcal{P}}_i^{\mathrm{same}}|} \sum_{j \in \widetilde{\mathcal{P}}_i^{\mathrm{same}}} \log \pi_{\mathrm{sc}}(y_{i,t} \mid x, z_j, y_{i,<t}) \\ &= \log \frac{\left(\prod_{j \in \widetilde{\mathcal{P}}_i^{\mathrm{diff}}} \pi_{\mathrm{sc}}(y_{i,t} \mid x, z_j, y_{i,<t})\right)^{1/|\widetilde{\mathcal{P}}_i^{\mathrm{diff}}|}}{\left(\prod_{j \in \widetilde{\mathcal{P}}_i^{\mathrm{same}}} \pi_{\mathrm{sc}}(y_{i,t} \mid x, z_j, y_{i,<t})\right)^{1/|\widetilde{\mathcal{P}}_i^{\mathrm{same}}|}}. \end{aligned} \tag{10}$$

Thus, the raw branch contrast is the log ratio of the geometric-mean likelihood assigned to the same sampled token under different-answer and same-answer sibling contexts. A negative value indicates that, on average, different-answer sibling contexts assign lower likelihood to the sampled token than same-answer sibling contexts.

This identity characterizes only the raw signed contrast and is not the signal used for optimization. As described in Section 3.3, SibCredit first truncates each branch to retain only opposition, then uses the same-answer branch as a weighted control, and finally clips the calibrated signal.

A.2 Properties of the Control-Calibrated Signal

Using the opposition magnitudes $d_{i,t}$ and $c_{i,t}$ defined in Equation 6, the penalty magnitude is

$$-s_{i,t} = \min\left\{(d_{i,t} - \lambda c_{i,t})_+ , 1\right\}.$$

This form makes several properties immediate. First,

$$s_{i,t} < 0 \quad \Longleftrightarrow \quad d_{i,t} > \lambda c_{i,t},$$

so negative credit is retained exactly when different-answer opposition exceeds the λ-weighted same-answer control.

Second, because $\lambda c_{i,t} \geq 0$,

$$(d_{i,t} - \lambda c_{i,t})_+ \leq d_{i,t},$$

and therefore

$$-s_{i,t} \leq \min\{d_{i,t}, 1\}.$$

Thus, relative to using different-answer opposition alone, the same-answer branch cannot increase the magnitude of a negative signal: it can only weaken or remove the candidate penalty.

Third, for fixed $d_{i,t}$ and $c_{i,t}$, the quantity $(d_{i,t} - \lambda c_{i,t})_+$ is nonincreasing in λ. Hence, increasing λ can only move $s_{i,t}$ toward zero. When $c_{i,t} = 0$, the signal reduces to the clipped different-answer penalty.

Finally, when $\lambda \geq 1$,

$$(d_{i,t} - \lambda c_{i,t})_+ \leq (d_{i,t} - c_{i,t})_+ ,$$

which gives

$$0 \leq -s_{i,t} \leq \min\left\{(d_{i,t} - c_{i,t})_+ , 1\right\}.$$

Therefore, if $d_{i,t} \leq c_{i,t}$, and in particular if $d_{i,t} = c_{i,t}$, the calibrated signal is zero. Thus, for $\lambda \geq 1$, opposition that is no stronger under different-answer conditioning than under same-answer conditioning cannot produce negative credit.

By construction, $s_{i,t} \in [-1, 0]$.

A.3 Local Effect of the Auxiliary Advantage

We characterize the direct local effect of the auxiliary objective on a sampled token. Fix an eligible token (i,t) and condition on the sampled group. At $\theta = \theta_{\text{old}}$,

$$\rho_{i,t}(\theta_{\text{old}}) = \frac{\pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i,<t})}{\pi_{\text{old}}(y_{i,t} \mid x, y_{i,<t})} = 1.$$

