Separating Signal from Noise: SNR-Guided Reward for Active Reasoning Agents

Anonymous ACL submission

Abstract

In multi-turn active reasoning, an agent's steps contribute differently to the final outcome: some queries eliminate hypotheses, others stall or lead the agent into belief traps. Standard reinforcement learning methods including those with trajectory-level truncation, distribute credit uniformly across retained steps. We formalize step-level credit assignment as a heteroscedastic linear estimation problem and derive its closed-form Best Linear Unbiased Estimator (BLUE): each step is weighted in proportion to its precision. We translate this prescription into a plug-and-play modifier. On two distinct discrete and continuous tasks, our method improves PPO+T3 by $+7.67$ EM on MovieRecommendation ($38.00 \rightarrow 45.67$) and $+5.00$ EM on CircuitDecoding ($77.83 \rightarrow 82.83$) using a single set of hyperparameters; the 7B base model with our method *exceeds* o3-mini on MovieRecommendation. A controlled ablation attributes $+4.33$ EM to the step-level signal channel and a further $+3.34$ EM (43.5% of the SNR gain) to BLUE-based precision weighting—two complementary mechanisms with substantial individual contributions. We further identify the operating regime in which the BLUE assumption holds, characterizing how SNR weighting and trajectory truncation interlock without redundancy.

1 Introduction

In an active reasoning task, an agent's queries are not equally informative. A well-chosen question may halve the hypothesis space; a repeated query yields nothing new; one issued from a stalled belief produces noise rather than evidence. Despite this clear heterogeneity, standard reinforcement learning treats all steps as equally informative when distributing the final reward. We show that this heterogeneity matters quantitatively, formalize it as a heteroscedastic estimation problem, and demonstrate that the Best Linear Unbiased Estimator (BLUE) prescribes a simple, plug-and-play correction that improves PPO+T3 by $+7.67$ EM on MovieRecommendation and $+5.00$ EM on CircuitDecoding, enabling a 7B model to exceed o3-mini on movie recommendation.

The credit assignment bottleneck in active reasoning. Large language models, increasingly trained with reinforcement learning on outcome rewards (Shao et al., 2024; Zheng et al., 2025), are now deployed in multi-turn, information-seeking settings (Zhou et al., 2025; Badola et al., 2025). Such active reasoning unfolds over a long horizon and is rewarded only at termination, so credit must flow back to each intermediate decision. Existing methods—PPO (Schulman et al., 2017) with GAE, group-relative advantages in GRPO (Shao et al., 2024)/GSPO (Zheng et al., 2025), and trajectory-truncation methods (Zou et al.)—all share one assumption: within the steps they retain, each step contributes equally on average. In active reasoning, where step-level informativeness varies by orders of magnitude.

Credit assignment as heteroscedastic estimation. We reframe the problem. If each step produces an observable progress signal g_t that is a noisy proxy for its true contribution, then credit assignment is a classical statistical question: given heteroscedastic noisy observations summing to a known total, what is the best linear unbiased estimate of each observation's true value? The Gauss–Markov theorem answers directly: the BLUE weights each step by its precision, high-reliability steps retain their raw signal, low-reliability steps are softly pulled toward the budget-implied estimate. To our knowledge, this is the first formulation of active reasoning credit assignment as a heteroscedastic estimation problem with a closed-form optimal solution. Our contributions are as follows.

- **A statistical reframing** of step-level credit assignment as a heteroscedastic linear esti-

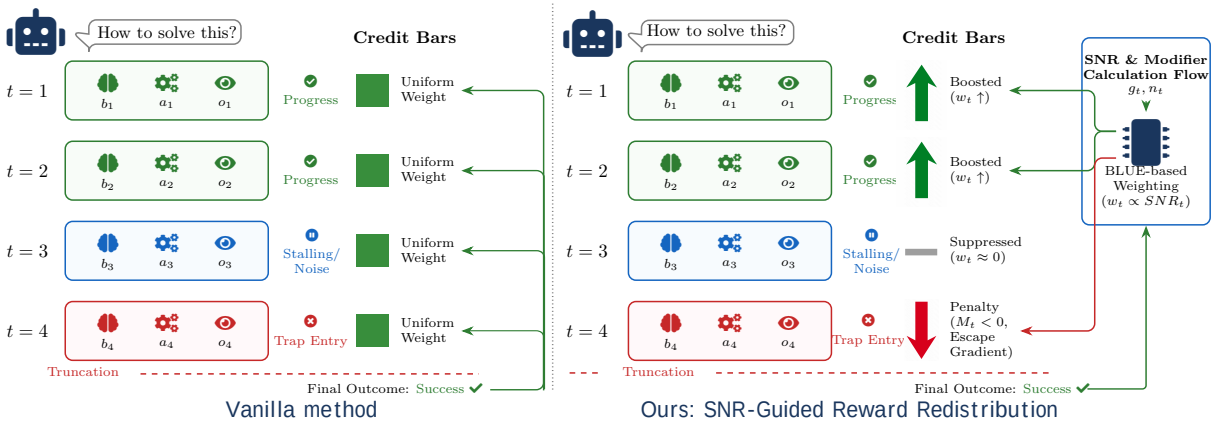

Figure 1: Overview of SNR-guided reward redistribution. **(a)** Steps in active reasoning have heterogeneous informativeness: informative queries (green) vs. repeated/trapped steps (orange/red). **(b)** The BLUE-derived weight w_t assigns high precision to novel productive steps and suppresses noisy ones. **(c)** The modifier $M_t = \lambda w_t g_t$ integrates additively with any base advantage estimator.

mation problem with a closed-form BLUE (Theorem 1).

