PRIVILEGED LIKELIHOOD IS NOT AUTOMATICALLY VALUE: THREE CHECKS FOR TOKEN CREDIT IN ON-POLICY SELF-DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

On-policy self-distillation aims to improve reinforcement learning from verifiable rewards (RLVR) by providing token-level scores derived from privileged information available only during training, such as reference solutions or critic feedback. These scores are treated as estimates of token-level action values, yet they measure how the model's prediction changes with enriched context rather than how the expected outcome changes when a token is changed. We examine three questions: whether the score tracks task success; whether feedback generated from the same rollout causes self-reference, and whether feedback from other rollouts mitigates it; and what behavior the training objective reinforces. On AIME 2025, the implemented two-sided score for `gpt-oss-20b` distinguishes correct from incorrect trajectories at approximately chance level (AUC $= 0.505$); feedback from a different rollout does not consistently improve discrimination; and all five token-score configurations achieve 24.2–33.9% Avg@4, compared with 64.2% for outcome-only GRPO. The highest-entropy token decile accounts for 57–71% of the total absolute token-advantage mass, despite the score being least informative about reasoning quality in this regime. By contrast, Qwen3-8B trained with SDPO on SciKnowEval Biology improves held-out Avg@8 by 26.8–28.0 percentage points, with trajectory-score AUCs of 0.813–0.924. Together, these results suggest that dense credit assignment through distillation can be effective when its likelihood-based scores are validated as proxies for outcome-relevant credit. Without this alignment, the resulting supervision may fail to generalize and substantially underperform outcome-based RL.

1 INTRODUCTION

Reinforcement learning from verifiable rewards (RLVR) has become a widely used approach for training large language models (LLMs) to reason (Shao et al., 2024). In this setting, an exact-answer verifier evaluates only the model's final output and returns a single scalar reward for the entire reasoning trajectory. Policy-gradient methods such as Group Relative Policy Optimization (GRPO) then broadcast the resulting trajectory-level advantage to every response token. Although each token contributes differently to the gradient through its log-probability, the verifier itself provides no indication of which reasoning steps were responsible for success or failure. Consequently, a decisive algebraic insight and a routine formatting token receive the same outcome-level supervision.

On-policy distillation (OPD) provides a natural mechanism for obtaining denser supervision than sequence-level rewards alone (Agarwal et al., 2024; Gu et al., 2024). A snapshot π_{old} of the student policy π_θ first generates a rollout $y \sim \pi_{\text{old}}(\cdot \mid x)$. A teacher then re-evaluates the student's trajectory and produces token-level probability distributions, which we denote by q_{score}. These distributions provide fine-grained supervision that complements the sparse outcome reward. On-policy *self*-distillation (OPSD) (Zhao et al., 2026a; Hübotter et al., 2026; Yang et al., 2026; Pan et al., 2026; Xia et al., 2026) is a special case of this framework in which the student and teacher are the same model: q_{score} can be obtained from a frozen copy of π_{old}. Crucially, in OPSD, the teacher is conditioned on *privileged information* c, such as a reference solution, environment feedback, or a judge-generated critique, that is available only during training. The resulting token-level signal, which we term the

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

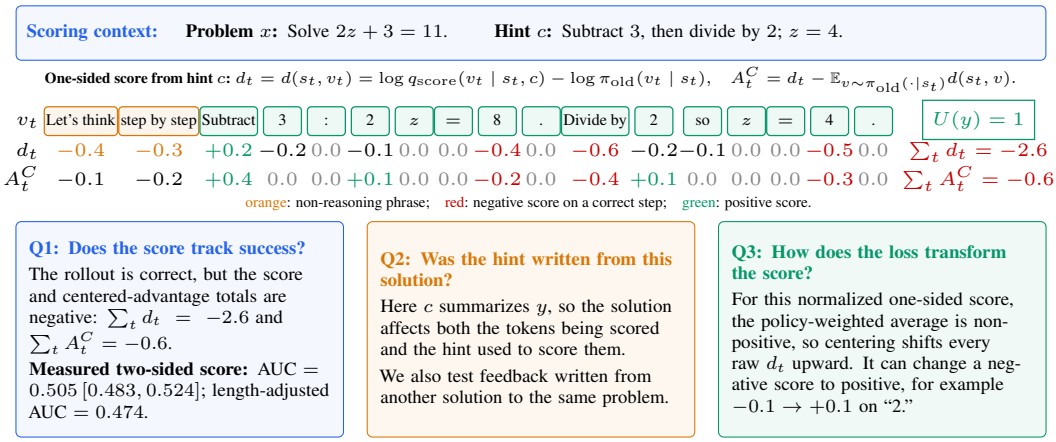


Figure 1: **A correct rollout can receive a negative one-sided score.** The token values shown are illustrative, while Q1 AUC values are measured on AIME 2025. Although the hint c specifies the correct steps, the conditioned model assigns a lower total likelihood to the observed correct rollout than does the rollout policy. **Q1** evaluates the score against task success. **Q2** tests whether the hint was written after observing the same solution. **Q3** tracks how the policy-weighted baseline shifts the score prior to training.

privileged-likelihood score, is defined as

$$d(s_t, v) = \log q_{\text{score}}(v \mid s_t, c) - \log \pi_{\text{old}}(v \mid s_t), \tag{1}$$

where $s_t = (x, y_{<t})$ denotes the reasoning state. The score compares the privileged scorer's probability for candidate token v with that of the rollout policy. When q_{score} is a copy of π_{old}, this comparison isolates the change induced by c. RLCSD (Pan et al., 2026) instead compares a helpful context c^+ with a negative context c^-, yielding the two-sided contrast

$$d_\pm(s_t, v) = \log q_{\text{score}}(v \mid s_t, c^+) - \log q_{\text{score}}(v \mid s_t, c^-). \tag{2}$$

In both cases, the privileged-likelihood score is implicitly treated as a proxy for the token-level action value $Q^\pi(s_t, v)$: the expected terminal utility obtained by choosing token v at state s_t and then continuing generation under policy π. If this approximation is accurate, incorporating privileged-likelihood scores into a sequence-level RL objective can reduce gradient variance by concentrating learning on outcome-critical tokens. If the approximation is poor, however, the resulting supervision introduces bias, encouraging the model to optimize for changes in likelihood that do not correspond to genuine contributions to task success.

Privileged-likelihood scores and action values capture fundamentally different signals. A privileged-likelihood score measures how the model's preference for a token changes when conditioned on additional privileged information, whereas an action value estimates the expected task outcome after selecting that token and continuing the generation. Their agreement is therefore not guaranteed and must be established empirically. Recent studies indeed show that OPSD can fail in several ways: the teacher may favor adherence to a reference trajectory rather than task correctness, remain insensitive to whether the privileged solution itself is correct, or inadvertently suppress useful exploration (Harne et al., 2026; Ichihara et al., 2026; Kaur et al., 2026; Zhu et al., 2026).

We examine three questions before treating the privileged-likelihood score as token credit. **Q1: Does the score track task success?** A likelihood change need not favor a correct solution. Figure 1 shows a correct rollout with utility $U(y) = 1$ and negative totals for both d_t and A_t^C. **Q2: Was the feedback written from the rollout it scores?** If so, the resulting score may reflect how well the rollout matches feedback written specifically about it, and not whether those tokens contribute to a correct answer. **Q3: What does the training loss do with the score?** Baseline subtraction, clipping, and token weighting determine which scores become parameter updates.

We examine these questions primarily with `gpt-oss-20b` trained on DeepScaleR (Luo et al., 2025) and evaluated on AIME 2025 (Mathematical Association of America, 2025). An external LLM judge

generates a helpful explanation c^+ and a plausible incorrect explanation c^- for each rollout. For Q1, we also report Qwen3-8B trained with SDPO (Hübotter et al., 2026) on SciKnowEval Biology (Feng et al., 2024), where the privileged information is the gold answer. Our findings are:

Q1: The score does not track task outcomes on AIME. On AIME, the two-sided score distinguishes correct from incorrect trajectories only at chance level (pooled AUC $= 0.505$, 95% CI $[0.483, 0.524]$); after adjusting for response length, the AUC is 0.474. The one-sided score also remains negative for both correct and incorrect rollouts, with only a small relative difference of $+0.017$. When the score does track outcomes, the same check predicts a different training result. On SciKnowEval Biology, Qwen3-8B trained with SDPO (Hübotter et al., 2026) using the gold answer as privileged information achieves trajectory-level AUC 0.813–0.924 and improves held-out Avg@8 by 26.8–28.0 percentage points. Outcome separation is therefore a practical first check of the score.

Q2: Feedback from another rollout does not recover a useful score. When feedback is written after reading the rollout being scored, the score may reflect textual agreement with that particular feedback. We therefore score each rollout with feedback written from a different rollout of the same problem, a procedure called *cross-fitting*. The resulting AUCs remain near chance or change direction across comparisons on AIME, so changing which rollout supplies the feedback does not make the score reliably track correctness.

Q3: Different token-level scoring configurations consistently degrade training. We evaluate three additive configurations and two KL-regularized configurations against outcome-only GRPO. The outcome-only baseline achieves 64.2% Avg@4, whereas the token-score configurations achieve 24.2%–33.9%, a decrease of 30.4–40.0 percentage points. A separate analysis finds that 57–71% of the sum of absolute token advantages falls in the highest-entropy token decile, indicating that the learning objective places most token-level weight at positions where the model is most uncertain.

Together, our analyses show that the score, the feedback used to compute it, and the training loss must be checked separately before a likelihood difference should be treated as token credit.

2 Background and Related Work

2.1 The OPSD recipe and its variants

The one-sided score $d(s_t, v)$ and two-sided score $d_\pm(s_t, v)$, defined in Equations 1 and 2, supply token-level signals; the training loss determines how they update the model. Existing methods use them in three ways.

As a distillation target. The simplest use treats the teacher distribution $q_{\text{score}}(\cdot \mid s_t, c)$ as a soft label and minimizes its divergence from the student $\pi_\theta(\cdot \mid s_t)$. OPSD (Zhao et al., 2026a) and SDPO (Hübotter et al., 2026) apply KL distillation across the rollout, while CriPO (Xia et al., 2026) restricts it to rubric-selected tokens. VPD (Li et al., 2026a) co-evolves the teacher from language feedback before distillation. These losses match the teacher without task reward. The frozen-target projection is derived in Theorem 7.

As an additive reward. A second family combines the token score with a verifier reward. RLSD (Yang et al., 2026) uses the score to weight token's utility advantage. RLCSD (Pan et al., 2026) compares helpful and incorrect contexts while retaining the verifier reward. OPPO (Li et al., 2026c) derives a Bayesian recursion that accumulates the log-ratio along a trajectory to estimate running success probability. SC-GRPO (Shan et al., 2026) uses per-token KL from a self-conditioned teacher as a multiplicative weight on the utility advantage. These methods are *reward-anchored*: the score modulates an objective with verifier feedback, and zeroing it out recovers outcome-only training.

As a KL regularizer. A third group uses the privileged teacher as a KL target and adds a penalty $\beta D_{\text{KL}}(\pi_\theta(\cdot \mid s_t) \| q_{\text{score}}(\cdot \mid s_t, c))$ to the objective. This resembles KL-regularized control (Geist et al., 2019; Rafailov et al., 2023). VPD (Li et al., 2026a) derives its student reverse-KL update from this optimum, an exponential tilt of a reference policy by a soft advantage (Theorem 10). This equivalence requires a fixed target and the KL penalty as the only shaping term; on-policy self-distillation may refresh q_{score} as π_θ changes across updates.