Since 1 lies in the interior of the PPO clipping interval, the clipped and unclipped surrogate contributions coincide in a neighborhood of θ_{old}. Moreover, $A^{\text{aux}}_{i,t}$ is treated as a stop-gradient quantity. Therefore, up to the positive normalization factor in Equation 8, the token's contribution to the auxiliary gradient is

$$\nabla_\theta \left[\rho_{i,t}(\theta) A^{\text{aux}}_{i,t}\right]\big|_{\theta=\theta_{\text{old}}} = A^{\text{aux}}_{i,t} \nabla_\theta \rho_{i,t}(\theta)|_{\theta=\theta_{\text{old}}} \tag{11}$$

$$= A^{\text{aux}}_{i,t} \nabla_\theta \log \pi_\theta(y_{i,t} \mid x, y_{i,<t})\big|_{\theta=\theta_{\text{old}}}. \tag{12}$$

Let

$$g_{i,t} = \nabla_\theta \log \pi_\theta(y_{i,t} \mid x, y_{i,<t})\big|_{\theta=\theta_{\text{old}}},$$

and consider an infinitesimal gradient-ascent step induced by this token alone,

$$\theta(\gamma) = \theta_{\text{old}} + \gamma A^{\text{aux}}_{i,t} g_{i,t}, \qquad \gamma \geq 0,$$

where any positive normalization factor can be absorbed into γ. Then

$$\frac{d}{d\gamma} \log \pi_{\theta(\gamma)}(y_{i,t} \mid x, y_{i,<t})\bigg|_{\gamma=0} = g_{i,t}^\top A^{\text{aux}}_{i,t} g_{i,t} \tag{13}$$

$$= A^{\text{aux}}_{i,t} \|g_{i,t}\|_2^2 \leq 0. \tag{14}$$

Because $A^{\text{aux}}_{i,t} \in [-1, 0]$, the direct first-order contribution is nonpositive and is strictly negative whenever $A^{\text{aux}}_{i,t} < 0$ and $g_{i,t} \neq 0$. Thus, a retained negative auxiliary advantage locally pushes down the log-probability of the sampled token.

This statement concerns the direct contribution of a single token. The complete update aggregates the GRPO gradient and auxiliary gradients from many tokens and rollouts. For the full training direction $d_{\text{train}} = \nabla_\theta J_{\text{train}}(\theta_{\text{old}})$, the corresponding first-order change is

$$\frac{d}{d\gamma} \log \pi_{\theta_{\text{old}} + \gamma d_{\text{train}}}(y_{i,t} \mid x, y_{i,<t})\bigg|_{\gamma=0} = g_{i,t}^\top d_{\text{train}},$$

whose sign is not determined by $A^{\text{aux}}_{i,t}$ alone. Hence, Equation 14 characterizes the local effect of the selected token's own auxiliary-gradient contribution, rather than a monotonic guarantee on its probability after the complete optimization step.

B Token Annotation Protocol

Sample construction. We randomly sample problems from DeepMath and generate rollout groups using Qwen3-4B until we collect 60 all-wrong groups that contain at least one eligible target rollout. A target is eligible if both same-answer and different-answer siblings are available within its group. We annotate up to four eligible targets per group, yielding 209 target rollouts in total. The scoring procedure and the number of sampled siblings follow the same settings as in the main experiments. These annotations support the offline diagnostic analysis in Figure 3 and are not used as training supervision.

Exact-error tokens. We annotate tokens that directly express an erroneous claim, value, or formula in a statement contributing to the incorrect final answer. We include all independent errors satisfying this criterion. Annotation is localized to the content carrying the error, rather than the entire surrounding reasoning step. For example, when a solution incorrectly assumes that a region is triangular, the word "triangle" carries the erroneous claim. GPT-5.6 Sol first identifies candidate spans using task instructions and illustrative examples, then performs two additional rounds of review to check and revise its annotations. On a random subset of target rollouts, author review confirmed 95% of the AI-annotated exact-error spans.

Table 3: Shared training hyperparameters.