- **A plug-and-play modifier** $M_t = \lambda w_t g_t$ that approximates the BLUE using only observable rollout indicators, integrating additively with any base advantage estimator.
- **Mechanistic characterization** via two complementary lemmas: automatic BTR suppression (Lemma 1) and explicit escape gradients (Lemma 2).
- **Quantitative validation** on two structurally distinct AR-Bench tasks (Zhou et al., 2025): $+7.67$ EM on MovieRecommendation (38.00 $\rightarrow$ 45.67, exceeding o3-mini's 44.67) and $+5.00$ EM on CircuitDecoding (77.83 $\rightarrow$ 82.83) with a single set of hyperparameters; ablations attribute 43.5% of the SNR gain to BLUE weighting.

2 SNR-Guided Reward Redistribution for Active Reasoning

2.1 Problem Setup

Active reasoning as multi-turn interaction. An active reasoning agent must identify an unobservable target state $s^\star$ (e.g., a hidden circuit configuration, a user's latent preference) through a sequence of queries to an environment. Following the standard formulation (Zou et al.), the interaction unfolds over up to T turns: at each step t, the agent issues an action a_t conditioned on the history $h_t = (a_1, o_1, \ldots, a_{t-1}, o_{t-1})$, observes a response o_t, and updates its internal belief about $s^\star$. The trajectory-level reward is delivered only at $t = T$. The agent is trained via policy optimization to maximize $\mathbb{E}_{\pi_\theta}[R]$.

The credit assignment problem. Because the only learning signal is the terminal reward R, gradient-based optimization must implicitly attribute the single scalar outcome to the T intermediate actions. Standard advantage estimators—GAE in PPO (Schulman et al., 2017), group-relative advantages in GRPO (Shao et al., 2024) and GSPO (Zheng et al., 2025)—perform this attribution either by temporal-difference propagation or by uniform group-level normalization. Both treat all steps as equally informative about the outcome. But in active reasoning, an informative query eliminating half the hypothesis space contributes far more than a repeated or stalled query at the same trajectory position.

Step-level observation model. While the agent's true belief $b_t \in \Delta(\mathcal{S})$ is not directly accessible, every active reasoning task provides rollout-level signals that locally indicate whether a step advanced the reasoning state. We summarize this as a scalar *progress signal* $g_t \in \mathbb{R}$, with $\mathrm{sign}(g_t) \in \{+1, 0, -1\}$ encoding progress, neutrality, or regression (task-specific instantiations such as candidate-set reduction and feedback validity are deferred to Appendix B.1). Crucially, g_t is computable from rollouts without access to $s^\star$. However, g_t is a *noisy* observation of each step's true contribution, and the noise level varies systematically: a repeated query yields $g_t \approx 0$ regardless of whether the original was informative (high noise); a novel early query produces a sharp

signal (low noise); a query from a stalled belief produces an observation determined by the agent's incoherent context (unreliable). We model this as

$$g_t = \mu_t + \varepsilon_t, \quad \varepsilon_t \sim (0, \sigma_t^2), \tag{1}$$

where μ_t is the latent true credit of step t and σ_t^2 varies across steps. This heteroscedasticity is not a modeling convenience: it is the empirical reality of multi-turn LLM agents, as we verify in Sec. 3.5.

Toward a reliability-weighted estimator. Standard advantage estimators implicitly assume homoscedastic noise and are therefore suboptimal under Eq. (1). A natural alternative weights each step by its reliability: trust precise signals, down-weight noisy ones. The rest of this section derives this estimator from first principles, translates it into a rollout-computable modifier, and characterizes its learning dynamics.

2.2 The BLUE for Step-Level Credit

The reliability principle admits a precise formulation: among all linear unbiased estimators that respect the trajectory-level reward budget, which minimizes total variance under heteroscedastic noise? The answer is the Best Linear Unbiased Estimator (BLUE), which weights each step in direct proportion to its precision.

Let $\mu_t \in \mathbb{R}$ denote the latent true credit of step t with $\sum_t \mu_t = R$. We observe each g_t as a noisy realization of μ_t with variance σ_t^2. We seek a credit estimator $\hat{\mu}_t$ that is (i) linear in $\{g_\tau\}$, (ii) unbiased ($\mathbb{E}[\hat{\mu}_t] = \mu_t$), (iii) budget-consistent ($\sum_t \hat{\mu}_t = R$), and (iv) of minimum total variance.