A noisy score may harm a standalone distillation target but help as a small addition to an outcome reward, while a systematically incorrect score can harm either objective. Thus, the training loss (Q3) must be diagnosed separately from the score (Q1).

2.2 CONCURRENT EMPIRICAL LANDSCAPE

We summarize the concurrent findings most relevant to our analysis.

Failure reports. Harne et al. (2026) study OPSD and find that the teacher follows the reference trajectory more closely than correctness, with much supervision falling on inconsequential tokens such as stopwords and punctuation and a per-token loss indistinguishable between correct and incorrect rollouts. Ichihara et al. (2026) find that another problem's worked solution, even when deliberately corrupted, can retain much of the OPSD gain, showing that the problem's own solution is not always necessary and that the transferred signal is not target-specific correctness. Kaur et al. (2026) report that privileged teachers can suppress exploratory tokens at uncertain reasoning steps, while Zhu et al. (2026) report better results with shared rules than with sample-specific solutions.

Success reports. A parallel literature reports gains when the score is combined with a verifier anchor or the privileged information is constructed differently. OPPO (Li et al., 2026c) derives a Bayesian credit recursion anchored to the GRPO advantage and reports improvements on Math. Purified OPSD (Shen et al., 2026) decomposes the teacher update by question presence, subtracting a reference-only teacher (no question) from the full teacher (question plus reference); this decomposition is orthogonal to our correctness-polarity contrast. SC-GRPO (Shan et al., 2026) uses per-token self-conditioned KL to weight a verifier-anchored objective, while CAST (Li et al., 2026b) uses answer-free self-teacher gaps to shape GRPO advantages. VPD (Li et al., 2026a) co-evolves a feedback-conditioned teacher and reports gains over RLVR and frozen-teacher distillation on scientific reasoning and code generation. PMD (Liu et al., 2026b) conditions the teacher on online procedural memory and improves over SDPO. Several other methods, including position-weighted scoring (Liu et al., 2026a), local error correction (Zhao et al., 2026b), and sibling-conditioned distillation (Tan & Hong, 2026), modify where or how the score is applied while retaining the basic privileged-likelihood construction. β-OPSD (Xu et al., 2026) uses a return-to-go estimator to address per-token KL myopia and recover an unbiased sequence-level gradient.

The emerging pattern. Overall, these results show that OPSD depends on both the information given to the teacher and how its probabilities enter the loss. Our experiments complement this work by testing whether the score separates correct and incorrect outcomes, comparing same-rollout feedback with cross-fitting, and comparing token-level advantage configurations with outcome-only training. Our theoretical analyses connect these conditions to sequence-level outcome success.

3 FRAMEWORK

Before treating a privileged-likelihood score as token credit, we ask three questions:

Q1. *Does the score track task success?* At a given reasoning state, does the score prefer next tokens that lead to better outcomes?

Q2. *Was the feedback written from the rollout being scored?* If so, the score may partly measure agreement with a description written specifically for that rollout. Does the result persist when each rollout is scored with feedback from another rollout, when self-dependence is removed?

Q3. *What does the training loss do with the score?* Baseline subtraction, clipping, and token weighting determine how the score changes the update. Does the resulting gradient point in a helpful direction?

These are separate questions: a score can fail Q1 while Q2 is irrelevant, or pass Q1 while the loss (Q3) destroys the signal. This section establishes why each separation is formally necessary.

3.1 Q1: WHEN DOES A LIKELIHOOD RATIO RANK ACTION VALUE?

At a reasoning state s_t, let $d(s_t, v)$ denote the likelihood score being evaluated and let $Q^\pi(s_t, v)$ be the expected terminal utility (i.e. outcome reward) after choosing v and continuing with policy π. For the two quantities to value every local redistribution of next-token probability in the same relative scale, the score must be a positive affine transformation of the action value. Otherwise, a score that preserves the ranking but distorts the magnitudes will over-concentrate mass on the top-ranked token. For example, action values $Q = (0, 1, 2)$ and scores $d = (0, 1, 4)$ rank three tokens identically but give different relative weight to probability shifts.

Proposition 1 (Exact local affine equivalence)**.** *At an interior policy, a score $d(s,\cdot)$ assigns a proportional first-order change to $Q^\pi(s,\cdot)$, with one positive scale shared across every zero-sum perturbation of the next-token distribution, if and only if there exist a state-dependent offset $\omega_0(s) \in \mathbb{R}$ and scale $\omega_1(s) > 0$, both independent of v, such that*

$$d(s,v) = \omega_0(s) + \omega_1(s)\, Q^\pi(s,v) \quad \textit{for every } v. \tag{3}$$

This is a standard expected-utility result (Herstein & Milnor, 1953; Skalse et al., 2023); the proof appears in §A.2.1. The reason privileged-likelihood scores are plausible as credit is that a log-ratio of this form does rank action value in one standard case. Hindsight credit assignment (Harutyunyan et al., 2019; Meulemans et al., 2023) uses the true success and failure conditionals: for each candidate token v, let $q_1(v \mid s)$ be how likely v is given that the trajectory will succeed, and $q_0(v \mid s)$ given that it will fail. Bayes' rule then gives

$$\log \frac{q_1(v \mid s)}{q_0(v \mid s)} \;=\; \operatorname{logit} P(U{=}1 \mid s, v) \;-\; \operatorname{logit} P(U{=}1 \mid s), \tag{4}$$

where $P(U = 1 \mid s, v) = Q^\pi(s,v)$, under the binary reward assumptions in §A.2.2. Because the logit is increasing, this log-ratio ranks binary action value. Tokens that lead to success get higher scores. This identity explains why privileged-likelihood scores are appealing. The two-sided score $d_\pm(s,v) = \log q_{\text{score}}(v \mid s, c^+) - \log q_{\text{score}}(v \mid s, c^-)$ has the same log-ratio form. The ranking guarantee applies only if the two distributions q_1 and q_0, expressed via context c^+ and c^-, are coherent success and failure conditionals from one joint distribution. However, free-form texts labeled "helpful" and "unhelpful" need not satisfy that condition, and may also differ in formatting, length, or which values they reveal.

This is the gap that Q1 targets. The form of the score guarantees nothing; agreement with action value is an empirical question that must be checked. The most accessible test is at the sequence level: average the token scores over a trajectory and ask whether the resulting scalar separates correct from incorrect rollouts (AUC). This is a necessary but not sufficient condition—if the mean token score cannot even tell correct trajectories from incorrect ones, the individual token scores cannot be ranking action values correctly, since any correct ranking must aggregate to a correct trajectory-level ordering.

3.2 Q2: Self-dependence and what cross-fitting does and does not fix

Q2 concerns the relationship between the privileged context and the rollout being scored. Privileged information comes in many forms, and the forms differ in a structural way that matters for the score's validity. A reference solution or a fixed worked example is determined by the problem alone: it exists before the student generates anything. A judge-written critique or environment feedback, by contrast, is typically generated *after reading* the student's rollout. In the first case the scoring context is independent of the rollout; in the second, it is not.

When the context depends on the rollout, the score picks up an additional source of variation: textual agreement between the rollout and its own description. A critique that says "do not divide by 4" is informative because it describes the error the student made, but this means the score depends on y through both the tokens being scored and the context $c(y)$. The score can be high because the rollout and its feedback are about the same thing, regardless of whether the rollout is correct.

Cross-fitting removes this loop by scoring each rollout with feedback written from a *different* rollout of the same problem. The target rollout no longer influences its own scoring context. This is a clean test of whether the score's apparent signal survives when c is not written from the target.

Proposition 2 (Target-side independence under cross-fitting)**.** *Let c be feedback constructed only from a donor rollout y^{don} and external randomness, and let y^{tar} be an independently selected target rollout for the same prompt x and fold assignment f. Then $c \perp y^{\text{tar}} \mid x, f$. This independence does not imply that the resulting score tracks task utility.*

Cross-fitting therefore defines a held-out-feedback score rather than a corrected version of same-rollout scoring. It answers how the score changes when the target does not supply its own feedback. Removing the dependence of c on the target rollout changes the scoring rule; it does not recover a debiased version of same-rollout scoring. Q1 still determines whether the new score tracks outcomes. The proof and the corresponding partial-gradient analyses are elaborated in §A.4.

3.3 Q3: How does the loss turn a score into an update?

Q1 asks what information the score carries. Q3 follows what happens after that score enters training:

$$d_{\pm}(s,v) \;\longrightarrow\; A^C(s,v) \;\longrightarrow\; A_{i,t} \;\longrightarrow\; \nabla_{\mathbf{z}}\mathcal{L} \;\longrightarrow\; \nabla_\theta \mathcal{L}.$$

Here A^C is the centered token advantage, $A_{i,t}$ is the final token advantage after combining the score with utility advantage A_i^U (if any) and applying any gates or clipping, $\mathcal{L}$ is the actor loss, $\mathbf{z}$ denotes the token-logit vector, and θ denotes the shared model parameters. The following steps distinguish changes to a sampled advantage, the local response of a loss, and the resulting parameter gradient.

From scores to token advantages. Given a sampled token $v \sim \pi_{\text{old}}(\cdot \mid s)$, the policy-weighted baseline gives

$$A^C(s,v) = d_{\pm}(s,v) - \sum_{v'} \pi_{\text{old}}(v' \mid s) d_{\pm}(s,v'). \tag{5}$$

For a two-sided score, the policy-weighted mean can have either sign, so centering can turn a positive score into a negative advantage. Thus, the loss acts on the centered signal rather than the raw score.

Losses respond differently at confident predictions. The same score can enter training as a bounded additive term or through KL regularization. For fixed $d_{\pm}(s,\cdot)$, reference policy π_{ref}, and positive α, β, maximizing $\alpha \mathbb{E}_{v\sim\pi_\theta(\cdot|s)} d_{\pm}(s,v) - \beta D_{\text{KL}}(\pi_\theta(\cdot \mid s) \| \pi_{\text{ref}}(\cdot \mid s))$ is equivalent to minimizing reverse KL toward the score-tilted target

$$\pi^\star(v \mid s) \;\propto\; \pi_{\text{ref}}(v \mid s) \exp\left(\frac{\alpha}{\beta} d_{\pm}(s,v)\right).$$

This connection permits a common local comparison of additive and KL objectives.

Proposition 3 (Local response near a confident prediction)**.** *Restrict the next-token distribution to two tokens, let v be the more likely token, and write $\varepsilon = 1 - \pi_\theta(v \mid s)$ for the probability of the other token. For a bounded fixed score $d_{\pm}(s,\cdot)$, the logit-gradient magnitude of the additive loss $-\mathbb{E}_{v'\sim\pi_\theta(\cdot|s)} d_{\pm}(s,v')$ is at most a constant times ε. For reverse KL $D_{\text{KL}}(\pi_\theta(\cdot \mid s) \| \pi^\star(\cdot \mid s))$ toward a fixed full-support target $\pi^\star$, the magnitude is proportional to $\varepsilon \log(1/\varepsilon)$ near $\varepsilon = 0$. Both gradients approach zero as the model becomes certain.*

Reverse KL therefore responds more strongly than a bounded additive score near this boundary, but both responses weaken as the policy becomes certain. Their gradients move toward the supplied score or target, which need not favor higher task utility. Full derivations, comparisons with other losses, and saved-state measurements appear in §A.5 and Figure 3.