Hyperparameter	Value
Framework	`veRL` 0.8.0.dev0
Thinking mode	Disabled
Learning rate	1×10^{-6}
LR warmup	50 steps
Global batch size	64
PPO mini-batch size	32
PPO clipping parameter ϵ_{clip}	0.2
KL coefficient	0
Group size G	12 (GRPO), 8 (other methods)
Training temperature	1.0
Top-p / Top-k	0.95 / 20
Maximum prompt length	2,048
Maximum response length	4,096

Epistemic-expression tokens. We identify epistemic expressions using a fixed lexicon of 21 words associated with exploration, verification, self-correction, and uncertainty: *wait, reconsider, oops, however, try, check, verify, consider, suppose, assume, maybe, perhaps, possibly, likely, seem, seems, guess, might, may, instead, alternatively*. This category provides a lexical proxy for reasoning expression; it does not imply that opposition to these tokens necessarily harms reasoning performance.

C Implementation Details

C.1 Training Data and Verification

Mathematical reasoning. We train Qwen3-1.7B and Qwen3-4B in non-thinking mode on DeepMath (He et al., 2025). Following Pan et al. (2026), we use difficulty levels 5–7 and 6–8 for the two models, respectively. Following CURE (Chen et al., 2026), we remove binary-answer problems to reduce spurious rewards from random guessing. Both models are trained for one epoch, corresponding to 600 and 500 optimization steps, respectively. We use Math-Verify (Kydlíček, 2025) for answer extraction and mathematical-equivalence checking.

Code generation. We train on TACO-Verified (Li et al., 2023; Li, 2024), retaining problems with at least five test cases following the filtering criterion used by DeepCoder (Luo et al., 2025). Both Qwen3-1.7B and Qwen3-4B are trained for 200 steps. Generated programs are executed in the sandbox provided by `veRL` and receive reward 1 only if they pass all selected tests. For code, we define a rollout's task-specific answer representation as its test-outcome vector over the selected tests. This behavior-based representation avoids distinguishing programs solely on the basis of variable naming, formatting, or other surface-level differences. When constructing SibCredit's answer-relation sets, two code rollouts are considered equivalent if and only if their representations are identical.

C.2 Optimization and Hyperparameters

All experiments are implemented with `verl` version `0.8.0.dev0` (Sheng et al., 2025), with `vLLM` for rollout generation and evaluation. All training experiments are conducted on four NVIDIA A800-SXM4 GPUs, each with 80 GB of memory. We disable thinking mode throughout training and evaluation. We use a global training batch size of 64, and a PPO mini-batch size of 32. All remaining optimizer settings follow the defaults of `verl`.

For rollout generation, self-teacher methods—OPSD, RLCSD, CPO, and SibCredit—use $G{=}8$ responses per problem, while GRPO uses $G{=}12$ to approximately compensate for the additional scoring forward passes required by self-teacher methods. Training rollouts are sampled with temperature 1.0, top-$p = 0.95$, and top-$k = 20$. The maximum prompt and response lengths are 2,048 and 4,096 tokens, respectively. Table 3 summarizes the shared settings.

C.3 Baseline Implementations

All training-based baselines use the same training data, shared optimization settings, and rollout configuration described above. For method-specific hyperparameters, we follow the configurations recommended in the original papers unless otherwise stated.

GRPO. We implement GRPO (Shao et al., 2024) using the standard group-relative advantage and PPO-style clipped objective in `veRL`.

OPSD. We follow the OPSD implementation used by RLCSD (Pan et al., 2026). Specifically, we use dense forward-KL distillation, pointwise vocabulary-level clipping with threshold 0.05, and a fixed teacher initialized from the pretrained policy. Following that implementation, the distillation target is computed over the teacher's top-100 vocabulary entries, with the remaining probability mass aggregated into a tail bucket. The privileged context is constructed from the reference solution provided by the training dataset.

RLCSD. We follow the original RLCSD configuration (Pan et al., 2026), using K=4, τ=0.02, λ=0.5, δ=0.02, and η=1.0. The self-teacher is implemented as a snapshot of the policy and synchronized every 10 optimization steps.

CPO. We implement CPO (Xu et al., 2026) following its reference-guided contrastive advantage-shaping formulation. Under our shared training configuration, the original setting $\alpha^+ = \alpha^- = 0.025$ resulted in unstable training. We therefore set $\alpha^+ = \alpha^- = 0.015$ for all CPO experiments. The reference-guided context uses the reference solution provided by the training dataset. All remaining rollout and optimization settings follow the shared configuration described above.