Theorem 1 (BLUE for step-level credit)**.** *Under model* (1) *with* $\sum_t \mu_t = R$*, the BLUE for* μ_t *is*

$$\hat{\mu}_t^\star = g_t + \frac{\sigma_t^2}{\sum_\tau \sigma_\tau^2} \cdot \big(R - \textstyle\sum_\tau g_\tau\big). \tag{2}$$

Proof sketch. Parametrize $\hat{\mu}_t = g_t + c_t\delta$ where $\delta \triangleq R - \sum_\tau g_\tau$. Since $\mathbb{E}[\delta] = 0$, the estimator is unbiased for any $\{c_t\}$; the budget constraint reduces to $\sum_t c_t = 1$. Minimizing the resulting total variance subject to this constraint yields $c_t^\star = \sigma_t^2 / \sum_\tau \sigma_\tau^2$. Full derivation in Appendix A.1. □

The signal retention coefficient $\rho_t \triangleq 1 - \sigma_t^2 / \sum_\tau \sigma_\tau^2$ admits a direct interpretation: $\hat{\mu}_t^\star$ is a convex combination of the raw observation g_t and a budget-implied estimate, with weight monotonically increasing in the step's precision $\pi_t = 1/\sigma_t^2$. Low-noise steps retain their raw signal ($\rho_t \to 1$); high-noise steps are softly pulled toward the budget-implied value ($\rho_t \to 0$). This precision-proportional structure is the theoretical anchor of our method.

The BLUE requires the true variances $\{\sigma_t^2\}$, which are not directly observable. We approximate the precision using a rollout-computable signal-to-noise ratio

$$\mathrm{SNR}_t = \frac{\max(g_t, 0)}{n_t + \epsilon}, \tag{3}$$

where n_t aggregates observable noise indicators (defined in Sec. 2.3). Under mild assumptions on the comparability of true credits across steps (Appendix A.2), SNR_t preserves the monotone ordering of the BLUE weights. The core modifier $M_t = \lambda \cdot \mathrm{SNR}_t \cdot g_t$ amplifies high-reliability signals and suppresses low-reliability ones, directly instantiating Theorem 1.

Two practical issues remain. First, the theorem only governs positive-progress steps; it is silent on how to handle regression events ($g_t < 0$). Second, raw SNR_t can be unbounded when $n_t \to 0$ at early steps. The next subsection resolves both within a unified weight function.

2.3 A Unified Modifier

We construct a unified step-level modifier $M_t = \lambda \cdot w_t \cdot g_t$, where the step weight w_t adapts to the epistemic role of step t.

Noise indicator. Following the BLUE structure, we estimate each step's noise level as $n_t = \mathrm{rep}_t + \mathrm{prox}_t$, the sum of two history-derived integers. The repetition count $\mathrm{rep}_t = |\{\tau < t : a_\tau = a_t\}|$ tracks how often the current action has been issued before—repeated queries yield no new information and inflate observation noise. The non-progress proximity $\mathrm{prox}_t = \max(0, K - |\{j \in [t-K, t) : g_j > 0\}|)$ counts non-progressive steps within the most recent K steps, flagging belief-stalling regions. Both indicators require no learned components.

Weight function. The weight w_t takes one of three forms dispatched by the sign and context of g_t:

$$w_t = \begin{cases} \mathrm{clip}\left(\dfrac{g_t}{n_t + \epsilon}, 0, \gamma_{\mathrm{clip}}\right) & g_t > 0 \\ \beta_{\mathrm{prec}} & g_t < 0,\ t \in \mathcal{P} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $\beta_{\text{prec}} = \mu/\lambda > 0$ and $\mathcal{P} = [t_{\text{neg}} - K, t_{\text{neg}})$ is the K-step window preceding the nearest negative-progress event t_{neg}. The first branch (PRODUCTIVE) instantiates the BLUE principle: novel productive steps with low n_t receive large weight; repeated or stalling steps are down-weighted. The second branch (PRECURSOR) provides an *escape gradient*: $M_t = \mu g_t < 0$ for steps adjacent to regression events; restricting the penalty to the precursor window enforces a causal interpretation that isolated negative observations are not penalized. The third branch (NEUTRAL) leaves the base advantage unmodified.

Integration with policy gradient. The modifier is added directly to any base advantage estimator,

$$A_t \leftarrow A_t^{\text{base}} + M_t, \tag{5}$$

preserving the unbiasedness of A_t^{base} while injecting step-level credit information. The base estimator can be the GAE estimate (PPO), the group-relative advantage (GRPO, GSPO), or any other standard form; our method is algorithm-agnostic and introduces no learned components, no additional model forward passes, and no architectural changes. The procedure adds $O(T)$ cost per trajectory. The single set of hyperparameters used across all experiments is detailed in Appendix B.4.

2.4 Learning Dynamics: Two Mechanisms

The unified modifier $M_t = \lambda w_t g_t$ reshapes learning through two complementary mechanisms in the presence of *belief trap regions* (BTRs) (Zou et al.)—regions of belief space where the agent's update rule fails to make progress and subsequent queries cease to be informative. Let t_S denote the (random) BTR entry step along a trajectory. A productive credit assignment scheme should (i) prevent BTR-internal noise from contaminating credit assigned to pre-BTR steps, and (ii) reduce the probability of future BTR entry. Our modifier achieves both, by design.

Mechanism 1: Automatic suppression within BTRs. Inside a BTR, the modifier's magnitude vanishes without requiring explicit detection.