From token advantages to parameter gradients. Let T_i be the response length, let $A_{i,t}$ be the detached advantage assigned to token t, and let $h_{i,t} = \nabla_\theta \log \pi_\theta(v_{i,t} \mid s_{i,t})$. Let g_A be the expected ascent update using $A_{i,t}$, and g_U for the outcome-only ascent update that uses A_i^U at every token; the corresponding actor-loss gradients have the opposite sign. Before policy-ratio clipping,

$$g_A - g_U = \mathbb{E}\left[\frac{1}{T_i} \sum_t \left(A_{i,t} - A_i^U\right) h_{i,t}\right]. \tag{6}$$

The advantage values alone do not determine this vector because the token score-function vectors $h_{i,t}$ generally point in different parameter-space directions.

Proposition 4 (Advantage totals do not determine the parameter gradient)**.** *Even if $A_{i,t} = A_i^U w_{i,t}$ with positive, mean-one weights $w_{i,t}$, preserving the advantage sign and its total across tokens does not generally preserve the expected policy-gradient estimate.*

For two independent token coordinates, changing the weights from $(1,1)$ to $(2-\varepsilon, \varepsilon)$ changes the expected update from $(1/8, 1/8)$ to $((2-\varepsilon)/8, \varepsilon/8)$, though the weights remain positive and mean one. More details are in §A.6. Token weighting can be useful because it changes the update; the proposition says that the advantage sign and total across tokens alone cannot certify that change.

4 EXPERIMENTAL SETUP

We study how privileged-likelihood scores behave in two settings. The main AIME experiments analyze score relationships, feedback construction, and a SciKnowEval study as a counterfactual.

Model and benchmark. The AIME study trains `gpt-oss-20b` with effort `low` on DeepScaleR (Luo et al., 2025) and evaluates on AIME 2025. For each problem, the model samples $K > 1$ solutions, and an equivalent-answer verifier assigns binary outcome utility $U_i \in \{0, 1\}$.

Feedback construction. For the AIME experiments, an external LLM judge (gpt-5.6-luna) receives the problem, reference solution, verifier result, and full reasoning trace. It produces helpful feedback c_i^+ and plausible incorrect or harmful feedback c_i^-. To prevent the student from merely copying the feedback, the helpful template masks selected numerical values and often includes formatting guidance, while the incorrect template supplies explicit intentionally plausible-yet-incorrect values. The two texts are intended to differ only in the mathematical claim being tested, so that their likelihood contrast isolates mathematical content. The resulting likelihood contrast can therefore respond to wording and formatting as well as mathematical content. More details are reported in §A.10.2.

Scoring. The rollout policy π_{old} generates the sampled trajectories, and a teacher distribution q_{score} supplies privileged probabilities. We use the one-sided score d from Equation (1) for the teacher–policy comparison in §5.1 and the two-sided score $d_{\pm}$ from Equation (2), centered as in Equation (5), for training. We denote this centered score by $A^C_{i,t}$, set $B(z) = 0.5 \tanh(z/0.5)$, and use the entropy gate $g_{H,i,t} = \min(1, H_{i,t}/0.2)$. The scorer checkpoint used is identified in §A.10.2.

Training configurations. The AIME configurations share the initialization, training data, optimizer, rollout batches, and verifier. The outcome-only configuration uses the leave-one-out utility advantage $A^U_i = U_i - \frac{1}{K-1}\sum_{j \neq i} U_j$. We compare it with three additive variants that apply entropy weighting, source clipping, or both, and two KL-regularized variants that use full or projected reference-policy corrections. In all five, the token-level term is disabled when $A^U_i = 0$. Table 3 gives the core advantages; exact definitions, execution order, and hyperparameters appear in §A.10.4.

Cross-fitting protocol. For Q2, each rollout is first paired with feedback written from that rollout. We then assign the rollout–feedback pairs within each prompt to folds using a content-independent permutation. Each target rollout is scored using feedback paired with a donor rollout from the opposite fold, so the target does not supply its own scoring context (§3.2). The resulting AUCs test whether the score's apparent signal survives removal of direct target-to-feedback dependence.

SciKnowEval Biology comparison. We also train Qwen3-8B (both thinking ON and OFF) with SDPO on SciKnowEval Biology questions. The teacher's only privileged information is the gold answer, written as "The correct answer is [gold answer]."

Evaluation. For AIME, we report Avg@4 at the latest checkpoint shared by all six configurations, together with paired prompt-bootstrap confidence intervals and exact sign-flip p-values. For SciKnowEval Biology, we report held-out Avg@8 and trajectory-score AUC. Further scorer, checkpoint-selection, and statistical details are in §A.10.

5 RESULTS

5.1 Q1: THE DENSE-REWARD SCORE DOES NOT RELIABLY DISCRIMINATE

One-sided teacher–policy residual. We first examine one-sided $d(s_{i,t}, v)$, by measuring the mean residual $\frac{1}{T}\sum_{t=1}^{T} d(s_{i,t}, v)$ across 18 reasoning checkpoints trained on DeepScaleR. If this residual provides useful credit, it should be positive for tokens from correct rollouts (the teacher supports the student's choices) and negative for tokens from incorrect ones (the teacher suppresses them). The empirical mean residual, however, is negative for both correct and incorrect rollouts (-0.0334 and -0.0502). The teacher down-scores the student's own sampled tokens regardless of correctness. The

gap between correct and incorrect ($+0.017$) is positive, indicating that the privileged context retains some relative outcome information, but the absolute signal points in the wrong direction for both.

Two-sided trajectory AUC. We test the two-sided score $d_{\pm}$ used in training. For each trajectory, we average the bounded, centered token advantages over response tokens and measure whether the result ranks a correct trajectory above an incorrect one. We measure that with AUC, which is the probability that a randomly drawn correct trajectory scores above a randomly drawn incorrect one (0.5 is chance). Pooled AUC compares all rollouts, length-adjusted AUC removes the score's linear association with response length, and within-prompt AUC compares solutions to the same problem.

Table 1: **Correctness discrimination on AIME 2025.** Values are AUCs with 95% prompt-bootstrap confidence intervals. Each row uses its predefined evaluation window.

Score construction	Pooled AUC	Length-adjusted AUC	Within-prompt AUC
Implemented two-sided score	0.505 [0.483, 0.524]	0.474 [0.452, 0.494]	0.499 [0.469, 0.533]
Feedback from the same rollout	0.588 [0.500, 0.673]	0.551 [0.463, 0.625]	0.609 [0.542, 0.670]
Other-rollout masked feedback	0.512 [0.440, 0.579]	0.485 [0.423, 0.548]	0.516 [0.450, 0.574]
Other-rollout correct-fact feedback	0.416 [0.347, 0.484]	0.446 [0.382, 0.513]	0.587 [0.525, 0.646]

As shown in Table 1, the implemented score is indistinguishable from chance in the pooled comparison (AUC $= 0.505$) and slightly favors incorrect trajectories after length adjustment (AUC $= 0.474$). This answers Q1 negatively for the AIME experiment: the implemented training signal does not reliably rank correct trajectories above incorrect ones. Because any score that provides accurate token-level credit must at minimum separate trajectories in aggregate, this sequence-level failure already implies failure at the token level. The near-chance AUC does not mean the available information is useless. An external judge given the same inputs selected the better solution at $\sim 94\%$ accuracy, yet the base model's token-likelihood score does not extract that information.

Positive comparison on SciKnowEval Biology. The same diagnostic behaves differently when Qwen3-8B is trained by SDPO with the gold answer as privileged information, which improves held-out Avg@8, and their raw and centered trajectory scores remain highly above chance, as seen in Table 2. Centering reduces discrimination in both configurations but does not remove it. Across these two systems, stronger outcome separation accompanies successful self-distillation training, supporting trajectory AUC as a practical Q1 screen.

Table 2: **One-sided SDPO with gold-answer context on SciKnowEval Biology.** Avg@8 accuracy on 50 held-out questions. Within-prompt AUC uses the raw trajectory-mean score.

Configuration	Avg@8, step 0 $\rightarrow$ 100	Gain (pp)	Raw AUC	Centered AUC	Within-prompt AUC
Thinking disabled	31.50 $\rightarrow$ 58.25	+26.75	0.924	0.751	0.833
Thinking enabled	26.25 $\rightarrow$ 54.25	+28.00	0.813	0.626	0.709

5.2 Q2: Feedback from another rollout does not recover a reliable score

The same-rollout score in Table 1 reaches pooled AUC 0.588, but each solution helped determine the feedback used to score it. Cross-fitting instead uses feedback written from another rollout of the same problem. The two other-rollout scores have pooled AUCs 0.512 and 0.416, and the latter changes to 0.587 within prompt. Changing the feedback source therefore changes what the score measures, but neither held-out construction gives a stable relationship with correctness across the reported comparisons. This answers Q2: removing self-dependence is mechanically sound but does not rescue the score. This result is consistent with Ichihara et al. (2026), who find that replacing the target's reference solution with a different problem's solution produces comparable training outcomes. Thus, whatever the privileged context transfers, it is not target-specific credit.

5.3 Q3: All tested token-score configurations underperform

Table 3: **Paired AIME 2025 comparison.** Accuracy is Avg@4. Differences are computed before rounding as configuration minus outcome-only GRPO, with 95% paired prompt-bootstrap confidence intervals; negative values favor outcome-only GRPO. Full details are in §A.10.4.

Configuration	Core advantage	Acc. (Avg@4)	Difference [95% CI] (pp)
Outcome-only GRPO	A_i^U	64.2	–
Entropy-gated additive	$A_i^U + g_{H,i,t}B(A_{i,t}^C)$	33.9	-30.4 $[-42.9, -17.9]$
Source-clipped additive	$A_i^U + B(A_{i,t}^{C,\text{clip}})$	27.5	-36.7 $[-49.2, -24.2]$
Entropy-gated + source-clipped	$A_i^U + g_{H,i,t}B(A_{i,t}^{C,\text{clip}})$	30.0	-34.2 $[-45.8, -23.3]$
Full KL-regularized	$A_i^U + A_{i,t}^{C,\text{full}}$	24.2	-40.0 $[-52.5, -27.5]$
Projected KL correction	$A_i^U + A_{i,t}^{C,\text{proj}}$	25.0	-39.2 $[-50.8, -27.5]$

We compare outcome-only GRPO and five token-level dense-reward configurations on the same AIME 2025 evaluation set at the latest shared checkpoint. All five dense-reward configurations underperform outcome-only GRPO. Outcome-only GRPO reaches 64.2% accuracy, while the tested dense variants range from 24.2% to 33.9%, corresponding to decreases of 30.4 to 40.0 percentage points. Every confidence interval is entirely below zero, as seen in Table 3.

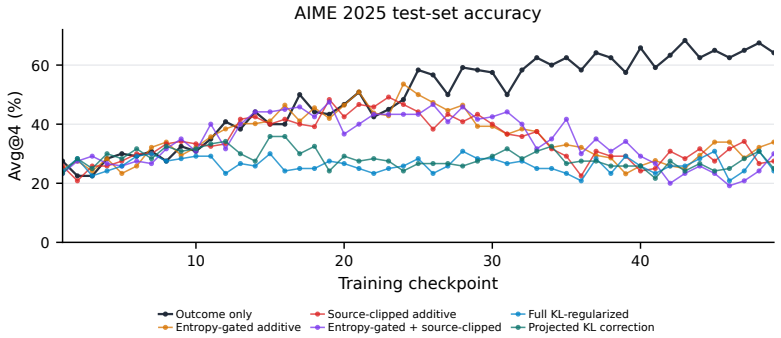


Figure 2: **AIME 2025 accuracy over training.** Test Avg@4 begins in a similar range across configurations. Outcome-only GRPO continues to improve, while all five token-score configurations finish below it. The fixed comparison in Table 3 uses update 49.

