OPPO: Bayesian Value Recursion for Token-Level Credit Assignment in LLM Reasoning

Yu Li[1] **Rui Miao**[2] **Tian Lan**[1*] **Zhengling Qi**[3*]
[1]Department of Electrical and Computer Engineering, George Washington University
[2]Department of Mathematical Sciences, The University of Texas at Dallas
[3]School of Business, George Washington University

Abstract

Reinforcement learning with verifiable rewards has become the standard recipe for improving LLM reasoning, but the dominant algorithm GRPO assigns a single trajectory-level advantage to the tokens of a response, diluting the signal at pivotal reasoning steps and injecting noise at uninformative ones. Critic-free alternatives derived from on-policy distillation supply per-token signals through oracle-conditioned likelihood ratios, yet apply each signal in isolation from the trajectory-level evidence accumulated up to that position. We propose Oracle-Prompted Policy Optimization (OPPO), which rests on a single observation: the oracle signal used by prior distillation-style methods for local discrimination is also the Bayesian update of the model's belief about eventual success. Accumulating the signal along a trajectory yields, in closed form and at the cost of one extra forward pass, a running estimate of the success probability along the trajectory, together with a token-level advantage that dispenses with a learned value network and with additional rollouts. First-order analysis factorizes the advantage into the per-token discrimination signal used by distillation methods, modulated by a state weight that concentrates credit on pivotal tokens and yields a directional variance-reduction guarantee. The framework admits two estimators differing only in which model scores the evidence: a *self-oracle* that reuses the student and recovers the on-policy distillation reward as a strict special case, and a *teacher-oracle* that delegates scoring to a stronger frozen model. On two instruction-tuned LLMs across seven mathematics, science, and code reasoning benchmarks, OPPO outperforms GRPO, DAPO, and SDPO, with gains of up to $+7.0$ points on AMC'23 and $+5.2$ points on AIME'24 over the underlying GRPO or DAPO baseline and gains that widen monotonically with response length.

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) has become a central paradigm for improving the reasoning capabilities of large language models [10, 41, 14]. Under RLVR, the model generates a complete reasoning trajectory, receives a binary reward for the correctness of the final answer, and updates its policy toward correct trajectories. Group Relative Policy Optimization (GRPO) [10] is the most widely adopted on-policy algorithm in the paradigm and estimates the advantage of each trajectory relative to a group of sampled responses.

GRPO broadcasts a single scalar advantage to the token positions of a response [28]. In a reasoning chain of several hundred tokens, a small subset of tokens determines the outcome, namely a strategy selection, a key algebraic step, or a critical deduction [31, 38]. Equal credit therefore dilutes the

*Corresponding authors: tlan@gwu.edu, qizhengling@gwu.edu.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

signal at pivotal tokens and injects noise at uninformative ones, and the problem worsens as chains grow longer [43, 17].

Recent work on the credit assignment bottleneck follows two strategies. The first refines the GRPO framework while retaining the trajectory-level advantage. Dr.GRPO [28], DAPO [43], GSPO [46], and GMPO [45] modify the importance ratio, clipping, or advantage normalization, and BiCC [23] conditions each sample on the correct and incorrect traces of its group, yet the tokens of a response still receive the same advantage [11]. The second seeks finer-grained credit at the token level. Monte Carlo resampling methods such as VinePPO [20] and TreeRL [12] require orders of magnitude more inference, implicit process reward models such as PRIME [7] require online updating, and segment-level aggregation in SPO [11] pools descendant outcomes within shared prefixes. Most relevant to the present work, on-policy distillation methods with privileged information [2, 9, 29, 42, 13] condition the model on the ground-truth answer y^* and use the resulting log-likelihood ratio $\log \lambda_t = \log \pi_{\text{teacher}}(y_t \mid s_t) - \log \pi_{\text{student}}(y_t \mid s_t)$ as a token-level credit signal. The ratio $\log \lambda_t$ is the one-step expert log-ratio between the success-conditional optimal policy π^* and the student, and these methods apply $\log \lambda_t$ with uniform weight and treat each token's credit as an isolated one-step quantity decoupled from the cumulative evidence along $y_{<t}$.

Motivated by this gap, we propose Oracle-Prompted Policy Optimization (OPPO), which rests on a single observation: the per-token oracle signal used in distillation-style methods is also the Bayesian update of the model's belief about eventual success. Accumulating the signal along a trajectory yields, in closed form and at zero extra rollout cost, a running estimate of the success probability along the trajectory. The resulting token-level advantage dispenses with a learned value network and sums exactly to the trajectory's outcome relative to its prior. The recursion admits two estimators that differ only in which model scores the evidence: a self-oracle reuses the student in one extra forward pass and recovers the on-policy distillation reward as a strict special case, while a teacher-oracle delegates scoring to a stronger frozen model. The advantage factorizes as the per-token discrimination signal modulated by a state weight $V_t(1 - V_t)$ that peaks where the running belief is most uncertain and vanishes once the trajectory has committed. A direction-anchoring step ties the sign of each token-level advantage to the sequence-level outcome, so the verifier decides which trajectories are reinforced while the Bayesian machinery redistributes credit within each trajectory.

❶ We show that the OPD per-token reward $\log \lambda_t$ is simultaneously an expert log-ratio surrogate for $A_t^{\pi^*}$ and the Bayesian sufficient statistic for V_t, with the probability measure selecting the self-oracle or teacher-oracle estimator.

❷ We derive a closed-form Bayesian value recursion whose token-level advantage dispenses with a critic and with extra rollouts, satisfies the telescoping budget $\sum_t A_t = R - V_0$, and factorizes as $A_t \approx V_t(1 - V_t) \log \lambda_t$ with a directional variance-reduction guarantee. Direction anchoring and evidence clipping integrate the advantage into GRPO at one extra forward pass per trajectory.

❸ We evaluate the method on Qwen3 and Phi models across seven reasoning benchmarks, where it surpasses trajectory-level and oracle-conditioned baselines by up to $+7.0$ points on AMC'23 and $+5.2$ on AIME'24 over the underlying GRPO or DAPO baseline. The gains widen with response length, and the advantage tracks Monte Carlo credit at 81% to 91% of the sampling ceiling.

2 Preliminaries

2.1 Reinforcement Learning for LLM Reasoning

We model autoregressive LLM generation as a finite-horizon Markov decision process $(\mathcal{S}, \mathcal{V}, \pi_\theta, R)$ over a finite vocabulary $\mathcal{V}$. Given a query x, the state at step t is the prefix $s_t \triangleq (x, y_{<t})$ with $y_{<t} \triangleq (y_1, \ldots, y_{t-1})$ and $s_1 = (x)$. The policy $\pi_\theta(\cdot \mid s_t) \in \Delta(\mathcal{V})$ samples a token y_t, yielding the deterministic transition $s_{t+1} = (s_t, y_t)$. Generation terminates at the first step $T \leq T_{\max}$ where y_T is the end-of-sequence symbol. A verifier assigns a terminal binary reward $R(\tau) \in \{0, 1\}$ indicating answer correctness, with zero intermediate rewards. Training maximizes the expected return through policy gradient

$$J(\theta) \triangleq \mathbb{E}_{x \sim \mathcal{D},\, \tau \sim \pi_\theta(\cdot|x)}[R(\tau)]\,, \qquad \nabla_\theta J(\theta) = \mathbb{E}\left[\sum_{t=1}^{T} A_t(s_t, y_t)\, \nabla_\theta \log \pi_\theta(y_t \mid s_t)\right], \tag{1}$$

where $\mathcal{D}$ is the query distribution and $A_t : \mathcal{S} \times \mathcal{V} \to \mathbb{R}$ is a per-token advantage estimator unbiased against $Q_t(s_t, y_t) \triangleq \mathbb{E}_{\pi_\theta}[R \mid s_t, y_t]$ when paired with the score function $\nabla_\theta \log \pi_\theta(y_t \mid s_t)$.

GRPO [10] forgoes a learned value baseline by drawing a group of $G \geq 2$ trajectories $\{\tau_i\}_{i=1}^G \sim \pi_\theta(\cdot \mid x)$ per query and assigning the tokens of τ_i the trajectory-level standardized advantage

$$\hat{A}_i^{\mathrm{GRPO}} \triangleq \frac{R_i - \mu}{\sigma + \varepsilon}, \qquad \mu \triangleq \frac{1}{G}\sum_{j=1}^{G} R_j, \quad \sigma \triangleq \sqrt{\frac{1}{G}\sum_{j=1}^{G}(R_j - \mu)^2}, \tag{2}$$

so that $A_t(\tau_i) \equiv \hat{A}_i^{\mathrm{GRPO}}$ for all $t \in \{1, \ldots, T_i\}$. The estimator is exchangeable in t, so if a subset of $k \ll T$ positions carries the responsibility for R, the signal there is attenuated by k/T while the remaining positions receive gradient mass with zero expected contribution [15, 25].

The cleanest reference quantity for token-level credit is the on-policy state value

$$V_t \triangleq P^{\pi_\theta}(R = 1 \mid x, y_{1:t-1}), \qquad V_0 \triangleq P^{\pi_\theta}(R{=}1 \mid x), \qquad V_{T+1} \triangleq R, \tag{3}$$

which satisfies the Bellman consistency $V_t = \mathbb{E}_{y_t \sim \pi_\theta}[V_{t+1}]$. The ideal token-level advantage is the one-step temporal difference $A_t \triangleq V_{t+1} - V_t = Q_t(s_t, y_t) - V_t(s_t)$, and telescoping along the trajectory gives

$$\sum_{t=1}^{T} A_t \;=\; V_{T+1} - V_0 \;=\; R - V_0, \tag{4}$$

so token-level credit assignment amounts to distributing the budget $R - V_0$ across the T positions. PPO [34] trains a value network $V_\phi(s_t)$ to approximate Eq. (3) at substantial memory cost, and GRPO sets $V_t \equiv V_0$ and distributes $R - V_0$ uniformly, so the variance of R enters the token positions [27, 37].

2.2 On-Policy Distillation

On-policy distillation (OPD) pairs a student policy π_θ with a frozen teacher π_T and supervises π_θ on its own rollouts under a per-token divergence [2, 9, 29]. In the reasoning setting most relevant to RLVR, the teacher is conditioned on privileged information such as the ground-truth answer y^*, so that $\pi_T(\cdot \mid x, y_{<t}) \equiv \pi_T(\cdot \mid x, y^*, y_{<t})$ approximates the success-conditional next-token distribution $P(\cdot \mid x, y_{<t}, R{=}1)$ [44, 13]. Drawing $y_{1:T} \sim \pi_\theta(\cdot \mid x)$, OPD minimizes the on-policy summed reverse KL

$$\mathcal{L}_{\mathrm{OPD}}(\theta) \triangleq \mathbb{E}_{x,\, y_{1:T} \sim \pi_\theta(\cdot|x)}\left[\sum_{t=1}^{T} \mathrm{KL}\big(\pi_\theta(\cdot \mid s_t) \,\big\|\, \pi_T(\cdot \mid s_t)\big)\right]. \tag{5}$$

Standard practice [2, 29] treats the rollout as a stop-gradient sample and differentiates only the inner KL through the student logits, under which the contribution at the sampled token y_t reduces to

$$\nabla_\theta \mathcal{L}_{\mathrm{OPD}}(\theta)\big|_{y_t} = -\log \lambda_t^{\mathrm{OPD}} \cdot \nabla_\theta \log \pi_\theta(y_t \mid s_t), \quad \log \lambda_t^{\mathrm{OPD}} \triangleq \log \pi_T(y_t \mid s_t) - \log \pi_\theta(y_t \mid s_t). \tag{6}$$

Comparing Eq. (6) with Eq. (1) shows that the OPD update is a REINFORCE estimator in which $\log \lambda_t^{\mathrm{OPD}}$ plays the role of a per-token reward in place of an advantage. The reward coincides with the KL-shaping term of PPO-based RLHF [34] when the privileged teacher is the reference policy and the environment reward vanishes, so OPD is token-level PPO with neither a value baseline nor clipping. The positions are then updated independently of their place in the trajectory, and $\log \lambda_t^{\mathrm{OPD}}$ stays decoupled from the cumulative agreement along $y_{<t}$ and from R [42, 37].

