STEP AHEAD, DISTILL BACK: ONE-STEP RL-GUIDED ON-POLICY DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy distillation (OPD) offers dense supervision for reasoning models, but its use with reward-trained experts commonly follows a train-then-distill workflow. We show that effective guidance can instead come from a teacher only one reward-gradient step ahead of the student. Our method, On-Policy Gradient Distillation (OPGD), rebuilds this teacher from the current actor at every iteration, allowing it to inherit accumulated learning while adapting to fresh verified outcomes. The teacher–actor log ratio isolates the preferences induced by this adaptation and supplies signed token advantages for OPD on the same rollouts. A first-order analysis connects these preferences to interactions between token score gradients, explaining how response-level feedback is redistributed within and across responses. Distillation reweights the outcome gradient through the policy's local geometry while preserving nonnegative first-order alignment. On Qwen3-4B, OPGD improves mean Pass@1 over GRPO by 9.2 percentage points across four mathematical benchmarks at a common 100-update budget. Separate training on coding prompts also yields faster learning and higher accuracy across three code-generation benchmarks.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has driven substantial progress in mathematical reasoning and code generation (Shao et al., 2024; Guo et al., 2025). Optimizing verifiable outcomes allows models to develop reasoning behavior from their own sampled trajectories without annotations of intermediate steps. The resulting policies can also serve as teachers, transferring the benefits of reward training to other models through distillation (Guo et al., 2025). Access to these experts' conditional distributions enables token-level supervision, carrying their reward-adapted preferences into the student's learning objective (Yang et al., 2026).

On-policy distillation (OPD) uses these conditional preferences on student-generated trajectories, keeping supervision aligned with the student's evolving rollout distribution (Agarwal et al., 2024; Gu et al., 2024). In a train-then-distill workflow, this feedback is supplied by an expert whose reward adaptation is already complete. An alternative is to obtain teacher–student preferences from incremental reward adaptation of the student itself, refreshing the teacher throughout learning. This raises the question of whether a teacher only one reward-gradient step ahead can already provide effective distillation supervision.

We propose On-Policy Gradient Distillation (OPGD), which couples reward adaptation and OPD through a teacher only one outcome-based policy-gradient step ahead of the actor. Reconstructed from the current actor at every iteration, the teacher inherits its accumulated learning, allowing a single adaptation step to supply fresh guidance throughout training. The teacher–actor log ratio isolates the preferences induced by the latest verified outcomes and serves as a signed token advantage for OPD on the same rollouts.

Distilling these preferences transforms the outcome gradient according to the actor's local geometry. Our first-order analysis identifies how token score similarities redistribute outcome feedback across contexts, yielding different supervision within a response and transferring feedback between responses. Fitting these preferences determines a new update that retains nonnegative first-order alignment with the outcome gradient. Experiments on Qwen3-4B show that OPGD improves mean Pass@1 over GRPO by 9.2 percentage points across four mathematical benchmarks at the same

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.

100-update budget. Separate code-generation experiments likewise show faster learning and higher accuracy.

In summary, our contributions are:

- We introduce a one-step teacher construction that converts outcome supervision into contextual OPD advantages, enabling distillation within the actor's reward training without an external expert.
- We derive a local kernel interpretation of these advantages and their induced update, explaining cross-response credit transfer and nonnegative first-order alignment with the outcome gradient.
- We validate OPGD on four mathematical reasoning benchmarks with Qwen3-4B and on three code-generation benchmarks using separate coding training, demonstrating faster learning and higher accuracy than GRPO.

2 Related Work

On-policy distillation and reward adaptation. Training on student-generated trajectories addresses the distribution mismatch that arises when sequence distillation relies on teacher-generated responses (Agarwal et al., 2024). The teacher then supervises prefixes encountered by the student, while the choice of divergence determines how its preferences enter the update. MiniLLM develops reverse-KL distillation through policy-gradient optimization (Gu et al., 2024). For reasoning models, G-OPD uses an immediate token-level log-ratio advantage and interprets distillation as optimizing the teacher's implicit reward (Yang et al., 2026). This formulation connects policy matching to reward maximization and permits reward extrapolation, allowing a student to learn beyond the teacher's performance rather than treating imitation as the endpoint. OPD^2 focuses the reward on changes acquired during post-training by comparing the teacher with its own pretrained base (Heo et al., 2026). It centers these delta advantages under the student distribution and retains them where their signs agree with the ordinary teacher–student advantages. The reference therefore isolates what the teacher acquired, while the student still determines which changes enter distillation. Empirical studies likewise find that teacher accuracy alone does not determine transfer: compatibility with the student and capabilities acquired through further teacher training affect the usefulness of its feedback (Li et al., 2026b).