Figure 2 shows how the performance gap develops during training. Outcome-only GRPO improves and stabilizes, while the token-level dense-reward configurations plateau or decline. The dense-reward configurations also produce boxed, complete answers more often than outcome-only GRPO, so their lower accuracy is not explained by a greater failure to return final answers.

How the score changes training. The five paired configurations disable the token advantage when $A_i^U = 0$. In a separate ungated additive analysis, 57%–71% of the sum of absolute token advantages comes from the highest-entropy token decile. Across the configurations, the additive variants move toward higher token entropy and larger reference-policy KL, while the KL-regularized variants remain near the outcome-only range but gain no accuracy (§A.10.4). Centering can also change the sign of an individual token advantage, as illustrated in Figure 1.

6 Conclusion

Privileged-likelihood scores should be treated as token credit only after checking outcome alignment, feedback construction, and how the loss uses them. On AIME 2025, our score was near chance at separating correct from incorrect trajectories and all tested dense-reward configurations underperformed outcome-only GRPO, whereas stronger separation accompanied successful SDPO training on SciKnowEval Biology. Dense self-distillation can work, but the score must be validated before use.

THE USE OF LARGE LANGUAGE MODELS

Large language models were used as a writing and coding aid. For the manuscript, they assisted with drafting and editing prose, tightening captions, and formatting references; every claim, number, and citation was checked against the source measurements and the cited papers by the authors. For the implementation, they assisted with routine code editing and test scaffolding; all operators, correctness tests, and measurements were designed, reviewed, and executed by the authors. Large language models were also used to derive and validate mathematical proofs in the paper. They were not used to generate research ideas, experimental designs, or results, and the authors take full responsibility for the content of this paper.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.13649.

Jose A. Arjona-Medina, Michael Gillhofer, Michael Widrich, Thomas Unterthiner, Johannes Brandstetter, and Sepp Hochreiter. RUDDER: Return decomposition for delayed rewards. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1806.07857.

M. D. Donsker and S. R. S. Varadhan. Asymptotic evaluation of certain markov process expectations for large time, I. *Communications on Pure and Applied Mathematics*, 28(1):1–47, 1975. doi: 10.1002/cpa.3160280102. URL https://doi.org/10.1002/cpa.3160280102.

Kehua Feng, Xinyi Shen, Weijie Wang, Xiang Zhuang, Yuqi Tang, Qiang Zhang, and Keyan Ding. SciKnowEval: Evaluating multi-level scientific knowledge of large language models. *arXiv preprint arXiv:2406.09098*, 2024. URL https://arxiv.org/abs/2406.09098.

Matthieu Geist, Bruno Scherrer, and Olivier Pietquin. A theory of regularized markov decision processes. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 2160–2169. PMLR, 2019. URL https://proceedings.mlr.press/v97/geist19a.html.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. MiniLLM: On-policy distillation of large language models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.08543.

Sarthak Harne, Chinmay Karkar, Yash Pandya, Ahmed Awadallah, and Akshay Nambi. Privileged, but biased: How PI-conditioned teachers break self-distillation. *arXiv preprint arXiv:2608.04794*, 2026. URL https://arxiv.org/abs/2608.04794.

Anna Harutyunyan, Will Dabney, Thomas Mesnard, Mohammad Azar, Bilal Piot, Nicolas Heess, Hado van Hasselt, Greg Wayne, Satinder Singh, Doina Precup, and Rémi Munos. Hindsight credit assignment. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1912.02503.

I. N. Herstein and John Milnor. An axiomatic approach to measurable utility. *Econometrica*, 21(2): 291–297, 1953. doi: 10.2307/1905540.

Jonas Hübotter, Frederike Lübeck, Lejs Behric, Anton Baumann, Marco Bagatella, Daniel Marta, Ido Hakimi, Idan Shenfeld, Thomas Kleine Buening, Carlos Guestrin, and Andreas Krause. Reinforcement learning via self-distillation. *arXiv preprint arXiv:2601.20802*, 2026. URL https://arxiv.org/abs/2601.20802.

Yuki Ichihara, Naoto Iwase, Mohammad Atif Quamar, and Junpei Komiyama. Privileged solutions or context-induced teacher behavior? dissecting on-policy self-distillation. *arXiv preprint arXiv:2608.09228*, 2026. URL https://arxiv.org/abs/2608.09228.

Simran Kaur, Narutatsu Ri, Yinghui He, Liam Fowl, and Sanjeev Arora. Rethinking on-policy self-distillation for thinking models. *arXiv preprint arXiv:2607.05184*, 2026. URL https://arxiv.org/abs/2607.05184.

Yang Li, Erik Nijkamp, Semih Yavuz, and Shafiq Joty. Learning from language feedback via variational policy distillation. *arXiv preprint arXiv:2605.15113*, 2026a. URL https://arxiv.org/abs/2605.15113.

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

Yu Li, Rui Miao, Tian Lan, and Zhengling Qi. OPPO: Bayesian value recursion for token-level credit assignment in LLM reasoning. *arXiv preprint arXiv:2605.21851*, 2026c. URL https://arxiv.org/abs/2605.21851.

Xiaogeng Liu, Xinyan Wang, Yingzi Ma, Yechao Zhang, and Chaowei Xiao. When are teacher tokens reliable? position-weighted on-policy self-distillation for reasoning. *arXiv preprint arXiv:2605.21606*, 2026a. URL https://arxiv.org/abs/2605.21606.

Ye Liu, Srijan Bansal, Bo Pang, Yang Li, Zeyu Leo Liu, Yifei Ming, Zixuan Ke, Shafiq Joty, and Semih Yavuz. Procedural memory distillation: Online reflection for self-improving language models. *arXiv preprint arXiv:2607.01480*, 2026b. URL https://arxiv.org/abs/2607.01480.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. DeepScaleR: Surpassing O1-Preview with a 1.5b model by scaling RL. Project page, 2025. Notion Blog.

Mathematical Association of America. American invitational mathematics examination (AIME), 2025. URL https://www.maa.org/maa-invitational-competitions.

Alexander Meulemans, Simon Schug, Seijin Kobayashi, Nathaniel Daw, and Gregory Wayne. Would i have gotten that reward? long-term credit assignment by counterfactual contribution analysis. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2306.16803.

Leyi Pan, Shuchang Tao, Yunpeng Zhai, Lingzhe Zhang, Zhaoyang Liu, Bolin Ding, Aiwei Liu, and Lijie Wen. RLCSD: Reinforcement learning with contrastive on-policy self-distillation. *arXiv preprint arXiv:2606.11709*, 2026. URL https://arxiv.org/abs/2606.11709.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 53728–53741, 2023. URL https://arxiv.org/abs/2305.18290.

John Schulman, Nicolas Heess, Theophane Weber, and Pieter Abbeel. Gradient estimation using stochastic computation graphs. In *Advances in Neural Information Processing Systems*, volume 28, 2015. URL https://arxiv.org/abs/1506.05254.

Yingyu Shan, Yuhang Guo, Zihao Cheng, Zeming Liu, Xiangrong Zhu, Xinyi Wang, Jiashu Yao, Wei Lin, Hongru Wang, and Heyan Huang. Learning from own solutions: Self-conditioned credit assignment for reinforcement learning with verifiable rewards. *arXiv preprint arXiv:2606.18810*, 2026. URL https://arxiv.org/abs/2606.18810.

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

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

Joar Skalse, Matthew Farrugia-Roberts, Stuart Russell, Alessandro Abate, and Adam Gleave. Invariance in policy optimisation and partial identifiability in reward learning. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, 2023. URL https://arxiv.org/abs/2203.07475.

Zhiquan Tan and Yinrong Hong. Self-supervised on-policy distillation for reasoning language models. *arXiv preprint arXiv:2605.17497*, 2026. URL https://arxiv.org/abs/2605.17497.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992. doi: 10.1007/BF00992696. URL https://doi.org/10.1007/BF00992696.

Mingxuan Xia, Yuhang Yang, Chao Ye, Shuai Zhu, Shenzhi Yang, Guangcheng Zhu, Yuhang Zhang, Cheng Peng, Haobo Wang, and Siqing Wang. CriPO: Enhancing rubric-based RL via self-distillation. *arXiv preprint arXiv:2607.18082*, 2026. URL https://arxiv.org/abs/2607.18082.

Jiawei Xu, Minghui Liu, Juzheng Zhang, Tom Goldstein, and Furong Huang. β-OPSD: Deriving with policy optimization, training with self-distillation. *arXiv preprint arXiv:2607.28582*, 2026. URL https://arxiv.org/abs/2607.28582.

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

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

Ziqi Zhao, Xinyu Ma, Liu Yang, Yujie Feng, Daiting Shi, Jingzhou He, Xin Xin, Zhaochun Ren, and Xiao-Ming Wu. ROSD: Reflective on-policy self-distillation for language model reasoning across domains. *arXiv preprint arXiv:2605.28014*, 2026b. URL https://arxiv.org/abs/2605.28014.

Siqi Zhu, Xuyan Ye, Hongyu Lu, Weiye Shi, and Ge Liu. The many faces of on-policy distillation: Pitfalls, mechanisms, and fixes. *arXiv preprint arXiv:2605.11182*, 2026. URL https://arxiv.org/abs/2605.11182.

A Supplementary Derivations and Protocol

A.1 Standing assumptions and scope

Unless a result states otherwise, the following assumptions apply.

1. **Actions and support.** The local vocabulary $\mathcal{V}$ is finite. Each result states the support it needs: forward KL requires $q_{\text{score}}(\cdot \mid s, c) \ll \pi_\theta(\cdot \mid s)$; likelihood ratios require positive common support for their numerator and denominator; reverse-KL centroid and Gibbs statements use full support where the optimized policy may place mass. Termination is an action and is included when a token identity is lifted to a trajectory identity.
2. **Utility.** $U(y)$ is integrable. The action value $Q^\pi(s, v)$ intervenes on $v_t = v$ and then follows the declared continuation policy π; it is not a teacher likelihood or an observational conditional unless the additional causal assumptions in §A.2.2 hold.
3. **Local objectives.** Frozen-target calculations hold the state/context weighting law, target $\pi^\star$, token score d or $d_\pm$, and reference policy π_{ref} fixed. They do not differentiate through target construction or state occupancy.
4. **Geometry.** Boundary rates use independent Euclidean softmax logits at one state. They are not invariant to a neural parameterization, natural-gradient metric, optimizer preconditioning, or parameter sharing across states.
5. **Detached estimators.** Policy-gradient advantages, baselines, token weights, feedback, and scorer outputs are detached unless an upstream score term is explicitly shown. Expectations and derivatives may be interchanged.
6. **Cross-fitting.** Fold metadata is assigned by a content-independent random permutation after rollout and feedback generation but before donor selection and target scoring. Donor and target randomness are conditionally independent, and the feedback judge's random seed is external to target generation.