C.4 SibCredit Implementation Details

For SibCredit, the self-teacher is a snapshot of the current policy synchronized every 10 optimization steps. For each eligible target, we uniformly sample without replacement up to K=2 siblings from the same-answer set and up to K=2 siblings from the different-answer set. If a set contains fewer than K siblings, all available siblings are used.

We set the same-answer control weight to λ=10 in all experiments. The auxiliary objective weight is α=0.5 for mathematical reasoning and α=0.2 for code generation. As defined in Section 3, the calibrated token signal is applied only to all-wrong targets for which both same-answer and different-answer siblings are available. All other targets receive no auxiliary SibCredit advantage and follow the standard GRPO objective.

Sibling-Conditioned Scoring Prompt

```
Problem: {problem}

Here is a reference solution to this problem:
=== Reference Solution Begin ===
{sibling_rollout}

Correct final answer: {sibling_answer}
=== Reference Solution End ===

After reading the reference solution above, make sure you understand
the reasoning behind each step.

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

Sibling-conditioning prompt. We use exactly the same prompt template for same-answer and different-answer sibling conditioning; the two branches differ only in the selected sibling. The placeholders {`problem`}, {`sibling_rollout`}, and {`sibling_answer`} denote the original

problem, the selected sibling's complete rollout, and its extracted final answer, respectively. Although every sibling used by SibCredit is verifier-confirmed incorrect, we deliberately present it to the self-teacher using the same reference-solution framing, including the "Correct final answer" label. This keeps the conditioning format identical across sibling types and makes the sibling content itself the only branch-specific input.

C.5 Evaluation Protocol

All models are evaluated in non-thinking mode using `vLLM`. We sample with temperature 0.6, top-p=0.95, top-k=20, and a maximum generation length of 8,192 tokens for all benchmarks. The same decoding and evaluation protocol is applied to the base models and all training-based methods.

For mathematical reasoning, we evaluate on AIME24, AIME25 (Zhang & Team Math-AI, 2024; 2025), HMMT25 (Balunovic et al., 2025), and OlympiadBench (He et al., 2024). For code generation, we evaluate on LiveCodeBench v5 (Jain et al., 2025), MBPP+, and HumanEval+ (Liu et al., 2023). MBPP+ and HumanEval+ are evaluated following the official EvalPlus protocol using their execution-based test suites; LiveCodeBench v5 is also evaluated by execution-based correctness.

We report mean@N. For each problem, we independently sample N responses, compute their mean binary correctness, and then average this quantity over all problems in the benchmark. We use N=32 for AIME24, AIME25, and HMMT25 to reduce sampling variance on these small competition benchmarks, and N=2 for OlympiadBench, LiveCodeBench v5, MBPP+, and HumanEval+.

C.6 Training FLOPs

We compare the average FLOPs per training step of SibCredit and GRPO in the Qwen3-1.7B and Qwen3-4B experiments on DeepMath. Both methods use 64 problems per step, with $G = 8$ rollouts per problem for SibCredit and $G = 12$ for GRPO.

We calculate FLOPs from the model architecture and the recorded sequence lengths, counting a multiplication and an addition as two floating-point operations. The accounting includes rollout generation, old-policy log-probability computation, policy forward and backward passes, and all additional sibling-scoring forward passes:

$$F_{\text{step}} = F_{\text{rollout}} + F_{\text{old}} + F_{\text{update}} + F_{\text{score}}. \tag{15}$$

The calculation accounts for attention, output projections, and activation recomputation during training. Standard autoregressive KV caching is included when computing rollout-generation FLOPs, while we do not assume additional savings from cross-request or shared-prefix caching. For SibCredit, F_{score} accounts for the complete conditioning contexts and target responses processed by the scoring model. The same accounting convention is applied to both methods.