Advantage under the optimal policy. The unbiasedness condition below Eq. (1) leaves freedom in the reference policy under which A_t is defined: replacing $A_t^{\pi_\theta}$ with the advantage under a policy π^* that dominates π_θ in value preserves a valid policy-improvement direction [34, 15]. Let

$$\pi^*(y_t \mid s_t) \;\triangleq\; P(y_t \mid x, y_{<t}, R{=}1), \qquad V_t^{\pi^*}(s_t) \;\triangleq\; P^{\pi^*}(R{=}1 \mid s_t), \tag{7}$$

denote the success-conditional optimal policy and its value, with $A_t^{\pi^*} \triangleq Q_t^{\pi^*}(s_t, y_t) - V_t^{\pi^*}(s_t)$ the corresponding one-step advantage. The OPD teacher is an approximation $\pi_T \approx \pi^*$, so $\log \lambda_t^{\mathrm{OPD}}$ in Eq. (6) is the one-step expert log-ratio between π^* and π_θ, a per-token surrogate for $A_t^{\pi^*}$ in the spirit of advantage-weighted regression.

Bridge to our method. The GRPO advantage in Eq. (2) resolves the credit budget $R - V_0$ uniformly across the T tokens, while the OPD signal $\log \lambda_t$ in Eq. (6) is purely local and ignores both the cumulative evidence along $y_{<t}$ and the terminal outcome R. Between the two extremes lies an estimator that retains the dense π^*-anchored signal, reweights it by a position-dependent coefficient that tracks the running posterior of success in Eq. (3), and dispenses with a learned value network.

3 Bayesian Evidence Aggregation

The bridge of Section 2.2 singles out V_t as the object of token-level credit assignment, since the ideal advantage $A_t = V_{t+1} - V_t$ telescopes to $R - V_0$ in Eq. (4). A tracker of V_t therefore distributes the budget across tokens, and the challenge is to track V_t while dispensing with a learned critic. The recursion below reduces the problem to an additive log-odds update driven by a per-token Bayes factor, two surrogates for the factor follow, and Table 1 fixes the symbols used throughout.

Table 1: The table fixes the notation of Sections 3 and 4.

$\lambda_t^\star$	exact Bayes factor of token y_t, Eq. (9)	V_t, ℓ_t	running belief of success and its log-odds
λ_t	its estimator under a scorer $\hat{\pi}$, Eq. (14)	A_t	raw advantage $V_{t+1} - V_t$, Eq. (16)
$\log \hat{\lambda}_{i,t}$, ϵ_t	clipped evidence, Eq. (20), and success-branch KL, Eq. (15)	$\hat{A}_{i,t}$, $\tilde{A}_{i,t}$	anchored and group-normalized advantage, Eq. (21)

3.1 Bayesian Value Recursion

Applying Bayes' theorem to update the success probability after observing y_t gives

$$V_{t+1} \;=\; P^{\pi_\theta}(R{=}1 \mid x, y_{1:t}) \;=\; \frac{P(y_t \mid x, y_{<t}, R{=}1)\, V_t}{P(y_t \mid x, y_{<t})}. \tag{8}$$

Define the per-token Bayes factor between the success and failure branches as

$$\lambda_t^\star \;\triangleq\; \frac{P(y_t \mid x, y_{<t}, R{=}1)}{P(y_t \mid x, y_{<t}, R{=}0)} \;=\; \frac{P(y_t \mid x, y_{<t}, y_T{=}y^*)}{P(y_t \mid x, y_{<t}, y_T{\neq}y^*)}, \tag{9}$$

and let $\ell_t \triangleq \log \frac{V_t}{1-V_t}$ collect the running log-odds of success. The probability P in Eqs. (8) and (9) can be taken under either π_θ or π^* from Eq. (7), yielding valid Bayesian recursions for $V_t^{\pi_\theta}$ and $V_t^{\pi^*}$, two values that differ in general, with $V_t^{\pi^*}$ targeted by the expert log-ratio of Section 2.2. Using the law of total probability $P(y_t \mid x, y_{<t}) = P(y_t \mid x, y_{<t}, R{=}1)\, V_t + P(y_t \mid x, y_{<t}, R{=}0)\,(1 - V_t)$, Eq. (8) can be rewritten as the additive log-odds recursion

$$V_{t+1} \;=\; \frac{\lambda_t^\star\, V_t}{\lambda_t^\star\, V_t + (1 - V_t)}, \qquad \ell_{t+1} \;=\; \ell_t + \log \lambda_t^\star, \tag{10}$$

in which the tokens contribute additively to ℓ_t via its log-Bayes-factor. Unrolling from $\ell_0 = \mathrm{logit}(V_0)$ and applying the sigmoid yields the closed-form belief at position t,

$$V_t \;=\; \sigma\!\left(\ell_0 + \sum_{s<t} \log \lambda_s^\star\right), \qquad V_{T+1} \;=\; R, \tag{11}$$

so the running belief is a function of the prior and the accumulated evidence alone, and estimators that respect the additive log-odds structure of Eq. (10) track it through the same accumulation. Tracking V_t therefore reduces to estimating $\log \lambda_t^\star$, whose definition involves the success- and failure-conditional next-token distributions, and neither distribution is directly observable.

3.2 Two Oracle Surrogates for Bayesian Evidence

We construct estimators of $\lambda_t^\star$ under either probability measure (π_θ or π^*). Both arise from the same pair of modeling choices applied to whichever scorer $\hat{\pi}$ realizes the measure.

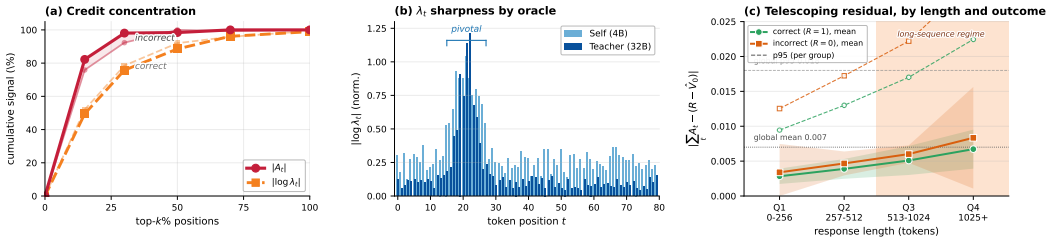

Figure 1: The figure analyzes 200 MATH-500 trajectories from Qwen3-4B-Instruct. (a) $|A_t|$ concentrates more sharply than $|\log \lambda_t|$, with the gap widest on incorrect trajectories where V_t saturates early. (b) The Qwen3-32B teacher produces sharper $|\log \lambda_t|$ at pivotal tokens and lower background in determined regions. (c) The budget residual $|\sum_t A_t - (R - \hat{V}_0)|$ grows with the length quartile for both outcomes, and evidence clipping caps the recursion in the long-sequence regime.

Oracle Conditioning

The success-conditional distribution is approximated by appending y^* to the prompt:

$$P(y_t \mid x, y_{<t}, R{=}1) \;\approx\; \hat{\pi}(y_t \mid x, y_{<t}, y^*). \tag{12}$$

The justification is the privileged-teacher rationale of Section 2.2: conditioning on y^* constrains the scorer to tokens consistent with reaching the correct answer.

Failure Branch

The failure-conditional distribution is approximated by the marginal under the same scorer:

$$P(y_t \mid x, y_{<t}, R{=}0) \;\approx\; \hat{\pi}(y_t \mid x, y_{<t}). \tag{13}$$

The marginal admits the decomposition $\hat{\pi}(y_t \mid s_t) = P(y_t \mid s_t, R{=}1)\, V_t + P(y_t \mid s_t, R{=}0)\,(1 - V_t)$: the failure branch dominates as $V_t \to 0$ and the approximation is tightest where incorrect trajectories spend most of their length, while the gap is largest near $V_t = \frac{1}{2}$ where the state weight $V_t(1 - V_t)$ peaks. Conditioning the denominator on a wrong answer sampled from the incorrect rollouts estimates the failure branch directly at the cost of a second answer-conditioned pass; Appendix D.2 reports both forms, whose evidence sequences correlate at 0.91 and whose accuracies differ by at most 0.4 points, so the marginal is the default.

Substituting Modeling Choices 3.2 and 3.2 into Eq. (9) yields the unified estimator

$$\lambda_t \;\triangleq\; \frac{\hat{\pi}(y_t \mid x, y_{<t}, y^*)}{\hat{\pi}(y_t \mid x, y_{<t})}, \qquad \log \lambda_t \;=\; \log \hat{\pi}(y_t \mid x, y_{<t}, y^*) - \log \hat{\pi}(y_t \mid x, y_{<t}), \tag{14}$$

parameterized by a single scorer $\hat{\pi}$. Setting $\hat{\pi} = \pi_\theta$ yields the *self-oracle*, which requires only one extra forward pass per trajectory and recovers the OPD per-token reward of Eq. (6) exactly, since the OPD privileged teacher is the student conditioned on y^* [44]. Setting $\hat{\pi} = \pi_\phi$ for a frozen scorer that approximates π^* at lower KL distance than π_θ [2, 29] yields the *teacher-oracle*, which reduces approximation error at the cost of a separate scoring model.

The pointwise gap between either estimator and the exact Bayes factor is captured by the success-branch KL

$$\epsilon_t \;\triangleq\; D_{\mathrm{KL}}\big(P(\cdot \mid x, y_{<t}, R{=}1) \,\big\|\, \hat{\pi}(\cdot \mid x, y_{<t}, y^*)\big), \tag{15}$$

which traces a design axis: the self-oracle yields the largest ϵ_t, and a perfect teacher drives $\epsilon_t \to 0$ and recovers the exact recursion. Substituting λ_t for $\lambda_t^\star$ in Eq. (10) gives the implementable form, with the budget residual controlled by $\sum_t \epsilon_t$ (Appendix D.1). The telescoping identity of Section 3.3 applies to the raw advantage before clipping, sign anchoring, and group normalization in Section 4; the implemented training advantage redistributes credit within each trajectory and inherits its direction from the verifier. Figure 1(c) stratifies the residual by length quartile and outcome: the mean rises monotonically from Q1 to Q4 in both classes, with $R{=}0$ above $R{=}1$ because V_t saturates earlier on incorrect rollouts.