Outcome supervision and contextual credit. Policy ratios also connect distillation to dense feedback learned from coarse supervision. DPO parameterizes preference-derived rewards through a policy–reference log ratio (Rafailov et al., 2023), and its token-MDP formulation relates these ratios to sequential credit assignment (Rafailov et al., 2024). With outcome labels, Yuan et al. (2025) show that the same reward parameterization yields an implicit process reward model without step annotations. PRIME makes this construction part of online RL, updating the implicit reward model on verified rollouts and combining returns from its token rewards with the outcome advantage (Cui et al., 2025). Process supervision instead learns step correctness directly from annotations (Lightman et al., 2024), whereas VinePPO estimates prefix values by sampling additional continuations, replacing learned value predictions with Monte Carlo estimates (Kazemnejad et al., 2024). With only the original rollouts, KTAE supplements the response advantage with token-type associations estimated from successful and unsuccessful responses (Sun et al., 2025), while entropy-based selection concentrates updates on uncertain positions where alternative continuations remain plausible (Wang et al., 2025). OPGD derives its signed coefficients from the actor's response to the verified outcomes, making the feedback sensitive to each token's prefix without a persistently trained process model or additional continuation sampling.

Self-distillation and teacher construction. Teacher-free formulations replace external feedback with advantages based on the student's probabilities and entropy (Ding & Zhang, 2026). Context-based self-distillation instead gives the teacher information unavailable in the student's original prompt. Reference solutions (Zhao et al., 2026), successful attempts, and environment feedback (Hübotter et al., 2026) can improve the model's conditional predictions, which are distilled back into its behavior under the original prompt. RISE obtains a teacher through reward adaptation, extrapolating an RLVR-updated policy relative to an earlier or averaged anchor, then distilling the extrapolated policy into the actor after its RL update (Li et al., 2026a). RISE therefore retains the

RLVR step and adds distillation, whereas OPGD uses reward adaptation only to construct actor-relative token advantages for its retained distillation update.

3 PRELIMINARIES

Let $\pi_\theta(x \mid q)$ be an autoregressive policy and $R(q, x)$ a terminal verifier reward. For response x_i, the token prefix is $c_{it} = (q_i, x_{i,<t})$. At iteration k, rollouts are sampled from π_{θ_k}. Their prefixes remain fixed during optimization. The valid response positions are indexed by $\mathcal{K}$, with $\mathcal{K}_i$ denoting those in response i.

3.1 GROUP-RELATIVE POLICY OPTIMIZATION

GRPO (Shao et al., 2024) uses G responses per prompt to compute group-normalized outcome advantages:

$$A_i = \frac{R_i - \bar{R}_{q_i}}{\sigma_{q_i} + \varepsilon}, \qquad \bar{R}_q = \frac{1}{G} \sum_{j:q_j=q} R_j. \tag{1}$$

Here $R_i = R(q_i, x_i)$, σ_q is the group reward standard deviation, and $\varepsilon > 0$. At the rollout policy, with inactive ratio clipping and no reference-policy penalty, the outcome gradient is

$$\boldsymbol{s}_{it} = \nabla_\theta \log \pi_{\theta_k}(x_{it} \mid c_{it}), \qquad g_k = \sum_{(i,t)\in\mathcal{K}} w_{it} A_i \boldsymbol{s}_{it}. \tag{2}$$

The fixed nonnegative weights w_{it} include masking, loss normalization, and any fixed importance correction. The response-level coefficient A_i is shared across positions, whereas the token scores $\boldsymbol{s}_{it}$ depend on the sampled prefixes.

3.2 ON-POLICY DISTILLATION

OPD evaluates a teacher on student-generated prefixes (Agarwal et al., 2024; Gu et al., 2024). We use the token-local reverse-KL formulation, with the prefix distribution $\mathcal{C}_k$ held fixed:

$$\mathcal{L}_{\mathrm{KL}}(\theta; \phi) = \mathbb{E}_{c\sim\mathcal{C}_k} \left[\mathrm{KL}(\pi_\theta(\cdot \mid c) \,\|\, \pi_\phi(\cdot \mid c)) \right]. \tag{3}$$

For a fixed teacher, its gradient at θ_k admits the sampled-token advantage form

$$\begin{aligned} A^{\phi}_{\mathrm{OPD}}(c, a) &= \log \pi_\phi(a \mid c) - \log \pi_{\theta_k}(a \mid c), \\ -\nabla_\theta \mathcal{L}_{\mathrm{KL}}\big|_{\theta_k} &= \mathbb{E}_{c,a}\Big[A^{\phi}_{\mathrm{OPD}}(c, a) \nabla_\theta \log \pi_{\theta_k}(a \mid c) \Big], \end{aligned} \tag{4}$$

with $a \sim \pi_{\theta_k}(\cdot \mid c)$. This immediate log-ratio advantage is the token-level OPD signal used by Yang et al. (2026). OPGD instantiates it with a reward-updated teacher and freezes the resulting advantages in an empirical loss on the current rollouts.