A.2 Affine equivalence and coherent outcome conditionals

A.2.1 Proof of exact local affine equivalence

Proof. Suppress s in d and Q^π. We use the full zero-sum affine tangent space, corresponding to unrestricted local perturbations at an interior policy. Positive-affine equivalence says that, for one $\omega_1 > 0$ shared by all such perturbations,

$$\langle \eta, d - \omega_1 Q^\pi \rangle = 0 \quad \text{for every } \eta \text{ such that } \mathbf{1}^\top \eta = 0.$$

The orthogonal complement of the zero-sum subspace in $\mathbb{R}^{|\mathcal{V}|}$ is $\operatorname{span}\{\mathbf{1}\}$. Hence $d - \omega_1 Q^\pi = \omega_0 \mathbf{1}$, which is (3). Conversely, the constant term vanishes against every zero-sum perturbation, so (3) gives $\langle \eta, d \rangle = \omega_1 \langle \eta, Q^\pi \rangle$. □

Ranking is insufficient. Let three actions have $Q^\pi = (0, 1, 2)$ and $d = (0, 1, 4)$. The rankings are identical, but no positive affine map sends Q^π to d. Indeed, for $\eta = (1, -2, 1)$, $\langle \eta, Q^\pi \rangle = 0$ while $\langle \eta, d \rangle = 2$. AUC, rank correlation, or any strictly increasing nonlinear transformation can therefore pass while exact local functional equivalence fails.

A.2.2 Outcome-conditional likelihood ratios

Proposition 5 (Outcome-conditional odds identity; standard)**.** *Suppose $U \in \{0, 1\}$, both outcome classes have positive probability at state s, and $q_u(v \mid s) = P(v_t = v \mid s_t = s, U = u)$ have common support. Then*

$$\log \frac{q_1(v \mid s)}{q_0(v \mid s)} = \operatorname{logit} P(U = 1 \mid s, v) - \operatorname{logit} P(U = 1 \mid s). \tag{7}$$

If the action at s is randomized according to the declared policy, consistency holds, there is no unmeasured action–outcome confounding conditional on s, positivity holds, and future actions follow π, then $P(U = 1 \mid s, v) = Q^\pi(s, v)$.

Proof. Bayes' rule yields

$$\frac{P(v \mid s, U=1)}{P(v \mid s, U=0)} = \frac{P(U=1 \mid s, v)}{P(U=0 \mid s, v)} \frac{P(U=0 \mid s)}{P(U=1 \mid s)}.$$

Taking logs proves (7). The causal substitution is the standard sequential identification step under the stated assumptions. □

This identity connects privileged likelihood to principled hindsight credit assignment (Harutyunyan et al., 2019; Meulemans et al., 2023), but also locates the missing assumption in free-form critique scoring. A text labeled "correct" need not be drawn from $P(v \mid s, U=1)$, and two separately prompted texts need not be conditionals of any common joint distribution. Even when they are coherent, the right side is a logit transform of binary outcome probability. It therefore gives ranking but does not generally guarantee the affine equality in Theorem 1; special finite-action configurations, including any strictly ordered two-action pair, can satisfy that equality accidentally.

A.3 KL IDENTITIES

A.3.1 SOFT-OPTIMAL TRAJECTORY RATIOS ARE A SPECIAL CASE

Proposition 6 (KL-regularized log-ratio telescoping; standard)**.** *Let $\pi_{\mathrm{ref}}(y \mid x)$ have full support, let autoregressive transitions be deterministic given the token history, and let $R(y, c)$ be a terminal trajectory reward distinct from task utility $U(y)$, with finite partition function $Z(x, c) = \mathbb{E}_{y \sim \pi_{\mathrm{ref}}(\cdot \mid x)} \exp(R(y, c)/\beta)$. Assume also that $\mathbb{E}_{\pi^\star} |R(y, c)| < \infty$ for the Gibbs distribution defined below. The unrestricted optimum over a candidate distribution π of*

$$\max_{\pi} \; \mathbb{E}_{y \sim \pi(\cdot \mid x, c)} R(y, c) - \beta D_{\mathrm{KL}}(\pi(\cdot \mid x, c) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid x)), \qquad \beta > 0,$$

is

$$\pi^\star(y \mid x, c) = \frac{\pi_{\mathrm{ref}}(y \mid x) \exp(R(y, c)/\beta)}{Z(x, c)}.$$

For the complete autoregressive trajectory, including termination,

$$\sum_t \log \frac{\pi^\star(v_t \mid s_t, c)}{\pi_{\mathrm{ref}}(v_t \mid s_t)} = \frac{R(y, c)}{\beta} - \log Z(x, c). \tag{8}$$

Under the corresponding soft Bellman equations, each summand equals the regularized soft advantage $(Q^\star_R(s_t, v_t; c) - V^\star_R(s_t; c))/\beta$, where $Q^\star_R$ is the soft-optimal continuation value for R before the current action's KL cost and $V^\star_R(s; c) = \beta \log \sum_u \pi_{\mathrm{ref}}(u \mid s) \exp(Q^\star_R(s, u; c)/\beta)$. These values are defined by R, not by the task utility U.

Proof. The exponential-tilt optimizer follows from the Gibbs variational identity. Taking the log ratio of its trajectory density to π_{ref} and applying the autoregressive chain rule gives (8). The per-token identity follows by substituting the regularized Bellman optimality equations for $Q^\star_R$ and $V^\star_R$. These are standard KL-control and preference- optimization identities (Geist et al., 2019; Rafailov et al., 2023; Donsker & Varadhan, 1975). □

For fixed (x, c), the term $-\log Z(x, c)$ is constant across trajectories. A grouped prompt baseline removes it only when compared trajectories share c, or at least share the same normalizer, and when sequence weighting, normalization, and clipping are compatible. With a different rollout-specific context c_i for each rollout, the normalizer can vary across examples. An arbitrary text-conditioned scorer need not be a soft-optimal policy and need not satisfy (8). A terminal verdict also does not uniquely determine how reward should be divided among tokens; a learned model can infer such a division only by adding structure from data or prior knowledge.

A.3.2 FORWARD- AND REVERSE-KL PROJECTIONS

Proposition 7 (Forward-KL projection)**.** *For a fixed state s and context distribution, define the mean target $\bar{q}_{\mathrm{score}}(v \mid s) = \mathbb{E}_{c \mid s} q_{\mathrm{score}}(v \mid s, c)$. Here $I(v; c \mid s)$ is computed under the joint distribution $p(c \mid s) q_{\mathrm{score}}(v \mid s, c)$. Then*

$$\mathbb{E}_{c \mid s} D_{\mathrm{KL}}(q_{\mathrm{score}}(\cdot \mid s, c) \,\|\, \pi_\theta(\cdot \mid s)) = I(v; c \mid s) + D_{\mathrm{KL}}(\bar{q}_{\mathrm{score}}(\cdot \mid s) \,\|\, \pi_\theta(\cdot \mid s)). \tag{9}$$

The minimizing policy is $\pi_\theta = \bar{q}_{\mathrm{score}}$, and the minimum is $I(v; c \mid s)$.

Proof. For a fixed state, add and subtract $\log \bar{q}_{\text{score}}(v \mid s)$:

$$\mathbb{E}_{c|s} D_{\text{KL}}(q_{\text{score}}(\cdot \mid s, c) \,\|\, \pi_\theta(\cdot \mid s)) \tag{10}$$

$$= \mathbb{E}_{c,v|s} \log \frac{q_{\text{score}}(v \mid s, c)}{\bar{q}_{\text{score}}(v \mid s)} + \mathbb{E}_{c,v|s} \log \frac{\bar{q}_{\text{score}}(v \mid s)}{\pi_\theta(v \mid s)} \tag{11}$$

$$= I(v; c \mid s) + D_{\text{KL}}(\bar{q}_{\text{score}} \| \pi_\theta), \tag{12}$$

where the second equality marginalizes c in the second term. Nonnegativity of KL gives the minimizer and minimum. At a fixed state, forward KL differs from cross entropy by a target-only constant, so differentiation through independent softmax logits gives $\nabla_{\mathbf{z}} \mathbb{E}_{c|s} D_{\text{KL}}(q_{\text{score}} \| \pi_\theta) = \pi_\theta(\cdot \mid s) - \bar{q}_{\text{score}}(\cdot \mid s)$. In the full expectation this term is weighted by the frozen state law. □

Reverse-KL centroid. Under common support and $\mathbb{E}_{c|s} |\log q_{\text{score}}(v \mid s, c)| < \infty$ for every v,

$$\begin{aligned}
&\mathbb{E}_{c|s} D_{\text{KL}}(\pi_\theta(\cdot \mid s) \,\|\, q_{\text{score}}(\cdot \mid s, c)) \\
&\quad = \sum_v \pi_\theta(v \mid s) \log \pi_\theta(v \mid s) \\
&\qquad - \sum_v \pi_\theta(v \mid s) \mathbb{E}_{c|s} \log q_{\text{score}}(v \mid s, c).
\end{aligned}$$

A Lagrange multiplier for $\sum_v \pi_\theta(v \mid s) = 1$ gives

$$\pi_{\text{geo}}(v \mid s) = \frac{\exp(\mathbb{E}_{c|s} \log q_{\text{score}}(v \mid s, c))}{\sum_u \exp(\mathbb{E}_{c|s} \log q_{\text{score}}(u \mid s, c))}.$$

Thus the arithmetic-centroid/conditional-mutual-information identity is specific to forward KL; it cannot be transferred unchanged to reverse KL.

Positive conditional mutual information without utility harm. Consider a one-step bandit. Let c be an independent style bit, and let $q_{\text{score}}(\cdot \mid s, c)$ choose between two different but task-equivalent surface-form actions. If terminal utility is one for both, then $I(v; c \mid s) > 0$, while every target-supported action has the same task value. The blind student's irreducible forward-KL loss is positive, but the projection creates no utility regret. Conversely, $I(v; c \mid s) = 0$ can occur when all contexts induce the same uninformative target. Either direction blocks a utility conclusion from the projection scalar alone.

A.4 Cross-fitting and partial policy gradients

Proof of Theorem 2. For the target-side calculation, freeze the donor law. By design,

$$p(y^{\text{don}}, y^{\text{tar}} \mid x, f) = p(y^{\text{don}} \mid x, f) \pi_{\text{old}}(y^{\text{tar}} \mid x, f).$$

For external randomness ξ independent of y^{tar} conditional on (x, f, y^{don}), and $c = \phi(x, f, y^{\text{don}}, \xi)$, marginalizing (y^{don}, ξ) gives

$$p(c, y^{\text{tar}} \mid x, f) = p(c \mid x, f) \pi_{\text{old}}(y^{\text{tar}} \mid x, f),$$

so $c \perp y^{\text{tar}} \mid x, f$. Conditional on frozen c, assume each $r_{c,k}$ is determined by the target history through token k and does not depend on later target tokens. Standard score-function differentiation and zero conditional covariance with earlier rewards then give the target-side gradient below, where $R_c(y) = \sum_k r_{c,k}$ is the feedback-defined return:

$$\nabla_\theta \mathbb{E}_{y^{\text{tar}} \sim \pi_\theta} \left[R_c(y^{\text{tar}}) \mid c \right] = \mathbb{E}\left[\sum_t G_{c,t} \nabla_\theta \log \pi_\theta(v_t \mid s_t) \mid c \right], \qquad G_{c,t} = \sum_{k \geq t} r_{c,k}, \tag{13}$$

assuming no additional direct θ-dependence (Williams, 1992; Schulman et al., 2015). If a larger objective also changes how donors or feedback are generated, this expression is only the part of its gradient that passes through the target rollout. □

Upstream dependence. If the donor is policy generated, $y^{\mathrm{don}} \sim \pi_\theta(\cdot \mid x, f)$, and $c = \phi(y^{\mathrm{don}}, \xi)$, then the intended joint objective has an upstream donor score term

$$\mathbb{E}_{y^{\mathrm{don}}, y^{\mathrm{tar}}, \xi}\left[R_{\phi(y^{\mathrm{don}},\xi)}(y^{\mathrm{tar}}) \nabla_\theta \log \pi_\theta(y^{\mathrm{don}} \mid x, f)\right],$$

plus any allowed pathwise derivatives. Cross-fitting removes a target's direct ancestry from its feedback; it does not erase policy dependence in donor generation.