3.3 From Belief to Advantage

The token-level advantage is the one-step change of the belief, and telescoping it along the trajectory gives the credit identity,

$$A_t \;=\; V_{t+1} - V_t \;=\; \sigma(\ell_t + \log \lambda_t^\star) - \sigma(\ell_t), \qquad \sum_{t=1}^{T} A_t \;=\; \sigma(\ell_{T+1}) - \sigma(\ell_0) \;=\; R - V_0, \tag{16}$$

which fixes the total credit by the outcome and the prior alone. Since σ is Lipschitz with constant $\frac{1}{4}$, every token satisfies $|A_t| \le \min(\frac{1}{4}|\log \lambda_t^\star|, 1)$. Subtracting V_t from both sides of Eq. (10) gives the closed-form identity

$$A_t \;=\; \frac{V_t(1-V_t)(\lambda_t - 1)}{\lambda_t V_t + 1 - V_t}. \tag{17}$$

For $|\lambda_t - 1|$ small, $\lambda_t - 1 \approx \log \lambda_t$ and the denominator approaches 1, yielding the first-order approximation

$$A_t \;\approx\; V_t(1-V_t) \cdot \log \lambda_t. \tag{18}$$

The factorization separates two roles: the oracle evidence $\log \lambda_t$ provides per-token discrimination between success and failure branches, identical to the gradient signal of on-policy distillation in Eq. (6), while the state weight $V_t(1-V_t)$ modulates the evidence by the running belief. The Brier score of V_t in Appendix H.1 decreases monotonically from $t = T/4$ to $t = T$, so the intermediate belief tracks the evolving success probability. Appendix H.2 checks the advantage itself against a Monte Carlo estimate of per-segment credit on 200 MATH-500 trajectories. The self-oracle advantage reaches a Spearman correlation of 0.63 and the teacher-oracle 0.71 against a split-half reliability ceiling of 0.78, while the per-token evidence alone reaches 0.41 and the uniform GRPO advantage 0.02.

Since $V_t(1-V_t)$ collapses once the running belief commits, the factorization predicts that credit concentrates on undetermined positions. On incorrect trajectories V_t saturates toward 0 early, so the state weight collapses even where $\log \lambda_t$ remains elevated by answer-correlated content, and Figure 1(a) confirms that the cumulative $|A_t|$ rises more steeply than $|\log \lambda_t|$, with the gap widest on incorrect trajectories. The factorization also predicts that a stronger oracle sharpens λ_t at pivotal positions while keeping the magnitude small elsewhere, and panel (b) shows the 32B teacher concentrating $|\log \lambda_t|$ more tightly around strategy-selection tokens than the 4B self-oracle. The prior supplies a complementary adaptation, since $V_0(1-V_0)$ multiplies the entire trajectory and downweights easy and unsolvable queries while dispensing with manual thresholding [7].

The state weight admits a variance-reduction interpretation under an idealized score-vector assumption. Writing the policy gradient as $\hat{g} = \sum_t w_t h_t$ with $h_t = \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t})$ and treating the h_t as uncorrelated across positions, the variance gap between the state-blind weighting $w_t = \log \lambda_t$ and the Bayesian weighting $w_t = V_t(1-V_t)\log \lambda_t$ satisfies

$$\begin{aligned} \mathrm{Var}[\hat{g}_0] - \mathrm{Var}[\hat{g}_1] \;&\ge\; (1-\gamma^2) \sum_{t \in T_{\mathrm{det}}} (\log \lambda_t)^2 \cdot \mathrm{Var}[h_t], \\ T_{\mathrm{det}} \;&=\; \{\, t : |\log \lambda_t| \ge \delta,\; V_t(1-V_t) < \gamma \,\}. \end{aligned} \tag{19}$$

Score vectors in autoregressive models are correlated across positions, so Eq. (19) is a directional indication, and the set T_{det} expands once the model commits to a solution path, predicting larger gains on longer trajectories, consistent with Figure 1(c) and Table 4(b).

4 OPPO: Oracle-Prompted Policy Optimization

The recursion of Section 3 leaves the estimator of λ_t open, and OPPO fixes it together with the prior V_0 and the direction anchoring.

Oracle estimator. Computing λ_t requires scoring the sampled tokens both with and without y^* in context. In the *self-oracle* mode, the policy π_θ scores both contexts, adding one forward pass per trajectory while reusing the policy weights. In the *teacher-oracle* mode, a frozen π_ϕ replaces π_θ in both terms; π_ϕ stays frozen and need not be instruction-tuned, since its role is likelihood scoring. The two modes sit at different points of the ϵ_t axis of Section 3.2, and the steps after $\log \hat{\lambda}_{i,t}$ is identical.

Prior and ratio computation. Given G trajectories per query, the prior is the $\mathrm{Beta}(\alpha, \alpha)$ posterior mean, and ratios are clipped for numerical stability:

$$\hat{V}_0 = \frac{k+\alpha}{G+2\alpha}, \qquad \ell_0 = \log \frac{k+\alpha}{G-k+\alpha}, \qquad \log \hat{\lambda}_{i,t} = \mathrm{clip}\big(o_{i,t} - s_{i,t},\, -C,\, C\big), \tag{20}$$

where $s_{i,t} = \log \hat{\pi}(y_{i,t} \mid x, y_{i,<t})$ and $o_{i,t} = \log \hat{\pi}(y_{i,t} \mid x, y^*, y_{i,<t})$. Setting $\alpha = 1$ keeps the log-odds finite in all-correct and all-incorrect groups, which would otherwise collapse the advantages to zero, while the clip bound C prevents a single extreme $\log \lambda_t$ from saturating ℓ_t and silencing later positions. The log-odds then accumulate as $\ell_{i,t} = \ell_0 + \sum_{j<t} \log \hat{\lambda}_{i,j}$ and the raw Bayesian advantage is $A^{\mathrm{raw}}_{i,t} = \sigma(\ell_{i,t} + \log \hat{\lambda}_{i,t}) - \sigma(\ell_{i,t})$.

Direction anchoring. The sign of $A^{\mathrm{raw}}_{i,t}$ is determined by $\log \lambda_t$, not by the trajectory outcome. Tokens with $\log \lambda_t > 0$ in an incorrect trajectory would receive positive advantage and reinforce flawed steps, and tokens with $\log \lambda_t < 0$ in a correct trajectory would be penalized; the failure mode mirrors pure self-distillation, where the gradient direction follows the privileged teacher in place of the environment reward [42]. To prevent the inversion, OPPO anchors the signs of the token-level advantages to the sequence-level GRPO advantage and normalizes across the group:

$$\hat{A}_{i,t} = \mathrm{sign}\big(A^{\mathrm{seq}}_i\big) \cdot |A^{\mathrm{raw}}_{i,t}|, \; A^{\mathrm{seq}}_i = \tfrac{R_i - \mu}{\sigma}, \qquad \tilde{A}_{i,t} = \frac{\hat{A}_{i,t} - \bar{A}}{\sigma_A + \epsilon}. \tag{21}$$

The environment reward retains exclusive control over which trajectories are reinforced or penalized, while the Bayesian state weight and oracle evidence govern only the relative magnitude across positions within each trajectory.

Algorithm 1 Oracle-Prompted Policy Optimization

Require: Policy π_θ, oracle estimator $\hat{\pi}$ (π_θ or π_ϕ), group size G, prior α, clip C

for each query x with answer y^* **do**
  # Rollout and reward
  Sample $\{y^{(i)}\}_{i=1}^{G} \sim \pi_\theta(\cdot \mid x)$
  Obtain R_i; $A^{\mathrm{seq}}_i \leftarrow (R_i - \mu)/\sigma$
  $k \leftarrow \sum_i R_i$; $\ell_0 \leftarrow \log \frac{k+\alpha}{G-k+\alpha}$
  # Oracle evaluation
  for $i = 1, \ldots, G$ **do**
    $s_t \leftarrow \log \hat{\pi}(y_{i,t} \mid x, y_{i,<t})$
    $o_t \leftarrow \log \hat{\pi}(y_{i,t} \mid x, y^*, y_{i,<t})$
    $\log \hat{\lambda}_{i,t} \leftarrow \mathrm{clip}(o_t - s_t, -C, C)$
  end for
  # Bayesian credit assignment
  for $i = 1, \ldots, G$ **do**
    $\ell_{i,t} \leftarrow \ell_0 + \sum_{j<t} \log \hat{\lambda}_{i,j}$
    $A^{\mathrm{raw}}_{i,t} \leftarrow \sigma(\ell_{i,t} + \log \hat{\lambda}_{i,t}) - \sigma(\ell_{i,t})$
    $\hat{A}_{i,t} \leftarrow \mathrm{sign}(A^{\mathrm{seq}}_i) \cdot |A^{\mathrm{raw}}_{i,t}|$
  end for
  $\tilde{A}_{i,t} \leftarrow (\hat{A}_{i,t} - \bar{A})/(\sigma_A + \epsilon)$
end for
 # Policy update
Update θ via Eq. (22)

Policy update. The normalized advantages enter the clipped surrogate objective

$$\mathcal{L}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\min\Big(\rho_{i,t}\,\tilde{A}_{i,t},\; \mathrm{clip}(\rho_{i,t},\, 1-\epsilon,\, 1+\epsilon)\,\tilde{A}_{i,t}\Big)\right], \; \rho_{i,t} = \frac{\pi_\theta(y_{i,t}|x, y_{i,<t})}{\pi_{\theta_{\mathrm{old}}}(y_{i,t}|x, y_{i,<t})}. \tag{22}$$

Algorithm 1 summarizes the procedure, which replaces the uniform sequence-level advantage A^{seq}_i by the token-level $\tilde{A}_{i,t}$, at the cost of one additional forward pass per trajectory.

5 Results

5.1 Experimental Setup

Models and data. We train the methods on DeepScaleR [30], approximately 40K problem-answer pairs drawn from AMC, AIME, MATH, and OlympiadBench. Each problem comes with a ground-truth integer answer that serves both as the verifiable reward and as the oracle context y^* for computing λ_t. We evaluate on two instruction-tuned models of comparable scale, Qwen3-4B-Instruct-2507 [41] and Phi-4-mini-instruct [1].

Benchmarks. For mathematical reasoning we use GSM8K [5], MATH-500 [26], AMC 2023[2], and AIME 2024[3]; for scientific reasoning, GPQA-Diamond [33] and ARC-Challenge [4]; for code generation, LiveCodeBench [16]. The benchmarks are disjoint from the training data, and Pass@1 is averaged over 16 samples per problem.

[2] `https://huggingface.co/datasets/AI-MO/aimo-validation-amc`
[3] `https://huggingface.co/datasets/AI-MO/aimo-validation-aime`

Table 2: The table reports Pass@1 accuracy (%) on seven reasoning benchmarks. The models are trained on DeepScaleR, and the Baseline row is the released checkpoint before RL training. OPPO variants are grouped by the underlying optimization framework and the oracle estimator. Subscript arrows denote improvement over the corresponding baseline.