4 METHOD

Figure 1 contrasts the three paradigms. RLVR scores a response with a sequence-level outcome advantage, OPD distills a token-level signal from a separate teacher, and OPGD keeps that token-level signal but builds the teacher from one virtual RLVR step on the actor's own verified rollouts.

4.1 LEARNING FROM A ONE-STEP TEACHER

Starting from the current actor θ_k, we construct its teacher with one SGD step on the outcome objective:

$$\theta^E_k = \theta_k + \alpha g_k. \tag{5}$$

The teacher is adapted on the actor's own verified rollouts, and the actor serves as both its student and its pre-adaptation reference. The OPD advantage therefore coincides with the teacher's reward-adaptation delta,

$$d_{it} = \log \pi_{\theta^E_k}(x_{it} \mid c_{it}) - \log \pi_{\theta_k}(x_{it} \mid c_{it}). \tag{6}$$

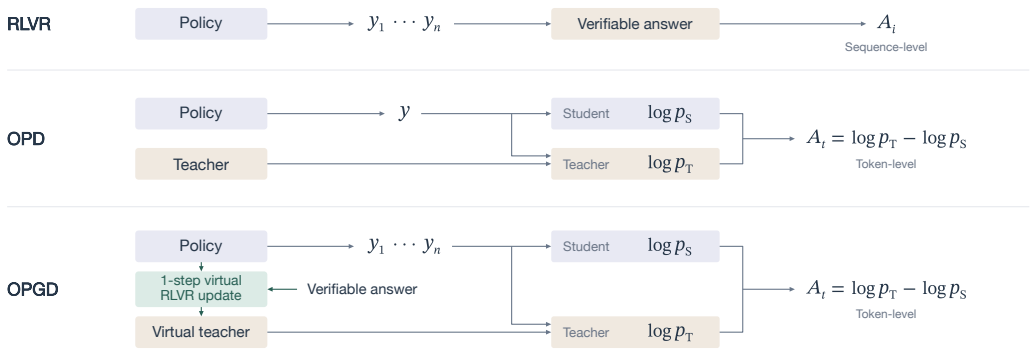


Figure 1: Comparison between RLVR, OPD, and OPGD.

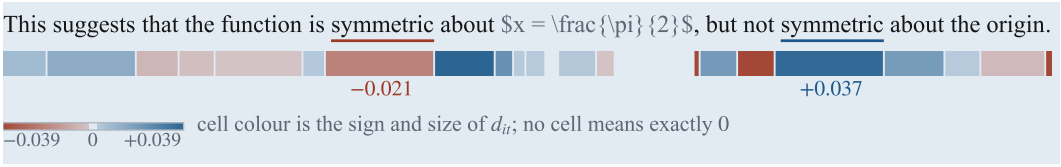


Figure 2: Contextual credit from a one-step teacher: in one zero-advantage response fragment, the two occurrences of *symmetric* receive opposite advantages.

This choice of reference connects the construction to delta distillation (Heo et al., 2026): teacher and student share their preceding training, and their log ratio isolates the preferences acquired through the current reward step. Each iteration builds a new teacher from the updated actor, so the teacher's one-step horizon incorporates the actor's accumulated learning.

We optimize the original actor with these detached advantages:

$$\mathcal{L}_{\text{OPGD}}(\theta) = -\sum_{i,t} v_{it}\ \text{sg}[d_{it}] \log \pi_\theta(x_{it} \mid c_{it}), \tag{7}$$

where v_{it} includes loss normalization and masking. For efficiency, teacher construction and distillation reuse the same rollout batch, avoiding additional generation (Algorithm 1). The teacher consequently depends on the same sampled contexts and actions used to estimate its distillation gradient, introducing bias relative to distillation on independent actor rollouts with the teacher held fixed.

We use plain SGD for the virtual step, consistent with its competitive performance against AdamW in RLVR (Mukherjee et al., 2026). This stateless update requires no separate teacher moments and avoids inheriting the actor's Adam moments, which accumulate gradients of the distillation objective.