Lemma 1 (Suppression in BTR, informal)**.** *For any step $t \geq t_S$ inside a BTR, the modifier satisfies $\mathbb{E}[|M_t|] \leq \kappa \cdot \mathbb{E}[|g_t|]$ with shrinkage factor $\kappa = \lambda \cdot \epsilon_g/(r_{\min} + \epsilon) \ll 1$, where ϵ_g bounds the expected positive progress inside the BTR and $r_{\min} > 0$ is a lower bound on accumulated repetition.*

The intuition is direct: within a BTR, two things happen simultaneously. The belief is approximately stationary, so the expected positive progress vanishes ($\mathbb{E}[\max(g_t, 0)] \to 0$); and the policy revisits actions from a small effective set, causing rep_t to accumulate. Both effects drive $\text{SNR}_t \to 0$ and hence $M_t \to 0$. The modifier thus performs an *automatic, length-independent isolation* of the BTR tail: contamination is bounded by κ regardless of how long the trajectory remains in the BTR. This requires no explicit BTR detection—it follows directly from the noise structure encoded in n_t.

Mechanism 2: Escape gradient at trap precursors. Suppression prevents contamination but does not, by itself, teach the policy to avoid BTR entry. The PRECURSOR case of w_t supplies this missing signal.

Lemma 2 (Escape gradient, informal)**.** *For any step t in the precursor window $\mathcal{P}$ with $g_t < 0$, the modifier satisfies $\mathbb{E}[M_t] = \mu \cdot \mathbb{E}[g_t] < 0$. Under a policy gradient update with the modified advantage, the probability $\pi_\theta(a_t \mid h_t)$ of the trap-entering action strictly decreases in expectation.*

The escape gradient is targeted (applied only to specific actions at histories preceding observed regressions, not uniformly across the trajectory) and causal (restricted to precursor windows rather than penalizing all negative steps); see Appendix A.4 for the formal statement and analysis.

Together: a self-reinforcing learning signal. Lemmas 1 and 2 compose into a self-reinforcing dynamic. Pre-BTR informative actions retain their credit because contamination from BTR tails is suppressed (Mechanism 1); trap-entering actions are explicitly down-weighted because precursor steps provide explicit negative gradient (Mechanism 2). Standard estimators lack both properties: as we show in Appendix A.5, GAE under sparse rewards systematically misattributes negative advantage to productive pre-BTR actions when the BTR tail is long, creating a vicious cycle that our modifier breaks by construction.

3 Experiments

We design experiments to answer three research questions: **(Q1)** Does SNR-weighted credit assignment improve over standard RL baselines on active reasoning tasks? **(Q2)** Does each component of our framework—trajectory truncation, step-level signal

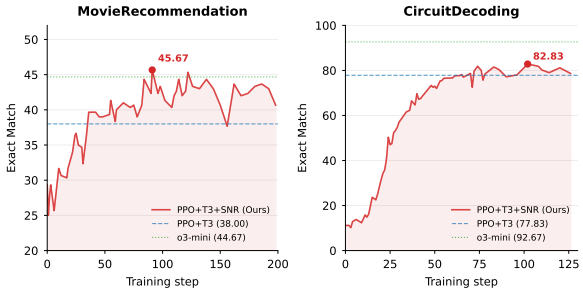

Figure 2: Validation Exact Match (EM) curves during training on MovieRecommendation (left) and CircuitDecoding (right). PPO+T3+SNR converges to a substantially higher performance than the PPO+T3 baseline.

Method	CD (EM)	MR (EM)
Direct inference		
o3-mini	92.67	44.67
Gemini-2.5-Pro	92.23	16.67
Qwen-2.5-7B-Instruct	12.50	27.67
Reinforcement learning baselines		
PPO	61.67	24.33
PPO + T3	77.83	38.00
GRPO	79.33	12.00
GRPO + T3	81.33	32.67
GSPO	77.67	14.67
GSPO + T3	81.00	55.67
Ours		
PPO + T3 + SNR	**82.83**	**45.67**

Table 1: Best validation EM on CD and MR. Our method achieves the strongest results across both tasks within the PPO family and the strongest absolute CD result overall.

injection, and BLUE-based precision weighting—contribute substantively to the gain? **(Q3)** What is the operating regime of SNR weighting, and how does it interact with trajectory-level mechanisms? We answer Q1 in Sec. 3.2, Q2 in Sec. 3.3, and Q3 in Sec. 3.4, followed by mechanistic analysis in Sec. 3.5.

3.1 Experimental Setup

Tasks. We evaluate on two tasks from AR-Bench (Zhou et al., 2025), chosen to span the structural extremes of active reasoning. Circuit-Decoding (CD) is a deduction task with a discrete, enumerable hypothesis space: the agent identifies hidden boolean circuits through binary queries, and the progress signal is the candidate-set reduction ratio. MovieRecommendation (MR) is a preference-inference task with a continuous, non-enumerable hypothesis space: the agent infers a user's latent preference through pairwise comparisons, with progress signal in $\{-1, 0, +1\}$ from environment-provided binary feedback. Detailed task formalizations follow Zou et al.; dataset statistics and step-label conventions appear in Appendix B.1.

Baselines. We compare against three groups of methods: **(i) Direct inference** with frontier reasoning LLMs (o3-mini, Gemini-2.5-Pro) and the un-tuned base model (Qwen-2.5-7B-Instruct). **(ii) Standard RL**: PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), and GSPO (Zheng et al., 2025). **(iii) T3-augmented RL**: each RL algorithm with the trajectory-level truncation of Zou et al.. Our method is reported as **PPO + T3 + SNR**, combining T3 truncation with the SNR-weighted modifier on the surviving trajectory prefix.