Table 4: Average FLOPs per training step on DeepMath. One PFLOP denotes 10^{15} floating-point operations. Relative FLOPs are normalized to GRPO within each model size.

Model	Method	Rollouts per problem	PFLOPs per step	Relative FLOPs
Qwen3-1.7B	GRPO	12	26.15	1.00
	SibCredit	8	22.41	0.86
Qwen3-4B	GRPO	12	66.830	1.00
	SibCredit	8	53.342	0.80

As shown in Table 4, under the chosen group sizes, SibCredit uses approximately 14.3% and 20.2% fewer FLOPs per step than GRPO for Qwen3-1.7B and Qwen3-4B, respectively. Thus, these comparisons do not give SibCredit a larger per-step training-compute budget than GRPO.

D Additional Experimental Results

D.1 Results on an Additional Backbone

To examine whether the gains extend beyond the Qwen3 family, we evaluate SibCredit on SmolLM3-3B (Bakouch et al., 2025) for mathematical reasoning. We use the same training data as for Qwen3-1.7B and compare SibCredit with the base model and GRPO.

Table 5: Mathematical reasoning results on SmolLM3-3B (%).

Method	AIME24	AIME25	HMMT25	OlympiadBench	Avg.
Base	8.54	9.38	3.33	35.16	14.10
GRPO	9.38	11.04	**4.06**	36.42	15.23
SibCredit (Ours)	**10.00**	**13.02**	**4.06**	**37.24**	**16.08**

As shown in Table 5, SibCredit improves average accuracy over GRPO and the base model by 0.85 and 1.98 percentage points, respectively. Together with the Qwen3 results, these results provide evidence that SibCredit's gains in mathematical reasoning extend to an additional model family.

D.2 Additional Ablation Details

This section details the variants in Section 4.3 and complements Table 2 with training dynamics. All ablations use Qwen3-1.7B for mathematical reasoning under the same training and evaluation setup as the full method, with the changes described below.

Randomized answer relation. We randomly reassign failed siblings to the two conditioning branches while preserving the original branch sizes and target eligibility. All remaining settings are unchanged. Average accuracy decreases from 20.5 to 18.9, but remains above GRPO (18.2). One possible explanation is that randomization preserves the two-branch calibration mechanism. Conditioning effects shared across failed siblings, including opposition to epistemic expressions, can occur in both random branches and thus still be suppressed. In contrast, opposition to error tokens can vary with the specific reasoning content of each sibling, allowing some useful negative evidence to survive calibration even without answer-based grouping. However, random assignment weakens the distinction between candidate error-related opposition and shared conditioning effects. This may explain the residual gain over GRPO and the additional benefit of grouping siblings by answer relation.

Without same-answer control. The *Different-answer only* variant removes the same-answer branch. We set its auxiliary weight to $\alpha = 0.1$ to approximately match the full method's effective signal strength, measured as α times the mean absolute auxiliary advantage over tokens included in the auxiliary objective. Reducing the auxiliary weight weakens all penalties uniformly, whereas same-answer calibration selectively suppresses opposition shared across sibling contexts. Without this control, penalties on epistemic expressions and other shared conditioning effects can remain alongside error-related penalties. The training dynamics in Figure 4 show an initial accuracy gain followed by deterioration as responses shorten. Together with the token-level analysis in Figure 3, these results suggest that the control contributes through selective calibration beyond simply reducing the overall penalty magnitude.