Teacher strength. The lookahead rate α controls the teacher's adaptation independently of the actor learning rate. Reward extrapolation (Yang et al., 2026) combines teacher and actor logits as

$$z_{\text{ext}} = (1+\omega)z_E - \omega z_k, \qquad \omega \geq 0. \tag{8}$$

For a one-step teacher, this is locally equivalent to using virtual learning rate $(1+\omega)\alpha$. At the original actor, logit extrapolation scales the full-action OPD gradient exactly, without changing its direction (Appendix A.6).

4.2 NTK INTERPRETATION OF CONTEXTUAL CREDIT

The one-step teacher is optimized with response-level advantages, yet its OPD feedback can distinguish tokens within a response. This redistribution can be understood through the gradient interac-

Algorithm 1 On-Policy Gradient Distillation (OPGD)

Require: Actor θ_k, verifier R, prompt batch, group size G, lookahead rate α, retained fraction f

Collect verified trajectories

1: Sample G responses per prompt from π_{θ_k} and evaluate their rewards
2: Compute group advantages A_i and the outcome gradient g_k using Equations equation 1–equation 2

Construct the temporary teacher

3: Save actor parameters θ_k and sampled-token log probabilities ℓ^0_{it}
4: Apply the stateless SGD step $\theta \leftarrow \theta_k + \alpha g_k$
5: $d_{it} \leftarrow \mathrm{sg}[\log \pi_\theta(x_{it} \mid c_{it}) - \ell^0_{it}]$

Distill back into the actor

6: Restore $\theta \leftarrow \theta_k$ and clear temporary gradients
7: Retain the top fraction f of $|d_{it}|$ within each response ($f = 1$ retains all tokens)
8: Update the actor once using Equation equation 7 and advance its optimizer and schedule
9: **return** updated actor θ_{k+1}

tions induced by reward adaptation. Linearizing the teacher's log-probability changes gives

$$d_{it} \approx \alpha \boldsymbol{s}_{it}^\top g_k = \alpha \sum_{j,u} w_{ju} A_j \underbrace{\boldsymbol{s}_{it}^\top \boldsymbol{s}_{ju}}_{K(it,ju)} . \tag{9}$$

Here K is the empirical NTK of sampled-token log probabilities (Jacot et al., 2018), connecting the teacher's preferences to the cross-example gradient interactions studied in LLM finetuning by Ren & Sutherland (2025).

The diagonal contribution has the sign of the token's own response advantage, with magnitude proportional to its squared score norm. Interactions with other tokens can reinforce or reverse this contribution, allowing a common outcome signal to induce different preferences across contexts. Different uses of the same token can therefore receive opposite coefficients, as in Figure 2.

Because these interactions extend across responses, a zero outcome advantage does not preclude a distillation signal. When $A_i = 0$, all contributions from that response vanish and its targets arise entirely from other responses. Homogeneous-outcome groups can thus receive contextual feedback from outcome differences elsewhere in the batch despite contributing no reward gradient of their own.

4.3 Distillation geometry and reward alignment

Distilling these contextual preferences can change the actor's learning direction even when the teacher takes only the original outcome-gradient step. Matching conditional predictions weights that step through the actor's sensitivity across distillation contexts. Substituting Equation equation 9 into the distillation objective gives

$$h_k = -\nabla_\theta \mathcal{L}_{\mathrm{OPGD}}\big|_{\theta_k} \approx \alpha F_k g_k, \qquad F_k = \sum_{i,t} v_{it} \boldsymbol{s}_{it} \boldsymbol{s}_{it}^\top. \tag{10}$$

The empirical score second moment F_k therefore includes contexts from zero-advantage responses that do not contribute directly to g_k.

Directions with larger eigenvalues of F_k produce larger aggregate changes in sampled-token log probabilities and receive greater relative weight. Thus the distinction between distillation and the original reward step depends on the actor's local geometry: an isotropic F_k only rescales g_k, whereas anisotropy can redirect it.

This redirection preserves nonnegative alignment with the outcome gradient to first order because F_k is positive semidefinite for nonnegative distillation weights. In particular,

$$g_k^\top h_k \approx \alpha \sum_{i,t} v_{it} (\boldsymbol{s}_{it}^\top g_k)^2 \approx \frac{1}{\alpha} \sum_{i,t} v_{it} d_{it}^2 \geq 0. \tag{11}$$

Since each position contributes in proportion to $v_{it}d_{it}^2$, this alignment also provides a criterion for selecting which preferences to distill. With a fixed retention count and equal weights within each response, selecting the largest $|d_{it}|$ preserves the greatest local alignment. This motivates the magnitude-selection variant evaluated in Section 5.4, with the derivation and weighting conditions given in Appendix A.