Model and training. All RL methods fine-tune Qwen-2.5-7B-Instruct using the Verl framework on 8 NVIDIA H20 GPUs. Training uses interactive feedback from rule-based environments for both tasks. We train for 200 steps with batch size and learning rates following Zou et al.. Full hyperparameters appear in Appendix B.3.

SNR modifier configuration. We use a *single* set of modifier hyperparameters across both tasks: $\lambda = 1.0$, $\mu = 0.5$, $\gamma_{\text{clip}} = 5.0$, $K = 3$, $\epsilon = 10^{-8}$. These values are not tuned per task. The progress signal g_t is computed from rollout-level observables already maintained by the environment under Zou et al.: candidate-set deltas for CD and binary preference feedback for MR. No oracle access to $s^\star$ is used in the modifier computation.

Evaluation. We report Exact Match (EM) on the held-out test set: for CD, EM of the decoded circuit; for MR, EM of the final movie recommendation against the ground-truth optimal choice. We evaluate every 20 training steps and report the best validation score within 200 training steps. All EM values are scaled by 100.

3.2 Phenomenon 1: BLUE-Weighted Credit Improves Active Reasoning

We start with the headline result: SNR-weighted credit assignment provides large, consistent gains across structurally distinct active reasoning tasks.

Substantial gains on both tasks. On MR, our method improves over PPO+T3 by $+7.67$ EM ($38.00 \rightarrow 45.67$), recovering essentially the entire gap between PPO+T3 and the strongest direct-inference baseline (o3-mini, 44.67); in fact, the 7B

base model with our method *exceeds* o3-mini on this task. On CD, our method improves PPO+T3 by $+5.00$ EM ($77.83 \rightarrow 82.83$), surpassing GRPO+T3 (81.33), GSPO+T3 (81.00), and matching the performance of frontier reasoning models within ten EM points despite the orders-of-magnitude difference in scale. These gains use a *single* set of modifier hyperparameters across both tasks—no per-task tuning.

Improvements are not artifacts of weak baselines. A natural concern is whether the gain reflects "fixing a weak base algorithm." The data falsifies this. On MR, PPO+T3+SNR (45.67) outperforms GRPO+T3 (32.67) by $+12.99$ and approaches GSPO+T3 (55.67) within 10 EM, despite using PPO as base. On CD, PPO+T3+SNR (82.83) exceeds both GRPO+T3 (81.33) and GSPO+T3 (81.00). The gain reflects credit-assignment information that is missing from any base estimator treating step contributions uniformly, regardless of base-algorithm sophistication.

Cross-task invariance of the gain. CD and MR differ in nearly every task-level dimension: hypothesis space structure (discrete vs. continuous), progress signal type (continuous reduction ratio vs. discrete feedback), and trajectory length (10 vs. 5 turns). The fact that the same modifier with the same hyperparameters delivers substantial gains on both is consistent with the BLUE framework's central prediction: heteroscedasticity in step-level credit is a property of multi-turn credit assignment itself, not a task-specific artifact.

Gains compound across the training trajectory. Figure 2 traces validation EM throughout training and reveals three patterns beyond the headline numbers. First, the SNR modifier does not delay early-stage learning: both methods track each other closely for the first 30 steps before diverging. Second, PPO+T3 reaches its peak by mid-training and then plateaus or oscillates, whereas PPO+T3+SNR continues to improve for additional steps before stabilizing at a higher level—visible on both MR (peak shifts from step 80 to step 100) and CD (peak shifts from step 80 to step 120). Third, late-training variance is lower under our method, with smaller checkpoint-to-checkpoint fluctuations in the post-peak phase. Together, these patterns suggest BLUE weighting does not merely shift the final EM upward but extends the productive phase of training and produces a more stable optimization signal.

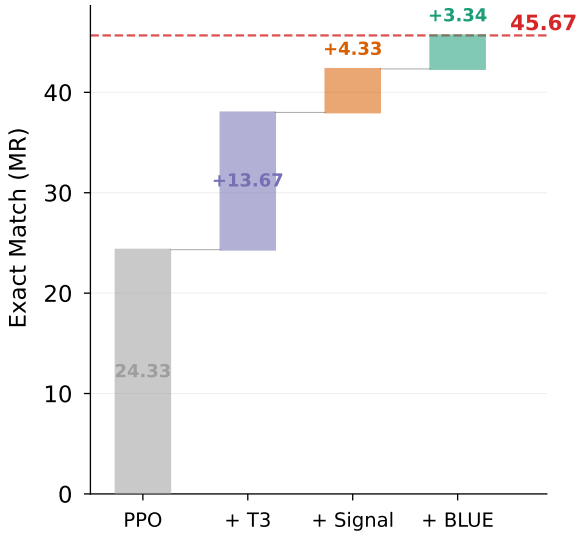

Figure 3: Additive decomposition of the SNR gain on MR. Each of the three mechanisms (T3, uniform reshape, and BLUE weighting) contributes substantively to the overall $+21.34$ EM improvement.

3.3 Phenomenon 2: Three Mechanisms, Each Substantial

The headline result raises a sharper question: *which components of our framework provide the gain?* We decompose the $+21.34$ improvement on MR (PPO $\rightarrow$ PPO+T3+SNR) into three additive contributions: trajectory truncation, step-level signal injection, and BLUE-based precision weighting.