What own-rollout agreement alone cannot show. Let a sampled action $v \in \{0, 1\}$ write its own feedback $c = v$, and score it by $R_c(v) = \mathbf{1}\{v = c\}$. Every realized action receives one. The feedback perfectly matches the sample yet defines no preference between actions and can be independent of task utility. Thus, own-rollout agreement alone cannot validate transferred action value. Held-out-feedback construction removes this direct dependence and asks a different question.

Independent but task-harming feedback. Let an independent donor $y^{\mathrm{don}} \sim \mathrm{Bernoulli}(0.9)$ set $c = y^{\mathrm{don}}$, and reward a target action v by $\mathbf{1}\{v = c\}$. Then $c \perp v$ before the target is sampled, but the held-out-feedback objective favors $v = 1$. If true utility is $\mathbf{1}\{v = 0\}$, optimizing that objective reduces task reward. Independence therefore tells us which objective is being optimized, not whether it is the right one.

Instantaneous versus future credit. Replacing $G_{c,t}$ by the instantaneous $r_{c,t}$ is exact only when

$$\mathbb{E}\left[\nabla_\theta \log \pi_\theta(v_t \mid s_t) \sum_{k>t} r_{c,k}\right] = 0.$$

Conditional action-independence of expected future reward is sufficient, but not necessary. A dense score at each position does not by itself justify omitting downstream effects.

A.5 Token-logit gradients for frozen targets

General chain rule. Suppressing the fixed state s, for token-logit vector $\mathbf{z}$, the softmax Jacobian is

$$\frac{\partial \pi_\theta(v')}{\partial z_v} = \pi_\theta(v')(\mathbf{1}\{v' = v\} - \pi_\theta(v)).$$

Therefore

$$\frac{\partial \mathcal{L}}{\partial z_v} = \sum_{v'} \frac{\partial \mathcal{L}}{\partial \pi_\theta(v')} \frac{\partial \pi_\theta(v')}{\partial z_v} = \pi_\theta(v)\left(\frac{\partial \mathcal{L}}{\partial \pi_\theta(v)} - \sum_{v'} \pi_\theta(v') \frac{\partial \mathcal{L}}{\partial \pi_\theta(v')}\right), \tag{14}$$

Reverse KL. For $\mathcal{L} = D_{\mathrm{KL}}(\pi_\theta \| \pi^\star)$, $\partial \mathcal{L}/\partial \pi_\theta(v) = \log(\pi_\theta(v)/\pi^\star(v)) + 1$, hence

$$\frac{\partial \mathcal{L}}{\partial z_v} = \pi_\theta(v)\left[\log \frac{\pi_\theta(v)}{\pi^\star(v)} - D_{\mathrm{KL}}(\pi_\theta \| \pi^\star)\right].$$

With two actions, let $p = \pi_\theta(v \mid s) = 1 - \varepsilon$ and denote the other token by $\bar{v}$. Direct cancellation gives

$$\frac{\partial \mathcal{L}}{\partial z_v} = p(1-p) \log \frac{p\pi^\star(\bar{v})}{(1-p)\pi^\star(v)}.$$

For fixed positive $\pi^\star(v), \pi^\star(\bar{v})$, this is $\Theta(\varepsilon \log(1/\varepsilon))$.

Generalized JSD. For $0 < \lambda < 1$,

$$\mathcal{L} = \lambda D_{\mathrm{KL}}(\pi_\theta \| m) + (1-\lambda) D_{\mathrm{KL}}(\pi^\star \| m), \qquad m = \lambda \pi_\theta + (1-\lambda)\pi^\star,$$

differentiation, including the dependence of m on π_θ, simplifies to $\partial \mathcal{L}/\partial \pi_\theta(v) = \lambda \log(\pi_\theta(v)/m(v))$. In the two-action case,

$$\frac{\partial \mathcal{L}}{\partial z_v} = \lambda p(1-p) \log \frac{pm(\bar{v})}{(1-p)m(v)}.$$

Because $m(\bar{v}) \to (1-\lambda)\pi^\star(\bar{v}) > 0$, this has the same $\Theta(\varepsilon \log(1/\varepsilon))$ order.

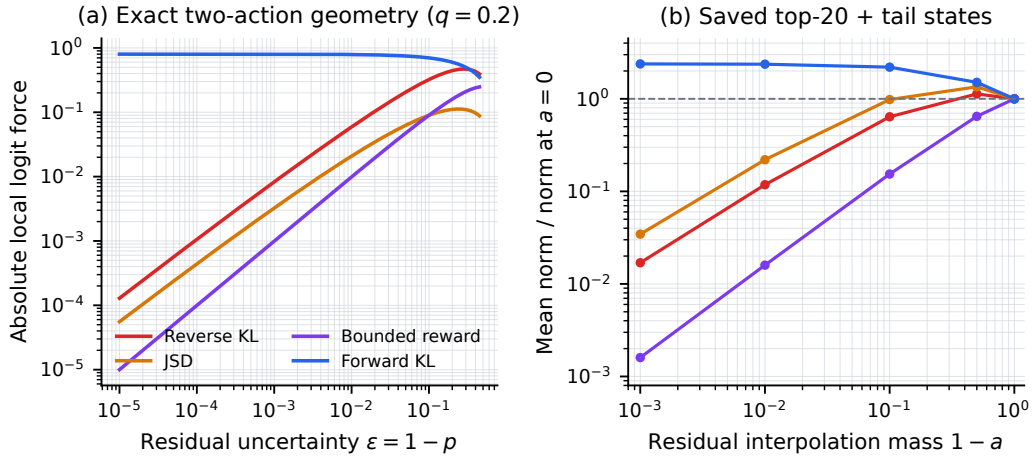


Figure 3: **Token-logit gradients for a frozen target.** The left panel evaluates exact two-action Euclidean-logit gradients as student confidence approaches one while a fixed target retains off-token mass. The right panel repeats the interpolation on 74,653 saved top-20-plus-tail distributions from the three dense-score configurations, with the remaining probability mass aggregated into one tail bin. Here $a \in [0, 1]$ is the interpolation weight toward the most likely token; the left panel's $q = 0.2$ denotes $\pi^\star(\bar{v} \mid s) = 0.2$. At $a = 0.999$, state-count-weighted Euclidean-logit gradient-norm ratios, normalized by their values at $a = 0$, are 0.0169, 0.0345, 0.0016, and 2.384 for reverse KL, generalized JSD, bounded reward, and forward KL. Constant loss rescaling changes their absolute magnitudes.

Bounded reward. For loss $\mathcal{L} = -\sum_{v'} \pi_\theta(v') d_\pm(v')$,

$$\frac{\partial \mathcal{L}}{\partial z_v} = p(1-p)\left(d_\pm(\bar{v}) - d_\pm(v)\right).$$

It is $O(\varepsilon)$ for bounded $d_\pm$ and $\Theta(\varepsilon)$ when the token-score gap $d_\pm(\bar{v}) - d_\pm(v)$ is bounded away from zero. The direction is corrective only when the score favors the desired alternative. A state-only baseline changes the realized advantage but not this expected gradient.

Forward KL. For $\mathcal{L} = D_{\mathrm{KL}}(\pi^\star \| \pi_\theta)$, $\partial \mathcal{L} / \partial \pi_\theta(v) = -\pi^\star(v)/\pi_\theta(v)$. Substitution into (14) yields $\partial \mathcal{L} / \partial z_v = \pi_\theta(v) - \pi^\star(v)$, and therefore $\partial \mathcal{L} / \partial z_v \to 1 - \pi^\star(v)$.

Scope of the singularity comparison. If near the relevant boundary $\mathcal{L}(\pi_\theta) = -\sum_v \rho(v) \log \pi_\theta(v) + O(1)$, with fixed weights and a remainder whose logit gradient vanishes, then $\mathcal{L}$ has forward-KL-like leading logit behavior. Faster singularities can also cancel or dominate the softmax Jacobian. The four losses derived above are compared in this common Euclidean-logit coordinate system.

A.6 Scalar token weights and expected parameter gradients

Proof of Theorem 4. By definition,

$$g_U = \mathbb{E}\left[\frac{A_i^U}{T_i} \sum_t h_{i,t}\right] \tag{15}$$

and

$$g_A = \mathbb{E}\left[\frac{A_i^U}{T_i} \sum_t w_{i,t} h_{i,t}\right].$$

Therefore

$$g_A - g_U = \mathbb{E}\left[\frac{A_i^U}{T_i} \sum_t (w_{i,t} - 1) h_{i,t}\right]. \tag{16}$$

Mean-one normalization only states $\sum_t (w_{i,t} - 1) = 0$; it does not state that the weighted sum of the generally distinct vectors $h_{i,t}$ is zero. □

Two-coordinate counterexample. Let v_1, v_2 be independent Bernoulli actions with separate logits and $P(v_1 = 1) = P(v_2 = 1) = 1/2$. Set

$$A^U = v_1 + v_2 - 1, \qquad (w_1, w_2) = (2 - \varepsilon, \varepsilon), \quad 0 < \varepsilon < 1.$$

The weights are positive and have mean one. Since the score for coordinate t is $v_t - 1/2$,

$$g_U = (1/8, 1/8), \qquad g_A = ((2 - \varepsilon)/8, \varepsilon/8).$$

The trajectory advantage keeps its sign and $(A^U/2)(w_1 + w_2) = A^U$, yet the expected policy-gradient vector changes.

Invariants that do survive. Before later clipping or normalization, $w_{i,t} > 0$ preserves each token advantage's sign, $A_i^U = 0$ remains zero, and mean-one weights preserve the scalar sequence budget. At a fixed, commonly observed position t, they do not preserve a prompt-group mean because, even if $\mathbb{E}_{i|x} A_i^U = 0$,

$$\mathbb{E}_{i|x}[A_i^U w_{i,t}] = \mathrm{Cov}_{i|x}(A_i^U, w_{i,t})$$

need not vanish. Variable-length masks must be included when the position is not common to all trajectories. The weights also do not preserve the expected policy-gradient estimate. If trajectory- or policy-dependent weights are recomputed and then detached, the resulting update rule need not be the gradient of any scalar objective. Thus, preserving scalar sign and total weight does not preserve the parameter update in verifier-anchored self-distillation (Yang et al., 2026; Pan et al., 2026).

Relation to reward redistribution. Valid return redistribution preserves a specified return or optimal-policy relation under explicit conditions (Arjona-Medina et al., 2019). Arbitrarily inserting mean-normalized weights into a score-function sum is not equivalent to constructing a reward process whose return-to-go yields the same policy gradient. The score vectors, not only the scalar total, determine that equivalence.