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

We fine-tune Qwen3-4B (Yang et al., 2025) on DeepMath-103K prompts (He et al., 2026) with binary verifier rewards. Each batch contains 128 prompts and eight responses per prompt, sampled at temperature 1.0 and top-$p = 1.0$, with a 16,384-token response limit. The mathematical training runs share their initialization and use full-parameter AdamW with learning rate 10^{-6}, no reference-policy KL penalty, and no entropy bonus. OPGD constructs its temporary teacher with an SGD step of size $\alpha = 0.1$. The remaining settings are given in Appendix C.

The main comparison includes GRPO, DAPO (Yu et al., 2026), entropy-masked GRPO, full-target OPGD, and responsewise top-10% OPGD. Entropy-masked GRPO retains the highest-entropy 20% of tokens in each response in its policy-gradient loss, with the other baseline settings unchanged. The GRPO run is evaluated within 100- and 200-update budgets, and the other configurations use a 100-update budget. Published 500-update GRPO weights from Yang et al. (2026) provide an external accuracy reference. Evaluation uses 32 completions per problem on AIME'24, AIME'25, HMMT-Feb'25, and HMMT-Nov'25. Following prior work (Ding & Zhang, 2026), each run is reported at its best checkpoint within its budget (Appendix C). Code-generation results use separate runs trained on Eurus-RL-Code prompts (Cui et al., 2025).

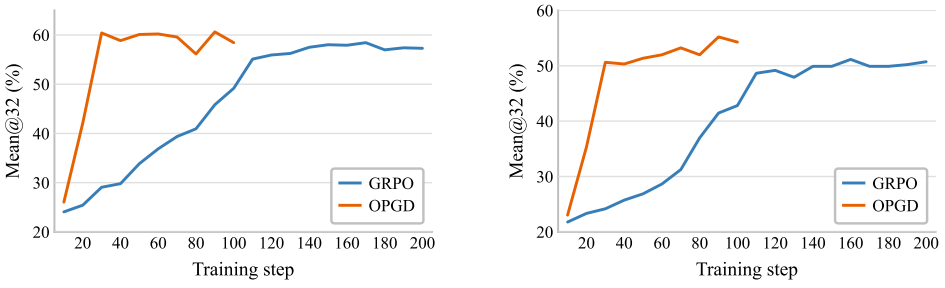
5.2 LEARNING PROGRESS ACROSS TASKS

Table 1: Mathematical reasoning performance at checkpoints selected within each training-update budget. The mean weights the four benchmarks equally. The 500-update GRPO reference uses published weights from Yang et al. (2026).

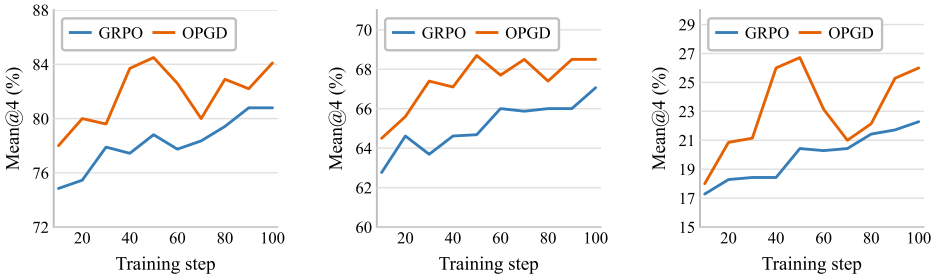
Metric	Method	Budget	AIME24	AIME25	HMMT-Feb25	HMMT-Nov25	Mean
Pass@1	Qwen3-4B	–	0.2229	0.2073	0.1094	0.0875	0.1568
	GRPO	100	0.4917	0.4281	0.2688	0.3135	0.3755
	GRPO (entropy 20%)	100	0.5458	0.4760	0.2833	0.3385	0.4109
	DAPO	100	0.5531	0.5240	0.2844	0.3552	0.4292
	GRPO	200	0.5844	0.5115	0.2969	0.3469	0.4349
	GRPO (external)	500	0.5823	**0.5604**	0.3219	0.3833	0.4620
	OPGD (full)	100	0.6063	0.5521	0.3302	0.3823	0.4677
	OPGD (top 10%)	100	**0.6667**	0.5448	**0.3542**	**0.3906**	**0.4891**
Pass@8	Qwen3-4B	–	0.4333	0.3339	0.2323	0.2745	0.3185
	GRPO	100	0.7843	0.6697	0.4483	0.5199	0.6056
	GRPO (entropy 20%)	100	0.7784	0.6787	0.4755	0.5398	0.6181
	DAPO	100	0.7736	0.6902	0.4649	0.5327	0.6154
	GRPO	200	0.7853	0.6804	0.5106	0.5452	0.6304
	GRPO (external)	500	0.7846	0.6925	0.4799	0.5409	0.6245
	OPGD (full)	100	0.7968	0.7212	0.4899	**0.5941**	0.6505
	OPGD (top 10%)	100	**0.8308**	**0.7354**	**0.5329**	0.5885	**0.6719**