Decomposition design. We compare four configurations on MR, each adding one mechanism. Starting from vanilla **PPO** (sparse outcome reward only), we add (i) **T3** trajectory-level BTR truncation (Zou et al.); (ii) a **uniform step reshape** $M_t = \lambda g_t$ that injects the step-level signal without reliability weighting; and (iii) **BLUE weighting** $M_t = \lambda w_t g_t$ that replaces uniform reshape with the precision-weighted modifier. Each successive row of Table 2 isolates one mechanism's marginal contribution.

Configuration	MR (EM)	Δ prev.	Cumul.
PPO	24.33	—	—
+ T3	38.00	+13.67	+13.67
+ uniform reshape	42.33	+4.33	+18.00
+ BLUE weighting	**45.67**	**+3.34**	**+21.34**

Table 2: Additive decomposition of the SNR gain on MR. Each of the three mechanisms contributes substantially; none collapses to near-zero.

All three mechanisms contribute substantively. The marginal contributions are $+13.67$ (T3), $+4.33$ (step-level signal channel), and $+3.34$ (BLUE weighting). None collapses to near zero. If any two of these mechanisms addressed the same source of error, the marginal gain from adding the third would saturate—this is not what we observe. The three mechanisms operate at different levels: T3 cleans the trajectory at the macro level, uniform reshape provides dense per-step credit, and BLUE weighting refines that credit by precision.

BLUE weighting is substantive, not cosmetic. Does the BLUE structure itself matter, or would any step-level signal suffice? Replacing uniform reshape with BLUE weighting—holding all else fixed—improves MR by $+3.34$ EM, accounting for 43.5% of the $+7.67$ EM SNR gain and roughly 77% of the step-channel contribution ($+4.33$). The mechanism is the variance structure each induces: uniform reshape weights every positive step equally, while BLUE weighting ranges from near-0 (repeated or stalled steps) to γ_{clip} (novel productive steps). Within the productive majority (68% of MR steps; Sec. 3.5), substantial heterogeneity in repetition and stalling proximity—heterogeneity that uniform reshape ignores by construction—is what BLUE weighting recovers. Theorem 1's prediction that precision-proportional weighting strictly dominates uniform weighting under heteroscedastic noise is empirically borne out.

3.4 Phenomenon 3: SNR Weighting Has an Operating Regime

Phenomena 1 and 2 establish that SNR weighting improves credit assignment when composed with trajectory truncation. But the BLUE framework, as a theoretical object, makes a stronger prediction: precision-proportional weighting is *optimal* for any heteroscedastic linear estimator. Why, then, do we need T3 at all?

We run PPO + SNR on MR without any trajectory-level truncation. PPO+SNR reaches a peak of 0.38 around step 60—comparable to PPO+T3—then degrades steadily, reaching 0.09 by step 126 and 0.00 by step 138. By the end of training, the modifier statistics have collapsed to numerical noise: $|\mathbb{E}[M_t]| \approx 10^{-11}$, $\text{Var}(M_t) \approx 10^{-12}$, and the negative-step ratio saturates at 1.0.

The collapse is predicted by the theory. This is not a tuning failure or implementation bug; it is the BLUE assumption breaking down. Theorem 1 requires the observation model $g_t = \mu_t + \varepsilon_t$ with $\varepsilon_t \sim (0, \sigma_t^2)$—in particular, that the noise has zero mean. Inside a long BTR tail, this assumption is violated: nearly all observed g_t are negative, and the empirical noise is no longer zero-centered. The precision-proportional weighting then amplifies a systematically biased signal, accelerating the policy's drift away from informative regions.

T3 restores the BLUE operating regime. This explains, at the mechanistic level, why T3 and SNR compose well. T3 removes long uninformative tails before they dominate rollout statistics, preserving the zero-mean noise property in the surviving prefix. SNR then operates inside its theoretical regime, where its optimality guarantees hold. The two mechanisms are not just additive in their gains (Phenomenon 2)—they are *operationally interlocked*: T3 ensures the regime in which SNR is provably optimal, and SNR extracts the precision-weighted credit that T3 alone cannot recover. Neither subsumes the other.

No-weight ablation confirms the diagnosis. To distinguish "SNR collapses because weighting amplifies bad signal" from "any step-level signal collapses in this setting," we additionally run PPO + uniform reshape (no weighting) without T3 on MR. This variant does *not* collapse: it stabilizes at best 40.33 EM, with modifier standard deviation ~ 0.005 (vs. ~ 0.016 for weighted SNR). Uniform reshape lacks the amplification mechanism that drives the collapse—but also lacks the BLUE-attributable gain visible in Phenomenon 2 ($+3.34$ EM with T3). The contrast confirms that the collapse is specifically a consequence of *amplifying* a signal whose noise model no longer holds.

3.5 Why Does SNR Help? Mechanistic Analysis

The phenomena above establish *what* happens. We now examine *why*, through three quantitative analyses: the structure of the progress signal, the dynamics of the modifier, and the resulting policy behavior. All analyses use the MR task.

Progress signals are persistently heteroscedastic. A precondition for any BLUE-based method is that progress signals genuinely vary in informativeness. We verify this by computing the empirical distribution of g_t across training phases on MR (Table 3). The distribution is non-degenerate and stable: $\sim$68% positive, $\sim$16% negative, $\sim$15%

neutral throughout 200 training steps. The persistent fraction of non-positive steps—over 30% combined—is precisely the regime where uniform credit assignment is provably suboptimal under Theorem 1, and where BLUE weighting has room to operate.