A.7 Sequence length and local optima

Proposition 8 (Elementary variable-length aggregation lemma)**.** *For each sequence length T, let*

$$\mathrm{Agg}_T(A^C) = \sum_{t=1}^{T} w_{T,t} A_t^C, \qquad w_{T,t} \geq 0.$$

Here $\mathbf{1}$ is the all-ones vector and e_j is the j-th standard basis vector. Fix $\mu \neq 0$ and $\delta > 0$. There are no constants $M < \infty$ and $\nu > 0$, uniform in T, such that both

$$|\mathrm{Agg}_T(\mu \mathbf{1})| \leq M \quad \textit{and} \quad \mathrm{Agg}_T(\delta e_j) \geq \nu\delta \quad \textit{for every } j \leq T \textit{ and every } T.$$

Proof. The singleton condition gives $w_{T,j} \geq \nu$ for every j. Hence $\sum_t w_{T,t} \geq \nu T$, and

$$|\mathrm{Agg}_T(\mu \mathbf{1})| = |\mu| \sum_t w_{T,t} \geq |\mu| \nu T,$$

contradicting a uniform bound. □

For the nonnegative linear aggregators defined in Theorem 8, a sum can retain an isolated event but accumulates a repeated offset as the sequence grows; a mean bounds that offset but gives an $O(1/T)$ response to a fixed number of events.

Proposition 9 (A fixed detached token score has a boundary optimum)**.** *At a fixed state, the maximizers of*

$$\sum_v \pi_\theta(v \mid s) d_\pm(s, v)$$

over the probability simplex are exactly the distributions supported on $\arg\max_v d_\pm(s, v)$*. With a unique maximizing token, the optimizer is a point mass.*

Proof. The displayed objective is a linear functional on the simplex. Any mass on an action below the maximum can be moved to a maximizing action to increase the objective. □

A strictly increasing, policy-independent pointwise transformation preserves token ordering and the same boundary maximizers; non-strict clipping can introduce ties.

A.8 KL REGULARIZATION CREATES A FINITE TARGET, NOT A CORRECT ONE

Proposition 10 (KL-regularized optimum; standard)**.** *Fix a finite two-sided score $d_\pm(s,\cdot)$, a full-support reference policy $\pi_{\mathrm{ref}}(\cdot \mid s)$, $\alpha \in \mathbb{R}$, and $\beta > 0$. Define*

$$F_{\mathrm{KL}}(\pi_\theta; s) = \alpha \mathbb{E}_{v\sim\pi_\theta(\cdot|s)} d_\pm(s,v) - \beta D_{\mathrm{KL}}(\pi_\theta(\cdot \mid s) \,\|\, \pi_{\mathrm{ref}}(\cdot \mid s)) \,.$$

Then

$$\pi^\star(v \mid s) = \frac{\pi_{\mathrm{ref}}(v \mid s) \exp((\alpha/\beta) d_\pm(s,v))}{Z_s}, \tag{17}$$

$$F_{\mathrm{KL}}(\pi_\theta; s) = -\beta D_{\mathrm{KL}}(\pi_\theta(\cdot \mid s) \,\|\, \pi^\star(\cdot \mid s)) + \beta \log Z_s, \tag{18}$$

where $Z_s = \sum_v \pi_{\mathrm{ref}}(v \mid s) \exp((\alpha/\beta) d_\pm(s,v))$. Thus $\pi^\star$ is the unique maximizer.

Proof. Substitute

$$\log \pi^\star(v \mid s) = \log \pi_{\mathrm{ref}}(v \mid s) + (\alpha/\beta) d_\pm(s,v) - \log Z_s$$

into $D_{\mathrm{KL}}(\pi_\theta \| \pi^\star)$ and rearrange. Uniqueness follows from strict convexity of KL in its first argument. This is the standard Gibbs/Donsker–Varadhan variational identity used in KL-regularized control (Donsker & Varadhan, 1975; Geist et al., 2019). □

The result guarantees a finite local target, not a good one. Its target is defined jointly by π_{ref} and the uncalibrated score $d_\pm$; it can be a finite-confidence target for the wrong action. If either object is refreshed, $\pi^\star$ is a per-update target rather than a demonstrated global equilibrium.

For the implemented KL configurations, let

$$d_\pm^{\mathrm{clip}}(s,v) = \mathrm{clip}(d_\pm(s,v), -2, 2), \qquad \ell(s,v) = \log \frac{\pi_{\mathrm{old}}(v \mid s)}{\pi_{\mathrm{ref}}(v \mid s)}.$$

Their clipped and full token advantages are

$$A_{i,t}^{C,\mathrm{clip}} = \mathrm{ctr}_{s_{i,t}}[d_\pm^{\mathrm{clip}}](v_{i,t}), \tag{19}$$

$$A_{i,t}^{C,\mathrm{full}} = \mathrm{ctr}_{s_{i,t}}[\alpha d_\pm^{\mathrm{clip}} - \beta \ell](v_{i,t}). \tag{20}$$

For the projected-KL variant, let

$$\widehat{\gamma}_s = \frac{\mathrm{Cov}_{v\sim\pi_{\mathrm{old}}(\cdot|s)}\left(d_\pm^{\mathrm{clip}}(s,v), \ell(s,v)\right)}{\mathrm{Var}_{v\sim\pi_{\mathrm{old}}(\cdot|s)}\left(d_\pm^{\mathrm{clip}}(s,v)\right)}. \tag{21}$$

Set $\widehat{\gamma}_s = 0$ when the denominator is zero, and let the configured scale be $\kappa > 0$. Its token advantage is

$$A_{i,t}^{C,\mathrm{proj}} = \kappa(\alpha - \beta \widehat{\gamma}_{s_{i,t}}) A_{i,t}^{C,\mathrm{clip}}. \tag{22}$$

The projected rule in (22) retains only the least-squares component of the reference-policy correction that varies with the centered feedback score. Because it keeps that score-aligned correction and discards the orthogonal remainder, it need not correspond to this full-distribution objective.

A.9 BASELINE DETAILS

At the on-policy point $\pi_\theta = \pi_{\mathrm{old}}$, let $d_\pm(s,\cdot)$ be fixed before $v \sim \pi_{\mathrm{old}}(\cdot \mid s)$ is sampled, and let $h(v) = \nabla_\theta \log \pi_\theta(v \mid s)$. Since $\mathbb{E}_{v\sim\pi_{\mathrm{old}}}[h(v) \mid s] = 0$, every state-only scalar baseline $b(s)$ satisfies

$$\mathbb{E}[(d_\pm(s,v) - b(s))h(v) \mid s] = \mathbb{E}[d_\pm(s,v)h(v) \mid s].$$

This cancellation is the unclipped on-policy score-function identity for a score fixed before sampling. If the context $c(y)$ is written from the sampled rollout y, the score is trajectory dependent, so the identity applies only after conditioning on a fixed feedback context. The baseline minimizing the scalar residual $\mathbb{E}[(d_\pm(s,v) - b(s))^2 \mid s]$ is $b(s) = \mathbb{E}[d_\pm(s,v) \mid s]$, which is (5). The baseline minimizing the trace of the one-state vector estimator's covariance is instead

$$b^\star(s) = \frac{\mathbb{E}[d_\pm(s,v)\|h(v)\|^2 \mid s]}{\mathbb{E}[\|h(v)\|^2 \mid s]}.$$

These coincide only under additional conditions. For sampled token v with $p = \pi_{\text{old}}(v \mid s)$, the same-slot baseline also exposes

$$d_\pm(s,v) - \mathbb{E}_{u\sim\pi_{\text{old}}(\cdot|s)} d_\pm(s,u) = (1-p)\left(d_\pm(s,v) - \frac{\sum_{u\neq v}\pi_{\text{old}}(u \mid s) d_\pm(s,u)}{1-p}\right).$$

This factorization describes a realized centered advantage. The vanishing expected softmax-logit gradients in Theorem 3 do not depend on choosing this particular baseline.

A.10 Additional protocol specification

A.10.1 One-sided score audit

For a fixed normalized scorer $q_{\text{score}}(\cdot \mid c^+)$, rollout policy π_{old}, and the set $\mathcal{G}$ of correct trajectories,

$$\mathbb{E}_{\pi_{\text{old}}(\cdot|\mathcal{G})} \log \frac{q_{\text{score}}(y \mid c^+)}{\pi_{\text{old}}(y)} = \log \frac{q_{\text{score}}(\mathcal{G} \mid c^+)}{\pi_{\text{old}}(\mathcal{G})} - D_{\text{KL}}\big(\pi_{\text{old}}(\cdot \mid \mathcal{G}) \,\|\, q_{\text{score}}(\cdot \mid c^+, \mathcal{G})\big).$$

The outcome-conditioned mean therefore combines the change in teacher mass on $\mathcal{G}$ with the within-$\mathcal{G}$ KL term. A negative conditional mean establishes lower geometric likelihood on the observed successful paths; the first term separately determines the teacher's total mass on $\mathcal{G}$. With rollout-specific feedback $c^+ = c^+(y)$, the diagonal scores do not in general form one normalized teacher distribution.

Table 4 reconstructs the positive teacher leg on fixed rollouts from two configurations that use the two-sided score. In both analyses, more than 99% of regraded correct trajectories have a negative complete-trajectory teacher–policy log ratio.

Table 4: **Fixed-rollout one-sided audit.** $\sum_t d(s_t, v_t)$ is the complete-trajectory one-sided teacher–policy log ratio. The correct and incorrect columns report the percentage of each class with a negative ratio. AUC uses the response-mean token residual, with 95% prompt-bootstrap intervals. These rows extract the positive leg from two experiments that use the two-sided score.

Configuration	Correct	Incorrect	AUC [95% CI]
Source-clipped additive trajectories	99.49%	99.89%	0.435 [0.393, 0.474]
Saliency-modulated trajectories	99.28%	99.51%	0.512 [0.470, 0.547]

A.10.2 Score provenance

Table 5 defines every score in Tables 1 and 6. The external LLM judge writes feedback; q_{score} is instantiated by the listed `gpt-oss-20b` checkpoint and supplies all displayed privileged token likelihoods.

Table 5: **Score definitions and model provenance.** "Frozen/earlier" means that the scoring checkpoint is not updated through the score being measured.

Displayed score	Feedback construction and writer	Likelihood scorer
Own-rollout feedback	Rollout-specific helpful and adversarial feedback; `gpt-5.6-luna`	frozen/earlier `gpt-oss-20b`
Implemented additive	Value-masked positive feedback against a same-problem mixture of plausible near-miss negatives; `gpt-5.6-luna`	frozen reference `gpt-oss-20b`
Token-weighting configuration, score sum	Value-masked positive feedback and plausible near-miss negative feedback; `gpt-5.6-luna`	frozen/earlier `gpt-oss-20b`
Held-out masked feedback, sum	Value-masked/near-miss feedback from opposite-fold rollouts of the same problem; `gpt-5.6-luna`	frozen `gpt-oss-20b`
Held-out masked feedback, mean	Same feedback as the preceding row, reduced by token mean; `gpt-5.6-luna`	frozen `gpt-oss-20b`
Held-out correct-fact feedback, mean	Correct versus incorrect interior facts from opposite-fold rollouts of the same problem; `gpt-5.6-luna`	frozen `gpt-oss-20b`
Different-problem correct-fact control	The preceding correct-fact construction with feedback from a different problem; `gpt-5.6-luna`	frozen `gpt-oss-20b`
Identical-feedback null	Identical positive and negative contexts; `gpt-5.6-luna`	frozen `gpt-oss-20b`

Table 6: **Secondary trajectory-score diagnostics.** Entries are AUCs with 95% prompt-bootstrap confidence intervals.