Within the same 100-update budget, full-target OPGD surpasses GRPO across all four mathematical benchmarks and all three Pass@k metrics, with mean scores exceeding those of both the 200-update run and the published 500-update reference (Table 1and Appendix Table 4). The advantage develops early: on AIME, OPGD reaches an accuracy plateau within the first few dozen updates while GRPO improves gradually (Figure 3a–b). That plateau persists as mean response length contracts by more than half from its peak (Appendix D). Selecting the strongest 10% of targets further improves the mean benchmark scores.

Separate training on coding prompts yields the same early advantage. On HumanEval+, MBPP+ (Liu et al., 2023), and LiveCodeBench v6 (Jain et al., 2025), OPGD reaches GRPO's update-100 accuracy within 30–40 updates and remains ahead at every shared evaluation (Figure 3c–e). Appendix E gives the code-generation results in detail.



(a) AIME'24 (b) AIME'25



(c) HumanEval+ (d) MBPP+ (e) LiveCodeBench v6

Figure 3: **Learning progress in mathematical reasoning and code generation.** Top: mean sample correctness over 32 completions per problem on AIME, with full-target OPGD trained for 100 updates and GRPO for 200. Bottom: code-generation Pass@1 from separate runs trained on coding prompts over the first 100 updates, using four completions per problem.

Computational cost. The faster progress per update comes with additional teacher computation. An OPGD update takes about 1.5 times as long as GRPO in the measured profile, with overhead concentrated in the virtual reward backward pass and teacher scoring (Figure 4). Target selection retains complete-sequence scoring and contextual backpropagation, so this overhead does not fall proportionally with the selected fraction. Total training time also depends on the evolving response lengths.

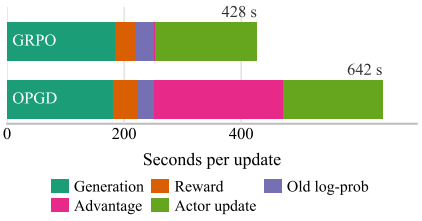


Figure 4: The first-update runtime.

5.3 Feedback from a one-step teacher

Nearly half the responses in the first diagnostic batch carry zero group advantage and contribute nothing to the outcome gradient, yet most of their teacher targets are nonzero, and the mixture of signs recurs across positive-, negative-, and zero-advantage responses (Figures 14 and 15). The sign follows the score interactions of Equation 9: a token's coefficient sums the batch outcome signal over its similarities with the other sampled tokens, so contexts drawing on different parts of the batch move oppositely.

Table 2: **Training cost against accuracy.** Wall clock per run, rescaled to 8×A800 by first-step latency, with the four-benchmark mean Pass@k scores from Table 1.

Method	Budget	Wall clock (h)	Pass@1	Pass@4	Pass@8
GRPO	100	15.88	0.3755	0.5515	0.6056
GRPO	200	56.58	0.4349	0.5799	0.6304
DAPO	100	40.00	0.4292	0.5698	0.6154
OPGD	100	31.31	**0.4677**	**0.6132**	**0.6505**

Magnitude instead tracks token uncertainty, which cannot separate the two signs: positions of comparable entropy split into both (Figure 9). Entropy-masked GRPO improves over the GRPO baseline through such a criterion, keeping the response-wide advantage at the positions it selects (Wang et al., 2025); full-target OPGD reaches higher means on all three metrics (Table 1) with a criterion that retains the sign.

Gradient structure. At the same actor and rollout batch, matching these preferences increases the concentration of singular-value energy in the parameter-gradient matrices before token selection (Figure 5). The median share of energy in the leading sixteen singular directions rises from 55.5% to 77.9%, with increases in nearly all matrices. Estimated spectral entropy also decreases, most strongly in later layers and the embedding. Because these spectra are normalized, the change reflects redistribution of gradient energy beyond a common rescaling of the outcome update (Appendix H).