With positive credit. The full method retains only sibling-induced decreases in the target token probability and discards increases. This variant additionally treats probability increases as candidate positive evidence while keeping the same answer-relation structure and calibration mechanism. The added positive component is clipped to $[0, 1]$, while the original negative component is retained. The variant reaches 18.8 average accuracy, compared with 20.5 for full SibCredit and 18.2 for GRPO. It scores slightly higher on HMMT25 (7.2 vs. 7.1) but lower on the other three benchmarks.

Shuffled token positions. We randomly permute the full method's auxiliary advantages across valid token positions within each rollout. This preserves the advantage values, the number of penalized tokens, and the total penalty magnitude within each rollout, while breaking their alignment with

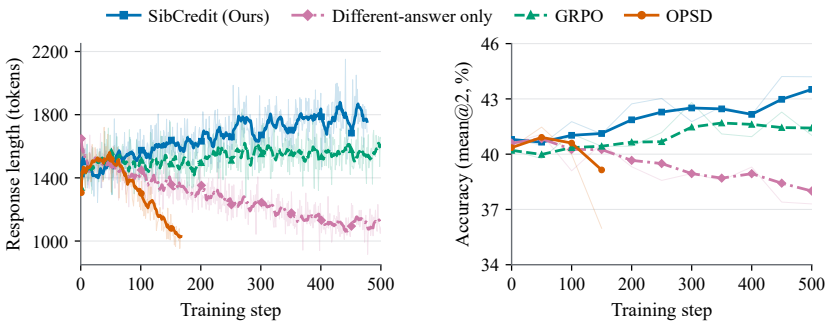

Figure 4: Training dynamics of response length (left) and OlympiadBench accuracy (right) on Qwen3-1.7B. SibCredit gradually generates longer responses and achieves higher accuracy than GRPO, whereas OPSD and Different-answer only exhibit response shortening accompanied by performance degradation. Faint curves denote raw measurements and bold curves denote trailing averages.

Table 6: Sensitivity to the control weight λ and sibling count K on Qwen3-1.7B (%). Avg. is the mean across the four benchmarks. The default setting is $\lambda = 10$, $K = 2$.

Setting	AIME24	AIME25	HMMT25	Olympiad	Avg.
Control weight ($K = 2$)					
$\lambda = 1$	11.3	10.3	5.1	36.5	15.8
$\lambda = 3$	15.1	11.9	5.8	42.3	18.8
$\lambda = 10$	16.7	13.9	7.1	44.3	20.5
Sibling count ($\lambda = 10$)					
$K = 1$	16.0	13.7	6.6	43.0	19.8
$K = 2$	16.7	13.9	7.1	44.3	20.5
$K = 4$	16.7	14.0	6.9	44.4	20.5

the tokens originally scored. Average accuracy decreases from 20.5 to 18.0, compared with 18.2 for GRPO. This result supports the importance of token-level localization: preserving the aggregate penalty alone does not preserve the benefit of SibCredit.

D.3 Training Dynamics

Figure 4 compares the training dynamics of SibCredit, GRPO, OPSD, and the Different-answer only variant on Qwen3-1.7B for mathematical reasoning. The intermediate-checkpoint OlympiadBench evaluations are performed post hoc solely for diagnostic analysis. GRPO maintains a relatively stable response length and achieves modest accuracy gains. In contrast, SibCredit gradually produces longer responses while attaining higher OlympiadBench accuracy. This pattern is consistent with allowing more room for multi-step reasoning, although response length alone does not establish reasoning quality. OPSD initially improves but subsequently exhibits pronounced response-length shrinkage accompanied by declining accuracy in our setup.

The same-answer-control ablation further highlights the role of calibration. Despite approximate signal-strength matching, Different-answer only improves early in training but subsequently deteriorates as its responses become shorter, resembling the trajectory of OPSD. Its final average accuracy across the four mathematical reasoning benchmarks is 14.6, below the pre-RL base model (17.0). Together, these observations suggest that the same-answer control helps mitigate response shortening and sustain performance gains during training.

D.4 Hyperparameter Analysis

We examine the control weight λ and the maximum number of sampled siblings K on Qwen3-1.7B for mathematical reasoning. Table 6 reports the full results.

Control weight. Performance improves consistently as λ increases over the tested range. In particular, the weak-control setting $\lambda = 1$ underperforms GRPO, whereas stronger calibration substantially improves performance. These results support using sufficiently strong same-answer calibration in the tested setting.

Sibling count. We use $K = 2$ throughout the main experiments to limit the additional sibling-scoring cost. As a post-hoc sensitivity analysis, we vary $K \in \{1, 2, 4\}$ while keeping all other settings fixed. Increasing K from 1 to 2 improves performance, whereas $K = 4$ yields essentially no additional gain. The average accuracy is unchanged at the reported precision when increasing K from 2 to 4.