	Mathematical Reasoning				*Scientific Reasoning*		*Code*
Method	GSM8K	MATH-500	AMC'23	AIME'24	GPQA-D	ARC-C	LCB
Qwen3-4B-Instruct-2507							
Baseline	89.5	70.2	52.5	44.0	37.8	87.1	28.6
GRPO	91.4	72.8	54.0	46.4	39.2	88.0	31.5
Dr.GRPO	91.6	73.0	54.5	46.3	39.4	88.3	31.4
DAPO	92.3	74.2	55.8	47.0	40.3	88.5	32.8
SDPO	92.8	74.6	56.5	48.2	40.0	89.0	32.4
on GRPO							
Self-OPPO	93.0$_{\uparrow 1.6}$	76.0$_{\uparrow 3.2}$	58.5$_{\uparrow 4.5}$	49.5$_{\uparrow 3.1}$	41.6$_{\uparrow 2.4}$	89.5$_{\uparrow 1.5}$	34.2$_{\uparrow 2.7}$
Teacher-OPPO	93.8$_{\uparrow 2.4}$	77.4$_{\uparrow 4.6}$	60.0$_{\uparrow 6.0}$	51.3$_{\uparrow 4.9}$	42.5$_{\uparrow 3.3}$	89.8$_{\uparrow 1.8}$	35.4$_{\uparrow 3.9}$
on DAPO							
Self-OPPO	93.6$_{\uparrow 1.3}$	76.8$_{\uparrow 2.6}$	59.2$_{\uparrow 3.4}$	50.0$_{\uparrow 3.0}$	41.8$_{\uparrow 1.5}$	89.6$_{\uparrow 1.1}$	34.6$_{\uparrow 1.8}$
Teacher-OPPO	**94.5**$_{\uparrow 2.2}$	**78.5**$_{\uparrow 4.3}$	**61.5**$_{\uparrow 5.7}$	**52.2**$_{\uparrow 5.2}$	**43.2**$_{\uparrow 2.9}$	**90.0**$_{\uparrow 1.5}$	**36.2**$_{\uparrow 3.4}$
Phi-4-mini-instruct							
Baseline	88.2	71.4	28.0	9.5	30.8	83.5	20.4
GRPO	89.8	75.0	32.5	12.0	32.4	84.6	23.0
Dr.GRPO	90.0	74.8	33.0	11.7	32.6	84.8	22.8
DAPO	90.5	76.2	34.0	13.3	33.5	85.0	24.2
SDPO	90.8	76.5	35.5	13.0	33.0	85.4	23.8
on GRPO							
Self-OPPO	91.4$_{\uparrow 1.6}$	78.2$_{\uparrow 3.2}$	37.0$_{\uparrow 4.5}$	15.3$_{\uparrow 3.3}$	34.5$_{\uparrow 2.1}$	86.2$_{\uparrow 1.6}$	25.8$_{\uparrow 2.8}$
Teacher-OPPO	92.0$_{\uparrow 2.2}$	80.0$_{\uparrow 5.0}$	39.5$_{\uparrow 7.0}$	17.0$_{\uparrow 5.0}$	35.8$_{\uparrow 3.4}$	86.8$_{\uparrow 2.2}$	27.2$_{\uparrow 4.2}$
on DAPO							
Self-OPPO	91.8$_{\uparrow 1.3}$	78.8$_{\uparrow 2.6}$	38.0$_{\uparrow 4.0}$	16.0$_{\uparrow 2.7}$	35.0$_{\uparrow 1.5}$	86.4$_{\uparrow 1.4}$	26.5$_{\uparrow 2.3}$
Teacher-OPPO	**92.5**$_{\uparrow 2.0}$	**80.8**$_{\uparrow 4.6}$	**40.5**$_{\uparrow 6.5}$	**17.8**$_{\uparrow 4.5}$	**36.4**$_{\uparrow 2.9}$	**87.0**$_{\uparrow 2.0}$	**28.0**$_{\uparrow 3.8}$

Comparison and configuration. Baselines are drawn from the GRPO family: GRPO [10], Dr.GRPO [28], DAPO [43], and SDPO [13]. We also apply OPPO on top of DAPO to isolate whether the Bayesian advantage adds gains beyond optimization mechanics. The teacher-oracle uses Qwen3-32B as the estimator π_ϕ for Qwen3-4B and Phi-4 (14B) for Phi-4-mini. Training uses $2\times$ H200 GPUs and the verl framework [36], with AdamW at learning rate 1×10^{-6}, batch size 32, group size $G = 8$, clip $\epsilon = 0.2$, evidence clip $C = 3.0$, and prior $\alpha = 1$.

5.2 Main Results

Token-level credit improves reasoning. In Table 2, Self-OPPO outperforms GRPO and its optimization variants on the seven benchmarks for both base models, with gains concentrated on competition-level tasks where reasoning chains are longest. The pattern matches the variance argument of Section 3.3: longer sequences accumulate more positions in determined regions where $V_t(1 - V_t) \approx 0$, where the Bayesian advantage suppresses the gradient signal and GRPO does not. The further gains over SDPO, which reads the oracle signal but leaves the belief untracked, show that the running state estimate adds headroom beyond per-token discrimination. The gain persists when the student itself is scaled: on Qwen3-8B-Base under the identical protocol, Self-OPPO raises GRPO from 87.8 to 89.5 on MATH-500, from 78.5 to 81.0 on AMC'23, and from 33.8 to 36.5 on AIME'24 (Appendix F.2).
Oracle quality translates into training gains. Teacher-OPPO improves over Self-OPPO on the seven benchmarks, with the largest gains on the hardest tasks, and since the Bayesian machinery is identical in both variants, the improvement comes from sharper λ_t estimates. Scoring student tokens under a larger same-family teacher approximates a controlled distillation signal, so Self-OPPO remains the cleaner test of the Bayesian mechanism. *The Bayesian advantage complements optimization improvements.* OPPO on top of DAPO yields gains comparable in magnitude to OPPO on GRPO,

Table 3: The table ablates each component of Self-OPPO on Qwen3-4B-Instruct-2507 over four long-chain benchmarks.

Configuration	M-500	AMC	AIME'24	GPQA
Self-OPPO (full)	76.0	58.5	49.5	41.6
remove anchoring	71.8	51.0	43.0	38.5
remove tracking	74.5	55.8	46.8	40.0
remove clipping	75.4	57.5	48.6	41.2
remove prior	75.8	58.0	49.0	41.5
$\log \lambda_t$ only	73.6	55.0	46.5	39.8
GRPO (uniform)	72.8	54.0	46.4	39.2

Table 4: The table reports oracle capacity in (a) and MATH-500 accuracy by response length in (b).

(a) Estimator	Size	M-500	AIME'24
Self	4B	76.0	49.5
Qwen3-8B	8B	76.8	50.2
Qwen3-14B	14B	77.0	50.8
Qwen3-32B	32B	77.4	51.3

(b) Method	0-256	257-512	513-1024	1025+
GRPO	88.5	80.2	68.4	52.0
SDPO	89.0	81.5	70.8	55.2
Self-OPPO	89.2	82.0	73.5	62.8

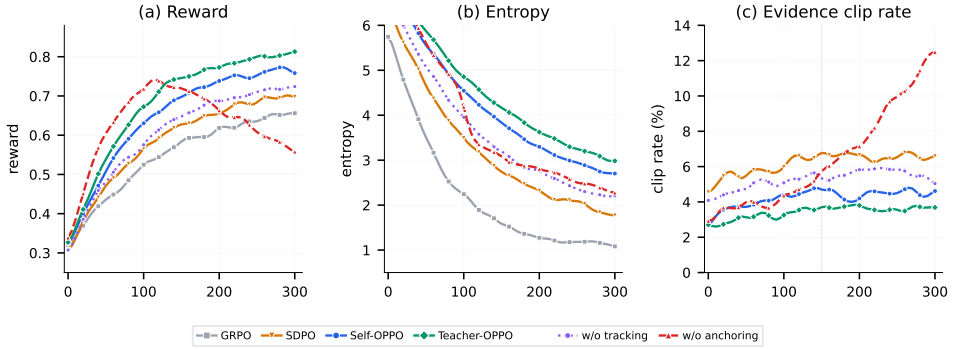

Figure 2: The figure tracks reward, entropy, and evidence clip rate over training for GRPO, SDPO, both OPPO variants, and two ablations. (a) Once anchoring is removed, reward declines after step 150. (b) Removing state tracking lowers the entropy that OPPO otherwise preserves. (c) In the same ablation, the evidence clip rate diverges from step 150 on.

and Teacher-OPPO on DAPO reaches the highest performance on the seven benchmarks for both models. Improved clipping and filtering are therefore complementary to finer credit assignment.

5.3 Ablation and Analysis

Table 3 removes each component of Self-OPPO individually on Qwen3-4B-Instruct. Direction anchoring is the most important: tokens in incorrect trajectories that correlate with y^* produce $\log \lambda_t > 0$ and receive positive advantage, reinforcing flawed reasoning, and AIME'24 performance falls below GRPO once anchoring is removed, mirroring the degradation of pure self-distillation [42]. State tracking ranks second, with the largest drops on AMC and AIME'24 where reasoning chains are longest: absent the $V_t(1 - V_t)$ factor, the positions receive weight proportional to $|\log \lambda_t|$ regardless of trajectory commitment, and longer chains accumulate more saturated positions. The "$\log \lambda_t$ only" row, which removes both state tracking and clipping, sits close to GRPO on the four benchmarks, so per-token oracle evidence on its own contributes little, and the gap to the full method is the joint contribution of state tracking and numerical control.

The training dynamics in Figure 2 corroborate the ablation. Reward ceiling and entropy floor both increase from GRPO through SDPO to Teacher-OPPO, since GRPO applies uniform gradient pressure along the trajectory whereas OPPO applies weak gradients where V_t has saturated and preserves exploration capacity. With anchoring removed, reward rises fastest in the early steps and declines after step 150 as the clip rate rises and $\log \lambda_t$ grows extreme on surface correlations with y^*, the runaway that direction anchoring prevents.

Table 4(a) scales the oracle while the student stays fixed: performance rises monotonically with teacher scale, with the largest jump from 4B to 8B, and the estimator needs only next-token prediction, since the teachers are larger same-family models. Table 4(b) partitions MATH-500 by response

length: the three methods perform comparably on the shortest bin because V_t stays near V_0, and the gap widens with length, as Section 3.3 predicts, while Appendix G reports the sensitivity analyses.

6 Conclusion

OPPO recovers token-level advantages by accumulating oracle-conditioned likelihood ratios into a running Bayesian estimate of trajectory success, concentrating credit where the outcome remains uncertain and adding one extra forward pass and dispensing with learned components. Experiments across mathematical, scientific, and code benchmarks confirm gains that widen on longer sequences and grow with the oracle, and the advantage tracks Monte Carlo credit close to the sampling ceiling.

Limitations. The framework requires ground-truth answers for the training queries, and the recursion models a binary outcome, so graded rewards enter as passing the full test suite and continuous rewards lack an analogue of the Bayes factor (Appendix J). The marginal denominator approximates the failure branch (Appendices D.1 and D.2), and the oracle pass adds 35% to the step time of GRPO in self-oracle mode and 52% with a batched Qwen3-32B teacher (Appendix H.3).

References

[1] Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, et al. Phi-4-reasoning technical report. *arXiv preprint arXiv:2504.21318*, 2025.

[2] Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos Garea, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *The twelfth international conference on learning representations*, 2024.

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

[4] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv:1803.05457v1*, 2018.

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

[6] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[7] Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

[8] Mohammad Ghavamzadeh, Shie Mannor, Joelle Pineau, and Aviv Tamar. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 8(5-6):359–483, 2015.

[9] Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*, 2023.