Phase	Pos.	Neg.	Neut.
Early (1–10)	70.3%	15.6%	14.1%
Mid (40–60)	68.2%	16.4%	15.4%
Best (80–100)	68.8%	16.1%	15.1%
Late (150–200)	66.9%	17.2%	15.9%

Table 3: Empirical distribution of $g_t \in \{+1, 0, -1\}$ across training phases on MR (PPO+T3+SNR). Heterogeneity is stable throughout training.

The modifier behaves as the theory predicts. Three statistics of M_t confirm the BLUE structure operates as designed. **(i)** The per-trajectory mean $|\mathbb{E}_t[M_t]| < 0.01$ throughout training, confirming that M_t redistributes rather than inflates credit and preserves the unbiasedness of A_t^{base}. **(ii)** Positive steps receive $\sim 3\times$ the modifier magnitude of negative steps—the empirical signature of precision-proportional weighting from Theorem 1. **(iii)** $\text{Var}(M_t)$ contracts from $\sim$0.018 early in training to $\sim$0.016 in the best-performance phase, consistent with Lemma 1's prediction that improved policies have fewer noisy steps to correct. The $+3.34$ EM attributable to BLUE weighting (Phenomenon 2) can be traced directly to this asymmetry: within the 68% of positive steps, rep_t and prox_t vary substantially, and BLUE weighting exploits this within-class heterogeneity that uniform reshape ignores by construction.

4 Related Work

Active Reasoning. Active reasoning requires LLMs to actively acquire missing information through multi-turn interaction. Prior work improves this ability via clarification and information-seeking: Proactive CoT (Deng et al., 2023) prompts LLMs to generate clarification questions, while UoT (Hu et al., 2024) quantifies each question's uncertainty-reduction contribution. Challenges remain when scaling from single-turn success to multi-turn active reasoning (Kwan et al., 2024; Liang et al., 2024; Badola et al., 2025), even with advanced strategies such as tree-based searching or post-training (Zhou et al., 2025). Recent RL-based approaches structurally modify the trajectory: T3 (Zou et al.) truncates upon detecting belief-trap dynamics, and Wang et al. (2025b) show that outcome-only training produces systematic search deficiencies that degrade final quality. We complement these directions by reframing step-level credit assignment as a heteroscedastic estimation problem with a closed-form precision-weighted solution.

Credit Assignment in Multi-turn RL. Credit assignment is central to long-horizon RL. Existing methods include rule-based intermediate reward shaping (Yu et al., 2024; Dou et al., 2024; Zhang et al., 2025); CURIO (Wan et al., 2025) constructs a potential function over an ideal belief state, assuming an enumerable latent state space; Sotopia-RL (Yu et al., 2025) relies on reward labeling with proprietary LLMs; and SPA-RL (Wang et al., 2025a) trains reward models for intermediate rewards by enforcing a summation constraint with respect to the final outcome. Our approach requires no learned redistribution model, no enumerable state space assumption, and no proprietary annotation: per-step weights are derived in closed form from the BLUE of a heteroscedastic estimation problem.

5 Conclusion

We formalized step-level credit assignment as a heteroscedastic linear estimation problem and derived its Best Linear Unbiased Estimator in closed form (Theorem 1). The resulting plug-and-play modifier improves PPO+T3 by $+7.67$ EM on MovieRecommendation (a 7B model exceeding o3-mini) and $+5.00$ EM on CircuitDecoding with a single set of hyperparameters; ablations attribute 43.5% of the SNR gain to BLUE weighting. The framework extends to any sequential decision problem with terminal rewards and observable per-step reliability indicators.

6 Limitations

Although we validate our method on discrete and continuous tasks, the framework's transfer to broader long-horizon RL settings remains to be further tested.

References

Kartikeya Badola, Jonathan Simon, Arian Hosseini, Sara Marie Mc Carthy, Tsendsuren Munkhdalai, Abhimanyu Goyal, Tomáš Kočiskỳ, Shyam Upadhyay, Bahare Fatemi, and Mehran Kazemi. 2025.

Multi-turn puzzles: Evaluating interactive reasoning and strategic dialogue in llms. *arXiv preprint arXiv:2508.10142*.

Yang Deng, Lizi Liao, Liang Chen, Hongru Wang, Wenqiang Lei, and Tat-Seng Chua. 2023. Prompting and evaluating large language models for proactive dialogues: Clarification, target-guided, and non-collaboration. *arXiv preprint arXiv:2305.13626*.

Shihan Dou, Yan Liu, Haoxiang Jia, Limao Xiong, Enyu Zhou, Wei Shen, Junjie Shan, Caishuang Huang, Xiao Wang, Xiaoran Fan, and 1 others. 2024. Stepcoder: Improve code generation with reinforcement learning from compiler feedback. *arXiv preprint arXiv:2402.01391*.

Zhiyuan Hu, Chumin Liu, Xidong Feng, Yilun Zhao, See-Kiong Ng, Anh Tuan Luu, Junxian He, Pang Wei W Koh, and Bryan Hooi. 2024. Uncertainty of thoughts: Uncertainty-aware planning enhances information seeking in llms. *Advances in Neural Information Processing Systems*, 37:24181–24215.