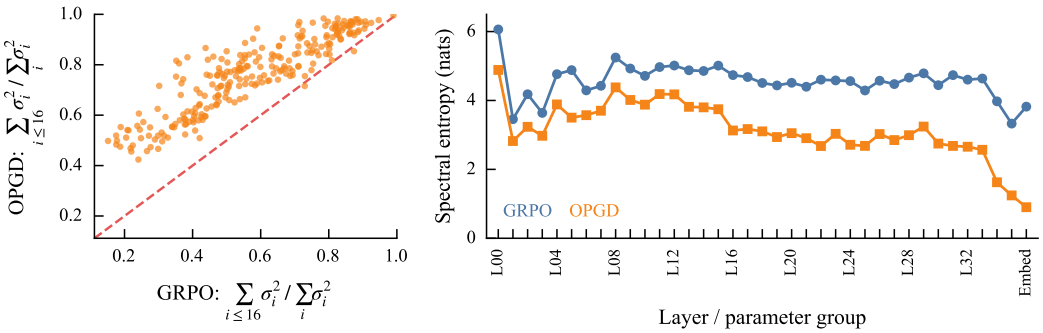


Figure 5: **Gradient structure under full-target distillation**, measured from the same actor and batch. Left: leading-16 energy shares for 253 gradient matrices, normalized by the exact squared Frobenius norm. Right: layer-group spectral-entropy proxy with a uniform unresolved tail.

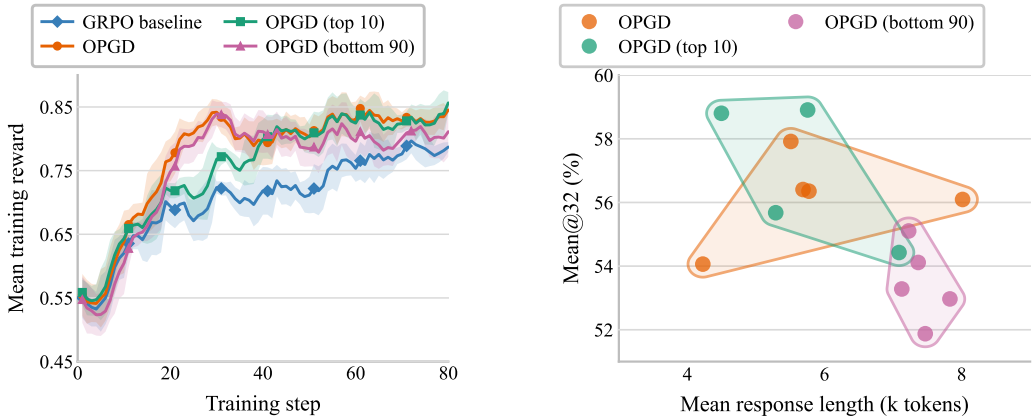
Table 3: Mathematical reasoning with complementary target subsets. Tokens are ranked within each response by absolute OPD advantage. Entries are percentages, and the mean weights the four benchmarks equally.

	Pass@1		Pass@4		Pass@8	
Benchmark	Top 10%	Bottom 90%	Top 10%	Bottom 90%	Top 10%	Bottom 90%
AIME24	66.67	59.48	81.22	76.67	83.08	80.47
AIME25	54.48	50.10	69.21	63.21	73.54	68.30
HMMT-Feb25	35.42	34.27	47.93	46.27	53.29	49.65
HMMT-Nov25	39.06	38.54	53.92	52.45	58.85	57.29
Mean	48.91	45.60	63.07	59.65	67.19	63.93

5.4 Effect of Target Selection

The full-target gains leave open how much of the teacher's feedback is needed for effective distillation. On the first diagnostic batch, the globally largest 10% of targets carry 76.9% of absolute mass, with approximately balanced positive and negative contributions (Appendix F). We compare retaining the largest 10% within each response with retaining the complementary 90%, constructing both teachers from the complete batch before selection.

Distilling the top 10% preserves rapid learning and approaches the full-target reward curve; the bottom 90% rises faster early on but converges in reward (Figure 6a). The top 10% still achieves higher scores across all benchmarks and sampling budgets (Table 3), and at later checkpoints combines higher accuracy with shorter responses (Figure 6b).



(a) Training reward

(b) Accuracy and response length

Figure 6: **Complementary target subsets.** Left: five-update reward averages with $\pm 1\sigma$ bands of raw rewards. Right: AIME Mean@32 (%) against mean response length at checkpoints after update 50, with each run's checkpoints enclosed by the outline of their convex hull. Each run keeps its colour across the two panels.