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

[11] Yiran Guo, Lijie Xu, Jie Liu, Dan Ye, and Shuang Qiu. Segment policy optimization: Effective segment-level credit assignment in rl for large language models. *arXiv preprint arXiv:2505.23564*, 2025.

[12] 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)*, pages 12355–12369, 2025.

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

[14] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[15] Adit Jain and Brendan Rappazzo. Learning to reason with mixture of tokens. *arXiv preprint arXiv:2509.21482*, 2025.

[16] 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. *arXiv preprint arXiv:2403.07974*, 2024.

[17] Samyak Jain, Ayush Agrawal, and Navin Goyal. Towards understanding the optimization landscape of grpo and its variants. In *First Workshop on Foundations of Reasoning in Language Models*, 2025.

[18] Michael I Jordan, Zoubin Ghahramani, Tommi S Jaakkola, and Lawrence K Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

[19] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960.

[20] Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Unlocking rl potential for llm reasoning through refined credit assignment. 2024.

[21] Maxim Khanov, Jirayu Burapacheep, and Yixuan Li. Args: Alignment as reward-guided search. *arXiv preprint arXiv:2402.01694*, 2024.

[22] Alex M Lamb, Anirudh Goyal Alias Parth Goyal, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. Professor forcing: A new algorithm for training recurrent networks. *Advances in neural information processing systems*, 29, 2016.

[23] Yu Li, Tian Lan, and Zhengling Qi. When right meets wrong: Bilateral context conditioning with reward-confidence correction for grpo. *arXiv preprint arXiv:2603.13134*, 2026.

[24] Yu Li, Binxu Li, and Tian Lan. Momentkv: Closing the directional gap in kv cache eviction for long-context inference. *arXiv preprint arXiv:2606.01563*, 2026.

[25] Yu Li, Sizhe Tang, and Tian Lan. Reason in chains, learn in trees: Self-rectification and grafting for multi-turn agent policy optimization. *arXiv preprint arXiv:2604.07165*, 2026.

[26] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023.

[27] Zheng Liu, Mengjie Liu, Siwei Wen, Mengzhang Cai, Bin Cui, Conghui He, and Wentao Zhang. From uniform to heterogeneous: Tailoring policy optimization to every token's nature. *arXiv preprint arXiv:2509.16591*, 2025.

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

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

[30] Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl, 2025. Notion Blog.

[31] Avinash Patil and Aryan Jadon. Advancing reasoning in large language models: Promising methods and approaches. *arXiv preprint arXiv:2502.03671*, 2025.

[32] Penghui Qi, Xiangxin Zhou, Zichen Liu, Tianyu Pang, Chao Du, Min Lin, and Wee Sun Lee. Rethinking the trust region in llm reinforcement learning. *arXiv preprint arXiv:2602.04879*, 2026.

[33] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *First conference on language modeling*, 2024.

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

[35] Guobin Shen, Chenxiao Zhao, Xiang Cheng, Lei Huang, and Xing Yu. Vespo: Variational sequence-level soft policy optimization for stable off-policy llm training. *arXiv preprint arXiv:2602.10693*, 2026.

[36] 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*, pages 1279–1297, 2025.

[37] Mingyang Song and Mao Zheng. A survey of on-policy distillation for large language models. *arXiv preprint arXiv:2604.00626*, 2026.

[38] Jiankai Sun, Chuanyang Zheng, Enze Xie, Zhengying Liu, Ruihang Chu, Jianing Qiu, Jiaqi Xu, Mingyu Ding, Hongyang Li, Mengzhe Geng, et al. A survey of reasoning with foundation models: Concepts, methodologies, and outlook. *ACM Computing Surveys*, 57(11):1–43, 2025.

[39] Richard S Sutton. Learning to predict by the methods of temporal differences. *Machine learning*, 3(1):9–44, 1988.

[40] Ronald J Williams and David Zipser. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280, 1989.

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

[42] 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.

[43] 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. *arXiv preprint arXiv:2503.14476*, 2025.

[44] 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.

[45] Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, et al. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*, 2025.

[46] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

A Extended Related Work

The appendix places OPPO within the wider literature on credit assignment and reinforcement learning for reasoning along three axes, extending the immediate baselines of Section 1.

A.1 Trajectory-Level Optimization and Critic-Based Alternatives

Group Relative Policy Optimization [10] and its descendants form the dominant on-policy paradigm for reasoning. Dr.GRPO [28] fixes the reference policy in the importance ratio denominator to remove the length-bias of the original objective. DAPO [43] introduces asymmetric clipping and filtered sampling. GSPO [46] aggregates probabilities at the sequence level. GMPO [45] replaces the arithmetic-mean group baseline with a geometric mean. Across these variants, the per-token advantage remains a single trajectory-level scalar broadcast uniformly across positions [11], which is the limitation that motivates OPPO. Two recent variants improve the optimizer along other axes: VESPO [35] stabilizes off-policy training under staleness through a variational sequence-level objective, and DPPO [32] replaces ratio clipping with a divergence-based trust region. Neither resolves credit within a trajectory, so both are complementary to the recursion below. The Bayesian recursion of Section 3 is orthogonal to the optimization-mechanic improvements, and Section 5 shows that applying OPPO on top of DAPO yields gains comparable in magnitude to applying it over standard GRPO.

The critic-based alternative is Proximal Policy Optimization [34], which pairs the clipped surrogate with a learned value network $V_\phi(s_t)$ that approximates the on-policy state value of Eq. (3) via temporal-difference regression. VinePPO [20] shows that the learned critic in PPO barely outperforms a random baseline on long-horizon reasoning trajectories, where credit-assignment difficulty grows with sequence length and the regression target is dominated by sparse outcome rewards. The Bayesian value recursion of Section 3 can be read as an analytical alternative to fitting V_ϕ: instead of learning the success probability through bootstrapped regression targets, the recursion computes it in closed form from the per-token oracle ratio, with the running posterior V_t playing the role of V_ϕ at zero parameter cost.

A.2 Critic-Free Token-Level Credit and Distillation-Based Methods

Beyond GRPO and PPO, several critic-free approaches resolve credit at finer granularity than the trajectory level. VinePPO [20] estimates per-step values through Monte Carlo rollouts that branch from each prefix; the estimator is unbiased but requires orders-of-magnitude more inference. TreeRL [12] extends the tree-search idea to multi-branch rollouts. PRIME [7] trains an implicit process reward model that supplies dense token-level rewards but requires online updating to avoid reward hacking. Segment Policy Optimization [11] aggregates descendant outcomes within shared prefixes, sitting between trajectory-level and token-level granularity. Process reward models more broadly [26] train a separate verifier to score intermediate reasoning steps. Compared with these methods, OPPO achieves token-level resolution at the cost of one extra forward pass per trajectory and with neither an auxiliary network nor extra rollouts, by exploiting the privileged y^* context to derive a closed-form per-token signal. The trade-off is that OPPO requires verifiable training-time answers; methods that use Monte Carlo rollouts, learned critics, or step verifiers do not.

The line of work most closely related to OPPO reads a signal from the change in likelihood that a teacher context induces, and it grew out of teacher forcing. Teacher forcing [40] supplies ground-truth prefixes during maximum-likelihood training, and scheduled sampling [3] and professor forcing [22] regulate the exposure bias that the practice induces; in the three methods the conditioning determines what the model observes during training, and the likelihood change is left unread as a per-token signal. Distillation methods were the first to read that change as a signal: on-policy distillation [2, 9, 29] scores student rollouts under a privileged teacher, and the per-token log-ratio between teacher and student drives the imitation gradient token by token. Reasoning-oriented methods carry the ratio into reinforcement learning by conditioning the teacher on the ground-truth answer [44], and RLSD [42] and SDPO [13] treat the resulting answer-conditioned ratio, which measures how much revealing the answer changes the likelihood of each token, as a dense per-token reward applied with uniform weight. A separate line estimates the underlying success probability while dispensing with oracle conditioning: process reward models [26] train a verifier on step-level labels, VinePPO [20] estimates per-step values through Monte Carlo branching, and PRIME [7] learns an implicit process reward

online. OPPO connects the last two lines. As established in Section 2.2, the per-token signal $\log \lambda_t$ is the same quantity in the distillation methods and in OPPO, and the two differ in how the signal is used. Where SDPO and RLSD apply $\log \lambda_t$ as an isolated per-token reward, OPPO reads it as a Bayes factor, accumulates it through the recursion of Eq. (10) into the success probability that the sampling-based and learned estimators approximate, and modulates each contribution by the running state weight $V_t(1 - V_t)$. Section 5 shows that the additional Bayesian machinery yields consistent improvements over SDPO across the seven benchmarks, with the largest gains on long-chain competition-mathematics tasks where the state weight has the most room to redistribute credit, and Appendix H.2 shows that the accumulated signal tracks Monte Carlo credit where the isolated signal does not.

A.3 Methodological Connections

The recursion of Section 3.1 is structurally a Bayesian filter over a binary latent variable (eventual success), with the per-token oracle ratio playing the role of the observation likelihood. The closed-form additivity in log-odds space is a standard property of Bayes' rule applied to binary outcomes [19, 18]. The novelty lies in identifying the filter with token-level RL credit assignment: the running posterior V_t coincides with the success probability targeted by the on-policy state value of Eq. (3), and the one-step posterior change $V_{t+1} - V_t$ recovers the temporal-difference advantage $A_t = Q_t - V_t$ in the perfect-oracle limit. The connection to TD learning [39] is that OPPO provides an analytical, non-bootstrapped value estimator in a regime where bootstrapping is expensive (full reasoning rollout) and the target signal is sparse (binary terminal reward). Bayesian RL methods that maintain explicit posterior beliefs over reward or value functions [8] typically require either conjugate priors or posterior-sampling machinery; the closed-form structure of the binary-outcome posterior here sidesteps both. The framework is therefore best understood as a Bayesian filtering interpretation of the existing on-policy distillation primitive, not as a fully Bayesian reinforcement-learning method.

A separate line of work uses learned or implicit value estimates to guide decoding at inference time, including verifier-augmented decoding [6] and value-guided sampling [21]. OPPO operates entirely at training time: it modifies how trajectory-level reward is redistributed across tokens during gradient updates, and the deployed policy dispenses with oracle context, verifiers, and tree search. The interpretation of V_t as a running success probability does, however, suggest an inference-time analogue: the same recursion could in principle be applied with a self-oracle estimator at decoding time to weight token candidates by their belief contribution. Such a use of the framework is outside the scope of this paper and is left for future work. We note also that token-level credits learned via OPPO at training time may align with regions where verifier-guided decoding methods place high mass at inference, since both quantities track the model's running belief about eventual success; an empirical correlation analysis is similarly outside scope.