Score construction	Raw pooled AUC	Length-adjusted AUC	Within-prompt AUC
Token-weighting configuration, score sum	0.553 [0.517, 0.585]	0.528 [0.493, 0.560]	0.547 [0.506, 0.583]
Held-out masked feedback, sum	0.453 [0.384, 0.526]	0.489 [0.431, 0.555]	0.520 [0.463, 0.584]
Different-problem correct-fact control	0.475 [0.418, 0.530]	0.528 [0.467, 0.583]	0.520 [0.458, 0.577]

Matched responses and format sensitivity. The own-rollout analysis uses 3,361 responses with a unique trajectory match.

Table 7 reports a format/completion sensitivity analysis among responses that both completed and contained a boxed answer. Because completion and formatting are post-generation behavior, this condition changes the population being ranked. Intervals use the same prompt-bootstrap procedure as Table 1.

Table 7: **Raw AUC among boxed, completed responses.** Brackets are 95% prompt-bootstrap confidence intervals.

Score	AUC [95% CI]
Own-rollout feedback	0.595 [0.510, 0.675]
Implemented additive	0.506 [0.486, 0.528]
Token-weighting configuration, score sum	0.552 [0.518, 0.588]
Held-out masked feedback, sum	0.451 [0.382, 0.515]
Held-out masked feedback, mean	0.514 [0.451, 0.573]
Held-out correct-fact feedback, mean	0.416 [0.344, 0.490]
Different-problem correct-fact control	0.472 [0.417, 0.526]
Identical-feedback null	0.503 [0.499, 0.507]

The token-weighting row in Table 6 reports a sum of signed contrasts, whereas the training weights use the mean contrast magnitude.

Label and feedback audits. Among 49 gradeable strong-consensus cases from a targeted audit, 18 (36.7%) were equivalent to the reference. This estimate is conditional on the selected strong-consensus cases and is not a corpus-wide label-error rate.

An external-judge pilot selected the better solution in 94.4% $(51/54)$ of valid judgments.

The c^+ and c^- prompts also differ slightly beyond their mathematical content. The positive template masks selected values, asks the model to work them out, and more often includes formatting

instructions, whereas the negative template supplies explicit near-miss values. The likelihood contrast can therefore reflect template wording as well as the intended mathematical distinction.

A.10.3 CROSS-FITTED CONSTRUCTION

Within each prompt group, fold labels are assigned by a seeded, content-independent permutation after the rollouts and their feedback records have been generated, but before donor selection and target scoring. Donor feedback for a target fold comes only from trajectories in the other fold. A target rollout is excluded from every record used to write its feedback. If too few donors exist, the example is marked unavailable rather than silently using feedback from the target itself. Donor count and aggregation (one donor, a fixed pool, or a declared log-mean-exp over negatives) are reported with each score.

We verify that swapping c^+ and c^- negates finite contrasts, setting $c^+ = c^-$ produces a numerical null, and the scored rollout never appears among the records used to write its feedback. Different-problem donors and template-only pairs serve as controls for problem-specific information and wording.

A.10.4 SCORE REDUCTION AND TRAINING-RULE CONFIGURATIONS

Raw sequence sums, token means, centered or squashed scores, and own-rollout or cross-fitted scores are distinct measurements. Table 8 specifies the feedback reduction, clipping, gating, and combination with outcome credit for each training configuration. Every matched configuration uses $A_i^U = U_i - \frac{1}{K-1}\sum_{j\neq i} U_j$, with the same length and validity handling, and the outcome-only control sets the token advantage to zero. The same-state baseline in (5) was approximated with the scorer's top-k tokens plus one bin for all remaining probability mass. The table states whether that aggregate tail-bin contribution was included.

Table 8: **Training configurations.** All configurations share the base training recipe, post-combination advantage clipping, and subsequent policy-ratio clipping. Each row reports the evaluated training recipe, including the jointly specified choices in the KL-regularized variants.

Configuration	How the token score enters training
Outcome-only	Uses A_i^U; token advantage identically zero.
Entropy-gated additive	Uses $0.5\tanh(A_{i,t}^C/0.5)$, multiplies it by $\min(1, H_{i,t}/0.2)$, gates it on nonzero A_i^U, prevents it from reversing the sign of A_i^U, and omits the aggregate tail bin.
Source-clipped additive	Clips every $d_\pm(s_{i,t}, v)$ to $[-2, 2]$ before centering, then uses $0.5\tanh(A_{i,t}^C/0.5)$; gates it on nonzero A_i^U, prevents reversal of the sign of A_i^U, and omits the aggregate tail bin.
Entropy-gated + source-clipped	Combines the preceding source clip, $\tanh$ reduction, and entropy multiplier; gates on nonzero A_i^U, prevents reversal of its sign, and omits the aggregate tail bin.
Full KL-regularized	Uses (20) with $\alpha = \beta = 1$, uses no $\tanh$, gates on nonzero A_i^U, and includes the aggregate tail bin.
Projected KL correction	Uses (22) with $\alpha = \beta = 1$ and $\kappa = 0.25$, uses no $\tanh$, uses the same nonzero-A_i^U gate, and includes the aggregate tail bin.

The execution order is: construct A_i^U and $A_{i,t}^C$; apply the dense gate and any entropy multiplier; combine the channels; apply the additive sign clamp where configured; clip the combined advantage to $[-1, 1]$; mask response tokens; and finally apply PPO importance-ratio clipping in the actor loss. Advantage normalization is disabled. Final advantage clipping can discard an added score when the utility advantage is already at the clip limit, or reduce it when the added score has the opposite sign.

Mechanism analysis. In a separately analyzed ungated additive configuration, 57%–71% of absolute scalar weight fell in the highest-entropy token decile. Across the matched configurations through update 25, the additive variants moved toward higher token entropy and reference-policy KL, while the KL-regularized variants remained near the outcome-only range.

A.10.5 PROMPT-LEVEL COMPARISONS AND UNCERTAINTY

For training configuration ℓ, prompt x, update n, and rollout i, let $\widehat{U}_{\ell,x,n,i} \in \{0,1\}$ be the corrected equivalent-answer label and let

$$\widehat{p}_{\ell,x,n} = \frac{1}{K_{\ell,x,n}} \sum_{i=1}^{K_{\ell,x,n}} \widehat{U}_{\ell,x,n,i}$$

be the corresponding prompt accuracy. The primary paired difference at a fixed update is

$$\widehat{\Delta}_n = \frac{1}{|\mathcal{X}_n|} \sum_{x \in \mathcal{X}_n} (\widehat{p}_{\ell,x,n} - \widehat{p}_{0,x,n}),$$

where subscript 0 denotes the outcome-only control and $\mathcal{X}_n$ contains prompts observed in both configurations. Prompt-cluster bootstrap samples prompts, carrying all their rollouts together. The matched comparison uses one seed per configuration, and its checkpoints are repeated measurements from that seed.

The paired configuration-label sign-flip test enumerates the prompt-level differences under a sharp exchangeability null for the two configuration labels within each paired prompt. The reported p-value is the exact tail probability under that null.

Across the ten method-by-update contrasts, the Bonferroni threshold is 0.005; all five update 49 exact p-values remain below this threshold. The reported confidence intervals are unadjusted.

Completion check. At update 49, the outcome-only configuration was 76.7% boxed and complete with 23.3% truncation, whereas the five token-score configurations were 94.2% to 98.3% boxed, 95.8% to 100% complete, and 0% to 4.2% truncated. The lower correctness of the token-score configurations occurs despite their higher completion and lower truncation.

For outcome-ranking analyses, the bootstrap likewise carries all trajectories from a prompt together. We report pooled AUC, within-prompt AUC, score–length association, a predeclared length/format/completion adjustment, and parameter-gradient projection as separate statistics because they answer different questions. AUC screens outcome ranking, while Theorem 1 characterizes exact local equivalence.

A.10.6 SHARED-CHECKPOINT SELECTION

We compare configurations at the latest checkpoint for which every configuration has enough evaluation responses. This rule selects update 49 and excludes per-configuration best-checkpoint selection. All configurations use the same AIME 2025 evaluation set, and each paired effect is recomputed on $\mathcal{X}_n$, the prompt intersection for that configuration and outcome-only GRPO at update n. Figure 4 reports these paired effects at every eligible checkpoint.

A.10.7 EQUIVALENT-ANSWER AUDIT

The evaluator first checks normalized exact equality and then symbolic equivalence subject to domain restrictions. A numerical fallback uses fixed points and precision chosen without configuration labels, while missing, malformed, and truncated answers remain separate format outcomes. High-consensus answers that disagree with the reference enter blinded adjudication. Regrading changed seven rollout labels in the matched evaluation.

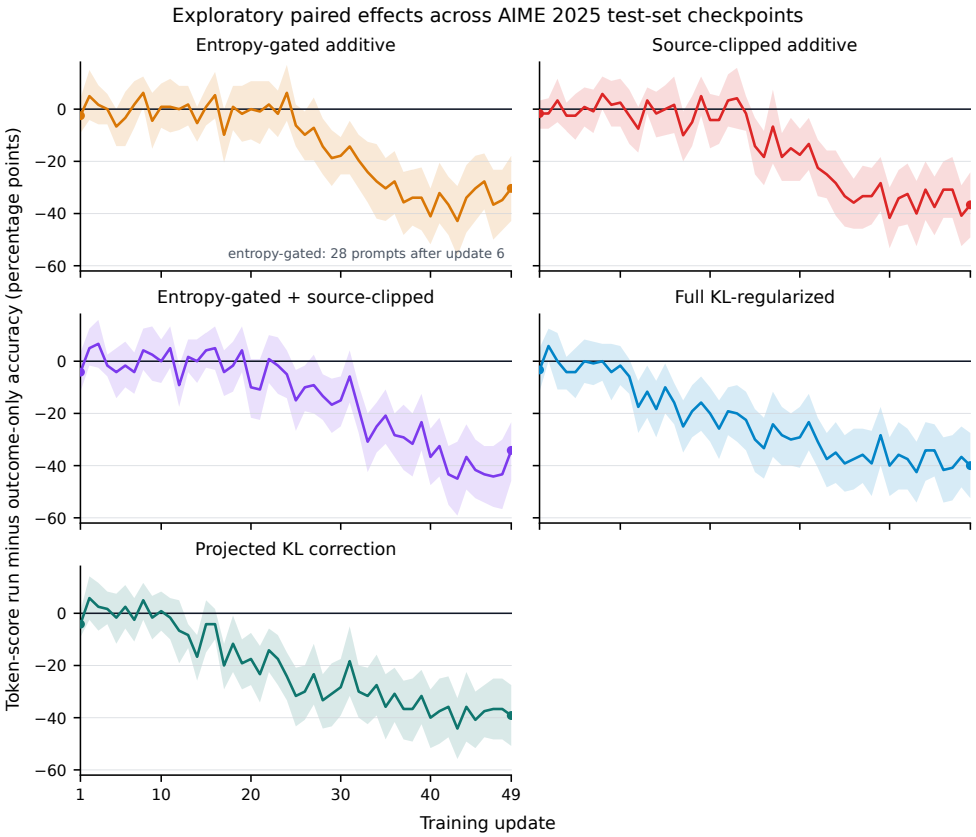


Figure 4: **Prompt-paired effects at every eligible checkpoint.** Each line is token-score configuration minus outcome-only accuracy, recomputed on the same prompts at each update. Shading is a pointwise 95% prompt-bootstrap interval with 20,000 resamples. All comparisons use the same AIME 2025 evaluation set. Update 49 is the fixed comparison.