6 Discussions

An implicit preference over policies. Similar training rewards accompany different benchmark accuracy and response lengths across target subsets. This suggests that the choice of distillation contexts influences which behavioral solutions are reached during reward optimization. One possible explanation is an implicit optimization bias induced by target matching, connecting OPGD to the role of optimization geometry in solution selection (Gunasekar et al., 2018). Here the geometry depends on the sampled contexts and, under screening, on the teacher's reward-directed preferences. It offers a route for favoring particular policy changes without adding a reference-policy KL penalty.

Designing the teacher and matching objective jointly. Bootstrapped meta-learning treats target construction and the matching metric as separate design choices (Flennerhag et al., 2022). For OPGD the interaction is concrete: under a shared full-target score moment, inverse-score preconditioning in the virtual step would locally cancel the reweighting from conditional-policy matching (Appendix A.4), so compatibility with the matching objective, not standalone reward, becomes a criterion for teacher design.

Batch construction and the learning rule. Pooling rollouts into one teacher generally differs from constructing separate teachers for rollout partitions and averaging the updates distilled on each partition. With full targets and consistent normalization, the pooled update includes interactions between one partition's reward gradient and another's distillation contexts under the local approximation. Separate teachers omit these cross-partition terms (Appendix A.2).

7 Conclusion

We showed that a teacher only one reward-gradient step ahead can sustain effective on-policy distillation throughout outcome-based training. OPGD turns incremental reward adaptation into contextual supervision, with feedback extending across responses and a retained gradient shaped by the actor's local geometry. On Qwen3-4B, this construction accelerates learning in separate mathematical reasoning and code-generation experiments. The results support continually renewing the teacher alongside the actor, making distillation part of policy optimization without a separate expert-training stage.

AI USE STATEMENT

AI assistants were used during the preparation of this manuscript to assist with limited aspects of the writing process, including reorganizing LaTeX source files, improving the clarity and readability of the prose, and refining the presentation of technical descriptions. All mathematical formulations, experimental results, analyses, and scientific claims were independently verified by the authors. The authors have carefully reviewed and revised all AI-assisted edits and remain fully responsible for the accuracy, integrity, and final content of the submitted manuscript.

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 *ICLR*, 2024.

Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998. doi: 10.1162/089976698300017746.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Yi Ding and Ruqi Zhang. Does on-policy distillation really distill? from noisy teacher to self-improvement. *arXiv preprint arXiv:2608.31046*, 2026.

Sebastian Flennerhag, Yannick Schroecker, Tom Zahavy, Hado van Hasselt, David Silver, and Satinder Singh. Bootstrapped meta-learning. In *ICLR*, 2022.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: Knowledge distillation of large language models. In *ICLR*, 2024.

Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pp. 1832–1841. PMLR, 2018.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. In *International Conference on Learning Representations*, volume 2026, pp. 138306–138322, 2026.

Byeongho Heo, Jaehui Hwang, Sangdoo Yun, and Dongyoon Han. On-policy delta distillation. *arXiv preprint arXiv:2607.15161*, 2026.

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. *arXiv preprint arXiv:2601.20802*, 2026.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *NeurIPS*, 2018.

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 *ICLR*, 2025.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Refining credit assignment in RL training of LLMs. *arXiv preprint arXiv:2410.01679*, 2024.

Yang Li, Semih Yavuz, and Shafiq Joty. RISE: Recursive improvement via self-extrapolating policy distillation. *arXiv preprint arXiv:2609.05295*, 2026a.

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

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 *ICLR*, 2024.

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

Sagnik Mukherjee, Lifan Yuan, Pavan Jayasinha, Dilek Hakkani-Tür, and Hao Peng. Do we need adam? surprisingly strong and sparse reinforcement learning with sgd in llms. *arXiv preprint arXiv:2602.07729*, 2026.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *NeurIPS*, 2023.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to Q^*: Your language model is secretly a Q-function. In *COLM*, 2024.

Yi Ren and Danica J. Sutherland. Learning dynamics of LLM finetuning. In *ICLR*, 2025.

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

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, 2025.

Wei Sun, Wen Yang, Pu Jian, Qianlong Du, Fuwei Cui, Shuo Ren, and Jiajun Zhang. KTAE: A model-free algorithm to key-tokens advantage estimation in mathematical reasoning. In *NeurIPS*, 2025.

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. *arXiv preprint arXiv:2506.01939*, 2025.

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*, 2025.

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

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

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels. In *ICML*, 2025.

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.