B Detailed Derivations

B.1 Bayesian Update Derivation

The full derivation of the value recursion proceeds from the definition $V_t = P(R{=}1 \mid x, y_{1:t-1})$. Bayes' theorem gives the posterior after observing y_t:

$$V_{t+1} = P(R{=}1 \mid x, y_{1:t}) = \frac{P(y_t \mid x, y_{<t}, R{=}1) \cdot V_t}{P(y_t \mid x, y_{<t})}. \tag{23}$$

The denominator expands by the law of total probability,

$$P(y_t \mid x, y_{<t}) = P(y_t \mid x, y_{<t}, R{=}1)\, V_t + P(y_t \mid x, y_{<t}, R{=}0)\, (1 - V_t). \tag{24}$$

Substituting Modeling Choice 3.2 into the numerator and Modeling Choice 3.2 into the second denominator term, then dividing numerator and denominator by $\pi(y_t \mid x, y_{<t})$, yields

$$V_{t+1} = \frac{\lambda_t\, V_t}{\lambda_t\, V_t + (1 - V_t)}, \qquad \lambda_t = \frac{\hat{\pi}(y_t \mid x, y_{<t}, y^*)}{\hat{\pi}(y_t \mid x, y_{<t})}, \tag{25}$$

which is Eq. (10) of the main text with λ_t in place of $\lambda_t^\star$.

B.2 Logit-Space Reduction

Define the running log-odds $\ell_t = \log \frac{V_t}{1-V_t}$. From Eq. (25),

$$\frac{V_{t+1}}{1-V_{t+1}} = \frac{\lambda_t V_t/(\lambda_t V_t + 1 - V_t)}{(1-V_t)/(\lambda_t V_t + 1 - V_t)} = \frac{\lambda_t V_t}{1-V_t} = \lambda_t \cdot \frac{V_t}{1-V_t}, \tag{26}$$

so taking logarithms collapses the multiplicative recursion in V_t to additive accumulation in ℓ_t:

$$\ell_{t+1} = \log \lambda_t + \ell_t, \qquad \ell_t = \ell_0 + \sum_{i=1}^{t-1} \log \lambda_i, \qquad V_t = \sigma(\ell_t) = \frac{1}{1+e^{-\ell_t}}. \tag{27}$$

B.3 Advantage Closed Form

The token-level advantage $A_t = V_{t+1} - V_t$ admits a closed-form expression. Subtracting V_t from Eq. (25) and clearing the common denominator,

$$A_t = \frac{\lambda_t V_t - V_t(\lambda_t V_t + 1 - V_t)}{\lambda_t V_t + 1 - V_t} = \frac{\lambda_t V_t - \lambda_t V_t^2 - V_t + V_t^2}{\lambda_t V_t + 1 - V_t} = \frac{V_t(1-V_t)(\lambda_t - 1)}{\lambda_t V_t + 1 - V_t}. \tag{28}$$

The first-order approximation follows from two small-quantity expansions when $|\lambda_t - 1| \ll 1$. The denominator satisfies $\lambda_t V_t + 1 - V_t = 1 + (\lambda_t - 1)V_t \approx 1$, and the numerator satisfies $\lambda_t - 1 \approx \log \lambda_t$ via the Taylor expansion $e^x - 1 \approx x$. Substituting both gives

$$A_t \;\approx\; V_t(1-V_t) \cdot \log \lambda_t, \tag{29}$$

which is Eq. (18) in the main text.

B.4 Telescoping Property

Using the identity $A_t = \sigma(\ell_{t+1}) - \sigma(\ell_t)$ from Eq. (16), the trajectory sum telescopes:

$$\sum_{t=1}^{T} A_t = \sum_{t=1}^{T} [\sigma(\ell_{t+1}) - \sigma(\ell_t)] = \sigma(\ell_{T+1}) - \sigma(\ell_1). \tag{30}$$

At the terminal state, the Bayesian posterior should converge to $V_{T+1} = R$, giving $\sigma(\ell_{T+1}) = R$ in the limit of an unconstrained recursion. With $\sigma(\ell_1) = V_0$ at the start, the budget identity becomes

$$\sum_{t=1}^{T} A_t \;\approx\; R - V_0. \tag{31}$$

Evidence clipping prevents ℓ_{T+1} from reaching $\pm\infty$ in practice, so $\sigma(\ell_{T+1})$ approximates R and does not equal it. The empirical residual on 200 MATH-500 trajectories has mean 0.007 and 95th percentile 0.018 (Figure 1(c)), confirming that the deviation from the exact identity is small in aggregate.

B.5 Lipschitz and Range Bounds on the Advantage

The sigmoid derivative satisfies $\sigma'(x) = \sigma(x)(1-\sigma(x)) \le \frac{1}{4}$ for all x, with equality at $x = 0$. The mean value theorem applied to σ yields the Lipschitz bound, while the fact that σ maps $\mathbb{R}$ into the open interval $(0, 1)$ yields a universal range bound:

$$|A_t| = \big|\sigma(\ell_t + \log \lambda_t) - \sigma(\ell_t)\big| \;\le\; \tfrac{1}{4}\,|\log \lambda_t|, \qquad |A_t| \;<\; 1, \tag{32}$$

for all real ℓ_t and $\log \lambda_t$. The two combine to give $|A_t| \le \min\big(\frac{1}{4}|\log \lambda_t|, 1\big)$, with the Lipschitz bound active when $|\log \lambda_t| < 4$ and the range bound active otherwise. With evidence clipping $|\log \hat{\lambda}_t| \le C$, the per-token contribution is further bounded by $|A_t| \le \min(C/4, 1)$. For the manuscript's choice $C = 3.0$, the Lipschitz bound is binding and gives $|A_t| \le 0.75$.

C Variance Reduction Proof

We provide the full derivation of the bound in Eq. (19), with the caveat already noted in Section 3.3: the uncorrelated-score-vector assumption fails for autoregressive language models, so the bound reads as a directional indication and not as a quantitative guarantee for practical training.

Proposition 1. *Let $\hat{g} = \sum_{t=1}^{T} w_t h_t$ denote the policy gradient estimator, where $h_t = \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t})$ is the score vector at position t. Suppose $\mathrm{Cov}[h_s, h_t] = 0$ for all $s \neq t$, and consider two weighting schemes,*

$$w_t^{\mathrm{blind}} = \log \lambda_t, \qquad w_t^{\mathrm{Bayes}} = V_t(1 - V_t) \log \lambda_t.$$

Then $\mathrm{Var}[\hat{g}^{\mathrm{blind}}] \geq \mathrm{Var}[\hat{g}^{\mathrm{Bayes}}]$, with the gap bounded below by

$$\begin{aligned} \mathrm{Var}[\hat{g}^{\mathrm{blind}}] - \mathrm{Var}[\hat{g}^{\mathrm{Bayes}}] \;&\geq\; (1 - \gamma^2) \sum_{t \in T_{\mathrm{det}}} (\log \lambda_t)^2 \; \mathrm{Var}[h_t], \\ T_{\mathrm{det}} &= \{\, t : |\log \lambda_t| \geq \delta,\; V_t(1 - V_t) < \gamma \,\}, \end{aligned} \tag{33}$$

for all thresholds $\delta, \gamma > 0$.

Proof. Under the uncorrelated-score assumption, the variance of each estimator decomposes additively across positions:

$$\mathrm{Var}[\hat{g}^{\mathrm{blind}}] = \sum_{t=1}^{T} (\log \lambda_t)^2 \, \mathrm{Var}[h_t], \qquad \mathrm{Var}[\hat{g}^{\mathrm{Bayes}}] = \sum_{t=1}^{T} V_t^2 (1 - V_t)^2 (\log \lambda_t)^2 \, \mathrm{Var}[h_t]. \tag{34}$$

The variance gap is therefore

$$\Delta = \mathrm{Var}[\hat{g}^{\mathrm{blind}}] - \mathrm{Var}[\hat{g}^{\mathrm{Bayes}}] = \sum_{t=1}^{T} \left[1 - V_t^2 (1 - V_t)^2\right] (\log \lambda_t)^2 \, \mathrm{Var}[h_t]. \tag{35}$$

Since $V_t \in [0, 1]$ implies $V_t(1 - V_t) \leq \frac{1}{4}$, every factor satisfies $V_t^2(1 - V_t)^2 \leq \frac{1}{16}$, so $1 - V_t^2(1 - V_t)^2 \geq \frac{15}{16} > 0$ and $\Delta \geq 0$. To obtain the interpretable lower bound, restrict the sum to T_{det} and use $V_t^2(1 - V_t)^2 < \gamma^2$ on that subset:

$$\Delta \;\geq\; \sum_{t \in T_{\mathrm{det}}} \left[1 - V_t^2 (1 - V_t)^2\right] (\log \lambda_t)^2 \, \mathrm{Var}[h_t] \;\geq\; (1 - \gamma^2) \sum_{t \in T_{\mathrm{det}}} (\log \lambda_t)^2 \, \mathrm{Var}[h_t]. \tag{36}$$

□

The bound is informative when T_{det} is large, which happens in two regimes. The first is long sequences, where V_t saturates after a pivotal decision and many subsequent positions retain nonzero $\log \lambda_t$ from answer-correlated content. The second is easy or unsolvable queries, where V_0 already sits near 0 or 1 before generation starts, so $V_t(1 - V_t) \approx 0$ throughout the trajectory while $\log \lambda_t$ remains nonzero. In both regimes the Bayesian state weight collapses to near zero on the saturating positions, so the corresponding gradient contributions are suppressed in $\hat{g}^{\mathrm{Bayes}}$ but not in $\hat{g}^{\mathrm{blind}}$. The empirical sequence-length stratification in Table 4(b) is consistent with the predicted pattern: the gap between Self-OPPO and the state-blind baselines widens monotonically with response length.

D Oracle Quality Analysis

D.1 Bias Under Imperfect Oracle

When $\epsilon_t > 0$, the estimated λ_t differs from the true posterior ratio. The deviation has a clean interpretation. Let $p_t = P(y_t \mid x, y_{<t}, R{=}1)$ denote the true success-conditional probability and $q_t = \hat{\pi}(y_t \mid x, y_{<t}, y^*)$ the oracle estimate. Define the true ratio λ_t^* and the estimated ratio $\hat{\lambda}_t$ on the same denominator:

$$\lambda_t^* = \frac{p_t}{\pi(y_t \mid x, y_{<t})}, \qquad \hat{\lambda}_t = \frac{q_t}{\pi(y_t \mid x, y_{<t})}, \qquad \log \hat{\lambda}_t - \log \lambda_t^* = \log q_t - \log p_t. \tag{37}$$

The KL definition $\epsilon_t = \mathbb{E}_{y_t \sim p_t}[\log p_t - \log q_t]$ gives the expected log-ratio bias under the true success-conditional distribution:

$$\mathbb{E}_{y_t \sim p_t}\left[\log \hat{\lambda}_t - \log \lambda_t^*\right] = -\,\epsilon_t. \tag{38}$$

The bias is non-positive on average, so the oracle estimate is conservative and underestimates the true evidence for success. Stronger oracles reduce ϵ_t and the magnitude of the bias, which is consistent with the monotonic improvement in Table 4(a).

D.2 Exact Failure Branch

Modeling Choice 3.2 approximates the failure-conditional distribution by the marginal under the same scorer. The direct alternative conditions the denominator on a wrong answer $\tilde{y}$ sampled from the incorrect rollouts of the group, which estimates $P(y_t \mid x, y_{<t}, R{=}0)$ free of the mixture of Eq. (13) and requires a second answer-conditioned forward pass per trajectory. Table 5 reports both forms for Self-OPPO on Qwen3-4B under the protocol of Section 5.1. The exact form is more accurate by 0.3 to 0.4 points on the three benchmarks and raises the step time from 16.8 to 21.2 seconds, since the marginal reuses the unconditioned pass that the gradient update already computes. The two evidence sequences $\log \lambda_t$ correlate at 0.91 over 200 MATH-500 trajectories, so the marginal recovers most of the accuracy at half of the oracle cost, and the main text adopts it as the default.

Table 5: The table compares the marginal and wrong-answer denominators for Self-OPPO on Qwen3-4B. Time per step is measured on 2× H200 GPUs.

Denominator	MATH-500	AMC'23	AIME'24	Time/step (s)	vs. GRPO
Wrong answer $\tilde{y}$	76.3	58.9	49.8	21.2	+71%
Marginal (default)	76.0	58.5	49.5	16.8	+35%
GRPO	72.8	54.0	46.4	12.4	—

D.3 Same-Family Teacher Choice

The oracle quality ϵ_t depends not only on model capacity but also on distributional alignment between the estimator and the true posterior. Cross-family teachers, for example a Llama model scoring Qwen trajectories, can produce large ϵ_t despite high capacity because tokenization, positional encoding, and pre-training corpus differ. The ratio structure $\log \lambda_t = \log \hat{\pi}(y_t \mid x, y_{<t}, y^*) - \log \hat{\pi}(y_t \mid x, y_{<t})$ cancels many model-specific biases through subtraction, but residual mismatch in the conditional shape still inflates ϵ_t. Same-family, larger-scale models are the recommended choice for the teacher-oracle role because the cancellation is most effective when the two scoring distributions share the same underlying parameterization. The experiments follow this choice and pair Qwen3-4B with Qwen3-32B and Phi-4-mini with Phi-4 (14B).

E Connection to Distillation

E.1 On-Policy Distillation as a Special Case

Section 2.2 introduced the on-policy distillation objective $\mathcal{L}_{\mathrm{OPD}}$ with the oracle-conditioned teacher $\pi_T(\cdot \mid x, y_{<t}) \equiv \hat{\pi}(\cdot \mid x, y_{<t}, y^*)$, and showed that the per-token gradient at the sampled token y_t reduces to a policy-gradient form with weight $\log \lambda_t$. Repeating the conclusion in the present notation,

$$\begin{aligned} \nabla_\theta \mathcal{L}_{\mathrm{OPD}}(\theta)\big|_{y_t} &= -\log \lambda_t \cdot \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t}), \\ \log \lambda_t &= \log \hat{\pi}(y_t \mid x, y_{<t}, y^*) - \log \pi_\theta(y_t \mid x, y_{<t}). \end{aligned} \tag{39}$$

Pure on-policy distillation is therefore a policy gradient with advantage $\log \lambda_t$, which the "$\log \lambda_t$ only" row of Table 3 reproduces with direction anchoring retained. The Bayesian advantage in Eq. (18) differs from the distillation gradient in two specific ways. The state weight $V_t(1 - V_t)$ concentrates the gradient on positions where the outcome remains uncertain, so determined regions receive negligible updates. Direction anchoring binds the signs of the token-level advantages to the sequence-level GRPO advantage, so the environment reward retains exclusive control over which trajectories are reinforced or penalized. The two modifications are independent: removing either one degrades performance, but the failure mechanisms differ, as documented in the ablation analysis of Section 5.

E.2 Spectrum of Methods

The factorization $A_t \approx V_t(1 - V_t) \cdot \log \lambda_t$ organizes a spectrum of methods along three axes: the per-token signal carried into the gradient, whether a state-tracking factor weights the signal, and which quantity controls the sign of each trajectory's update.

Table 6: The table places five methods on the spectrum induced by the Bayesian factorization. Each row corresponds to one setting in the ablation of Table 3: GRPO matches "uniform", SDPO matches "$\log \lambda_t$ only" with environment-anchored direction, and pure distillation (OPSD) matches "remove anchoring". The ablation rows share the OPPO implementation, so their numbers differ from the SDPO row of Table 2.

Method	Per-token signal	State tracking	Direction
GRPO [10]	uniform A_i^{seq}	—	environment reward
Anchored distillation (SDPO) [13]	$\lvert \log \lambda_t \rvert$	—	environment reward
Self-OPPO (ours)	$V_t(1-V_t) \lvert \log \lambda_t \rvert$	Bayesian	environment reward
Teacher-OPPO (ours)	$V_t(1-V_t) \lvert \log \lambda_t \rvert$, lower ϵ_t	Bayesian	environment reward
Pure distillation (OPSD) [44]	$\log \lambda_t$	—	teacher

Reading the table from top to bottom moves through the design space. GRPO sits at the trajectory-level extreme, lacking both per-token discrimination and state tracking. Anchored distillation introduces $\log \lambda_t$ as a per-token signal but applies it uniformly across positions. OPPO adds the Bayesian state weight $V_t(1 - V_t)$, modulating the per-token signal by the running belief about success. The self-oracle and teacher-oracle variants of OPPO differ only in the quality of the $\log \lambda_t$ estimate, parameterized by ϵ_t. Pure distillation drops direction anchoring and lets $\log \lambda_t$ control the sign of the update, which is the failure mode reproduced in the "remove anchoring" row of Table 3.

F Extended Experimental Results

F.1 Full Results on Phi-4-mini

Table 7 reports the complete results for Phi-4-mini across the seven benchmarks and the ablation configurations, complementing the Phi-4-mini-instruct rows of Table 2. The pattern matches the Qwen3-4B case: OPPO improves over the four baselines on the seven benchmarks, with the largest absolute gains on AMC'23 and the largest relative gains on competition-level tasks where the base model is weakest.

Table 7: Full results on Phi-4-mini (base). The methods are trained on DeepScaleR.

Method	GSM8K	M-500	AMC'23	AIME'24	GPQA-D	ARC-C	LCB
Base model	68.0	42.5	22.0	3.3	25.6	72.8	14.5
+ GRPO	76.8	55.2	35.0	8.3	29.4	78.6	19.2
+ Dr.GRPO	77.0	55.4	35.2	8.5	29.6	78.8	19.4
+ DAPO	78.2	57.0	37.5	10.0	30.4	79.4	20.2
+ SDPO	78.6	57.8	38.0	10.5	30.8	79.8	20.6
+ Self-OPPO	80.4	60.2	41.5	13.3	32.4	81.0	22.4
+ Teacher-OPPO	81.5	62.0	43.5	15.0	33.6	81.6	23.6
+ Dr.GRPO + OPPO	80.6	59.8	40.8	12.8	32.0	80.8	22.0
+ DAPO + OPPO	82.0	61.5	43.0	14.5	33.2	81.4	23.2

F.2 Scaling the Student

Table 4(a) scales the oracle while the student stays fixed. Table 8 scales the student itself, training Qwen3-8B-Base under the identical protocol of Section 5.1 with the self-oracle. The gains persist at 8B on MATH-500, AMC'23, and AIME'24, with the largest on AIME'24.

Table 8: Self-OPPO on Qwen3-8B-Base under the protocol of Table 2.

Qwen3-8B-Base	MATH-500	AMC'23	AIME'24
GRPO	87.8	78.5	33.8
Self-OPPO	89.5$_{\uparrow 1.7}$	81.0$_{\uparrow 2.5}$	36.5$_{\uparrow 2.7}$

F.3 Component Ablation on Phi-4-mini

The component ablation in Table 9 repeats the analysis of Section 5 on the Phi-4-mini base model of Table 7. The ranking of components reproduces the Qwen3-4B result: direction anchoring is the most critical, followed by state tracking, with clipping and the prior contributing modestly. The "remove anchoring" row falls below GRPO on AIME'24, reproducing the inversion-failure pattern documented in the main text.

Table 9: Component ablation on Phi-4-mini (base). The ranking of components matches the Qwen3-4B result, with direction anchoring the most critical and state tracking second.

Configuration	M-500	AMC	AIME'24	GPQA
Self-OPPO (full)	60.2	41.5	13.3	32.4
remove anchoring	52.0	30.5	5.0	27.8
remove tracking	57.5	38.0	10.8	30.6
remove clipping	59.5	40.5	12.5	32.0
remove prior	59.8	41.0	13.0	32.2
$\log \lambda_t$ only	56.5	36.5	9.5	29.8
GRPO (uniform)	55.2	35.0	8.3	29.4

F.4 Pass@K Results

Table 10 reports Pass@K for $K \in \{1, 4, 8, 16\}$ on MATH-500 and AIME'24. The Pass@1 column repeats the 16-sample values of Table 2, and larger K is estimated from 32 samples per problem. OPPO improves Pass@K at $K = 1, 4, 8, 16$ on both benchmarks, indicating that the gains over GRPO and SDPO are not driven by sharpening a single dominant mode at the expense of diversity. The relative improvement is largest at small K on AIME'24, where the base policy spreads probability across many incorrect reasoning paths and the Bayesian advantage redistributes credit toward the few that succeed.

Table 10: Pass@K on Qwen3-4B. OPPO outperforms GRPO and SDPO at $K = 1, 4, 8, 16$ on both benchmarks.

	MATH-500				AIME'24			
Method	@1	@4	@8	@16	@1	@4	@8	@16
GRPO	72.8	82.0	86.5	90.2	46.4	60.0	66.7	73.3
SDPO	74.6	83.5	87.8	91.0	48.2	62.2	68.9	75.6
Self-OPPO	76.0	85.2	89.4	92.5	49.5	64.4	71.1	77.8
Teacher-OPPO	77.4	86.5	90.2	93.0	51.3	66.7	73.3	80.0

G Hyperparameter Sensitivity

We study the sensitivity of Self-OPPO on Qwen3-4B to the three method-specific hyperparameters: evidence clip bound C, Beta prior strength α, and group size G.

G.1 Evidence Clip Bound C

The clip bound C in Eq. (20) controls the per-token contribution to the running log-odds. Table 11 reports performance across five settings.

Table 11: Sensitivity to evidence clip bound C on Qwen3-4B.

C	1.0	2.0	3.0	5.0	10.0
MATH-500	74.3	75.5	76.0	75.7	75.0
AIME'24	46.8	48.8	49.5	49.1	47.8

Setting C too small ($C = 1.0$) truncates the evidence at pivotal tokens, compressing the dynamic range of A_t toward the uniform-advantage regime. Setting C too large ($C = 10.0$) lets single tokens dominate the log-odds, saturating V_t prematurely and silencing later positions. Performance is stable across $C \in [2.0, 5.0]$, and the choice $C = 3.0$ used in the main experiments sits at the center of the stable region.

G.2 Beta Prior Strength α

The Beta prior in Eq. (20) smooths the empirical group success rate before converting it to log-odds. Table 12 reports performance across five settings.

Table 12: Sensitivity to Beta prior strength α on Qwen3-4B.

α	0.1	0.5	1.0	2.0	5.0
MATH-500	75.5	75.8	76.0	75.9	75.3
AIME'24	48.6	49.1	49.5	49.3	48.3

Small α makes $\hat{V}_0$ responsive to the empirical success rate but produces extreme initial log-odds in all-correct or all-incorrect groups. Large α pulls $\hat{V}_0$ toward 0.5 regardless of the group composition and weakens the prompt-difficulty adaptation discussed in Section 3.3. Performance is stable across $\alpha \in [0.5, 2.0]$, and $\alpha = 1$ corresponds to a Laplace prior that is the standard choice for a Bernoulli outcome.

G.3 Group Size G

The group size G controls both the granularity of the prior $\hat{V}_0 = (k + \alpha)/(G + 2\alpha)$ and the variance of the GRPO sequence-level advantage that drives direction anchoring. Table 13 reports performance across four settings.

Table 13: Sensitivity to group size G on Qwen3-4B, reported as MATH-500 Pass@1. OPPO improves over GRPO at $G = 4, 8, 16, 32$.

G	4	8	16	32
GRPO	71.0	72.8	73.7	74.2
Self-OPPO	74.0	76.0	76.7	77.0
Δ	+3.0	+3.2	+3.0	+2.8

Larger groups improve both GRPO and OPPO by reducing the variance of $\hat{V}_0$. The improvement of OPPO over GRPO is stable across group sizes, indicating that the token-level credit assignment provides gains complementary to the group-based prior estimation and independent of a particular group size.

H Additional Analysis

H.1 Value Estimation Accuracy

The state-weight argument in Section 3.3 requires that intermediate V_t track the evolving success probability, not merely that the terminal V_{T+1} converge to R. We assess calibration on 200 MATH-500 trajectories from Qwen3-4B at four positions: $t = T/4, T/2, 3T/4, T$. Calibration at each position is measured by the Brier score

$$\mathrm{BS}(t) = \frac{1}{N} \sum_{i=1}^{N} \left(V_t^{(i)} - R^{(i)} \right)^2, \tag{40}$$

which is zero only when $V_t^{(i)} = R^{(i)}$ for every trajectory and penalizes both miscalibration and low resolution.

Table 14: Brier score of V_t at four positions along the trajectory, on 200 MATH-500 trajectories. Lower is better.

Oracle estimator	$T/4$	$T/2$	$3T/4$	T
Self (4B)	0.071	0.058	0.049	0.042
Teacher (32B)	0.054	0.043	0.034	0.028

The Brier score decreases monotonically along the trajectory under both estimators, consistent with the recursion sharpening the posterior as more evidence is observed. The teacher estimator attains the lower score at the four positions, so the state weight $V_t(1 - V_t)$ acts on beliefs that already track the outcome before the trajectory ends.

H.2 Agreement with Monte Carlo Credit

The advantage of Eq. (16) is validated against a Monte Carlo estimate of per-token credit in the manner of Math-Shepherd [26]. On 200 MATH-500 trajectories from Qwen3-4B, the 8 highest-entropy positions serve as segment boundaries, giving 9 segments per trajectory. At each boundary we sample 32 completions at temperature 1.0 and record the empirical success rate as the Monte Carlo value at that point; the Monte Carlo credit of a segment is the change in this value across the segment. For each method we correlate the total advantage it assigns within a segment against the Monte Carlo credit, and Table 15 reports the Spearman correlation and the sign agreement. The last row is the reliability ceiling of the ground truth itself, obtained by splitting the 32 completions into two halves of 16 and correlating the two independent Monte Carlo estimates, so it bounds the correlation an estimator can reach at this sampling budget. GRPO assigns the tokens of a trajectory the same advantage and misses the per-token structure. The per-token evidence used as a dense reward recovers a moderate correlation, and accumulating the same evidence into the Bayesian posterior raises it to 81% of the ceiling for the self-oracle and 91% for the teacher-oracle. Since the dense-reward row and the Self-OPPO row read the identical per-token signal, the gap between them isolates the contribution of the state weight.

Table 15: The table reports the agreement between assigned segment advantage and Monte Carlo credit on 200 MATH-500 trajectories from Qwen3-4B.

Advantage	Spearman ρ	Sign agreement
GRPO (uniform)	0.02	51.3%
Per-token $\log \lambda_t$ as a dense reward	0.41	68.4%
Self-OPPO	0.63	79.6%
Teacher-OPPO	0.71	84.2%
Monte Carlo reliability ceiling (split half)	0.78	88.1%

H.3 Computational Overhead

OPPO requires one additional forward pass per trajectory for oracle evaluation, and the credit-assignment step itself is a scan of T scalar operations per trajectory once $\log \lambda_t$ is available, so its cost does not appear at the resolution of per-step timing. Table 16 reports wall-clock time per training step and peak memory on $2\times$ H200 GPUs.

Table 16: The table reports wall-clock time per training step and peak memory on $2\times$ H200 GPUs for Qwen3-4B, whose teacher-oracle is Qwen3-32B.

Method	Forward passes	Time/step (s)	Peak memory (GB)	vs. GRPO
GRPO	rollout + policy	12.4	61.2	—
Self-OPPO	rollout + oracle + policy	16.8	64.8	+35%
Teacher-OPPO, sequential	rollout + oracle (32B) + policy	24.2	138.5	+95%
Teacher-OPPO, batched	rollout + oracle (32B, batched) + policy	18.9	138.5	+52%

The self-oracle adds 35% to the step time and 6% to peak memory, because the oracle pass uses the same weights as the policy and skips generation, so the cost is a single teacher-forced pass over already-sampled tokens. The sequential teacher-oracle figure comes from an implementation in which the 32B scorer evaluates trajectories one at a time. Since the oracle pass is teacher-forced and skips autoregressive decoding, it is compute-bound and parallelizes across the group, and batching the evaluation over the G trajectories of a query reduces the step time to 18.9 seconds. The scorer stays frozen, so it can be placed on dedicated devices and overlapped with rollout generation. The overhead falls entirely in the advantage-computation phase and leaves rollout generation, the autoregressive bottleneck whose key-value cache grows with chain length [24], unchanged, so the relative overhead shrinks as generation length grows.

I Implementation Details

I.1 Training Configuration

The experiments use the verl framework [36] on 2× H200 GPUs with the configuration in Table 17.

Table 17: The table lists the training and evaluation settings used for the main-text experiments.

Hyperparameter	Value
Optimizer	AdamW
Learning rate	1×10^{-6}
Weight decay	0.01
Warmup steps	10
Learning-rate schedule	cosine decay
Global batch size	32
Group size G	8
Maximum response length	8192 tokens
Maximum prompt length	1024 tokens
Rollout temperature	1.0
PPO clip ϵ	0.2
Evidence clip C	3.0
Beta prior α	1
Training epochs	1
Gradient checkpointing	enabled
Precision	bfloat16
Evaluation samples per problem	16 (Pass@1), 32 (Pass@K)

I.2 Oracle Prompt Format

For self-oracle evaluation, the ground-truth answer y^* is appended to the prompt before the model's response, in the format

```
[Standard prompt]
Question: {question}
[Oracle augmentation]
The answer is {y*}.
[Model response]
{y_1, y_2, ..., y_T}
```

The oracle-conditioned forward pass scores the same response tokens $y_{1:T}$ under the augmented prompt, while the standard forward pass scores them under the original prompt. The log-likelihood ratio $\log \lambda_t$ is the per-token difference between the two scores. Generation is skipped during oracle evaluation, and the standard pass coincides with the forward pass of the policy update, so the self-oracle adds one forward pass per trajectory.

I.3 Reward Function

We use exact-match reward: $R = 1$ if the extracted numerical answer matches the ground-truth integer and $R = 0$ otherwise. Answer extraction follows the standard boxed-answer format used in

DeepScaleR. Partial-credit and format-bonus rewards are left out, so the binary outcome reward is the only environment signal driving direction anchoring.

J Limitations and Partial Credit

The present study has four limitations. First, the oracle context is built from the ground-truth answer, so the training queries must carry one; queries with an answer that cannot be verified carry unverifiable evidence and are excluded. Second, the recursion of Eq. (10) accumulates evidence in log-odds space, and the log-odds are defined for a Bernoulli outcome, so the latent variable is binary. Third, the marginal denominator of Modeling Choice 3.2 approximates the failure branch; Appendix D.1 bounds the resulting bias by the success-branch KL, and Appendix D.2 measures the gap to the exact form at 0.3 to 0.4 points. Fourth, the oracle pass adds between 35% and 52% to the step time of GRPO, as Appendix H.3 reports.

The binary outcome does not confine the framework to a single test. Where a response is graded by m unit tests, defining success as passing all m keeps the recursion unchanged, and the LiveCodeBench evaluation of Table 2 already uses that definition. A graded token-level advantage follows from running m parallel recursions, one per test with its own oracle context, and aggregating the resulting log-odds across tests, while each individual recursion remains binary. Continuous or rubric-scored rewards require a different treatment, since the Bayes factor of Eq. (9) lacks a direct analogue, and we leave that case open. The oracle context generalizes more readily than the reward: for code, y^* is a reference solution in place of an integer, and the ratio of Eq. (14) is unchanged. The models of Table 2 are trained on mathematics alone and evaluated zero-shot on LiveCodeBench, GPQA-Diamond, and ARC-Challenge, where the gains persist.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Section 1 state four contributions, each substantiated in the body: the Bayesian value recursion in Section 3.1, the closed-form factorization of the advantage in Section 3.3, the algorithmic integration in Section 4 (Algorithm 1), and the empirical evaluation in Section 5 (Table 2).

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations are discussed in Section 6, with the bias analysis in Appendix D.1 and the overhead breakdown in Appendix H.3 providing supporting detail.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All theoretical results are stated formally with explicit assumptions and proved in Appendix B, including the Bayesian update derivation, the closed-form advantage, the telescoping budget identity, the Lipschitz bound, and the directional variance bound (Proposition 1).

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Section 5 specifies the base models, training data, benchmarks, and optimizer configuration; Appendix I provides the full hyperparameter table (Table 17), the oracle prompt format, and the reward function.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: A code repository is provided in the supplementary material, and all training and evaluation datasets are publicly available; the base models and the teacher-oracle are released open-weight.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The training and evaluation protocols are reported in Section 5, the full hyperparameter configuration in Appendix I.1 (Table 17), and the sensitivity to method-specific hyperparameters in Appendix G.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [N/A]

Justification: Pass@1 is averaged over 16 sampling runs as is standard in the reasoning-benchmark literature, and Section 5 reports the typical run-to-run sampling variance to allow readers to identify which gains lie outside the noise band; multiple training seeds per cell are not feasible at the scales studied.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 5 reports training on $2\times$ H200 GPUs with the verl framework, and Appendix H.3 (Table 16) reports per-step wall-clock time for GRPO, Self-OPPO, and Teacher-OPPO.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We have reviewed the NeurIPS Code of Ethics, and the work uses publicly available open-weight models and reasoning benchmarks, involves no human subjects, and preserves anonymity in submission.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.

- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: OPPO is a foundational training-time credit-assignment method for reinforcement learning with verifiable rewards and has no direct path to specific applications or deployments; any downstream societal impact derives from the underlying base models rather than from the credit-assignment mechanism introduced here.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper releases only the training-procedure code; no pretrained model weights, scraped datasets, or other dual-use artifacts are introduced.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All external assets (the Qwen3 and Phi-4 base models, the DeepScaleR training set, the seven evaluation benchmarks, and the verl framework) are cited at first use and used within the terms of their original release.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The OPPO training code is released as an extension to the verl framework with a README, configuration files for the Self-OPPO and Teacher-OPPO variants, and the oracle prompt template documented in Appendix I.2.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.

- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve human subjects, so IRB approval was not required.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs in the paper are the experimental subjects (the policies being trained) rather than tools used to develop the core methodology, and the teacher-oracle is used only as a likelihood scorer in a manner standard for distillation-based RL methods; any incidental use of LLMs for writing or editing did not affect scientific rigor.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.