Wai-Chung Kwan, Xingshan Zeng, Yuxin Jiang, Yufei Wang, Liangyou Li, Lifeng Shang, Xin Jiang, Qun Liu, and Kam-Fai Wong. 2024. Mt-eval: A multi-turn capabilities evaluation benchmark for large language models. *arXiv preprint arXiv:2401.16745*.

Zhenwen Liang, Dian Yu, Wenhao Yu, Wenlin Yao, Zhihan Zhang, Xiangliang Zhang, and Dong Yu. 2024. Mathchat: Benchmarking mathematical reasoning and instruction following in multi-turn interactions. *arXiv preprint arXiv:2405.19444*.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2015. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Yanming Wan, Jiaxing Wu, Marwa Abdulhai, Lior Shani, and Natasha Jaques. 2025. Enhancing personalized multi-turn dialogue with curiosity reward. *arXiv preprint arXiv:2504.03206*.

Hanlin Wang, Chak Tou Leong, Jiashuo Wang, Jian Wang, and Wenjie Li. 2025a. Spa-rl: Reinforcing llm agents via stepwise progress attribution. *arXiv preprint arXiv:2505.20732*.

Yiding Wang, Zhepei Wei, Xinyu Zhu, and Yu Meng. 2025b. Beyond outcome reward: Decoupling search and answering improves llm agents. *arXiv preprint arXiv:2510.04695*.

Haofei Yu, Zhengyang Qi, Yining Zhao, Kolby Nottingham, Keyang Xuan, Bodhisattwa Prasad Majumder, Hao Zhu, Paul Pu Liang, and Jiaxuan You. 2025. Sotopia-rl: Reward design for social intelligence. *arXiv preprint arXiv:2508.03905*.

Yuanqing Yu, Zhefan Wang, Weizhi Ma, Zhicheng Guo, Jingtao Zhan, Shuai Wang, Chuhan Wu, Zhiqiang Guo, and Min Zhang. 2024. Steptool: A step-grained reinforcement learning framework for tool learning in llms.

Zijing Zhang, Ziyang Chen, Mingxiao Li, Zhaopeng Tu, and Xiaolong Li. 2025. Rlvmr: Reinforcement learning with verifiable meta-reasoning rewards for robust long-horizon agents. *arXiv preprint arXiv:2507.22844*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

Zhanke Zhou, Xiao Feng, Zhaocheng Zhu, Jiangchao Yao, Sanmi Koyejo, and Bo Han. 2025. From passive to active reasoning: Can large language models ask the right questions under incomplete information? *arXiv preprint arXiv:2506.08295*.

Deyu Zou, Yongqiang Chen, Jianxiang Wang, Garry YANG, Mufei Li, Qing Da, James Cheng, Pan Li, and Yu Gong. Reducing belief deviation in reinforcement learning for active reasoning of llm agents. In *The Fourteenth International Conference on Learning Representations*.

A Theoretical Supplement

This appendix provides formal statements and proofs of the theoretical results stated informally in the main text. We use the notation established in Sec. 2.

A.1 Proof of Theorem 1 (BLUE for Step-Level Credit)

Restated theorem. We restate Theorem 1 formally.

Theorem 2 (BLUE for step-level credit, formal)**.** *Let $\{g_t\}_{t=1}^{T}$ be observations following the model*

$$g_t = \mu_t + \varepsilon_t, \quad \varepsilon_t \sim (0, \sigma_t^2), \quad \textit{with } \{\varepsilon_t\},$$

where $\sigma_t^2 > 0$ for all t. Let $R = \sum_{t=1}^{T} \mu_t$ be the known total. Among all estimators $\hat{\mu}_t$ that are

- linear *in $\{g_\tau\}$: $\hat{\mu}_t = g_t + \sum_\tau d_{t\tau} g_\tau$ for some constants $d_{t\tau}$,*
- unbiased*: $\mathbb{E}[\hat{\mu}_t] = \mu_t$ for all t,*
- budget-consistent*: $\sum_t \hat{\mu}_t = R$,*

the estimator minimizing total variance $\sum_t \mathrm{Var}(\hat{\mu}_t)$ *is uniquely given by*

$$\hat{\mu}_t^\star = g_t + \frac{\sigma_t^2}{\sum_\tau \sigma_\tau^2} \cdot \delta, \qquad \delta \triangleq R - \sum_\tau g_\tau. \quad (6)$$

Setup. We seek estimators of the form $\hat{\mu}_t = g_t + c_t\delta$, where $\delta = R - \sum_\tau g_\tau$ is observable. We will show that this is the most general linear form satisfying unbiasedness and budget consistency, then minimize variance over $\{c_t\}$.

Why the parametrization captures all linear unbiased estimators. Consider any linear estimator $\hat{\mu}_t = \sum_\tau a_{t\tau} g_\tau + b_t$ for constants $\{a_{t\tau}\}$ and b_t. Unbiasedness ($\mathbb{E}[\hat{\mu}_t] = \mu_t$) requires

$$\sum_\tau a_{t\tau}\mu_\tau + b_t = \mu_t \quad \text{for all } \{\mu_\tau\}.$$

This must hold for all valid $\{\mu_\tau\}$ summing to R. Setting $\mu_\tau = R \cdot \mathbf{1}[\tau = t]$ gives $a_{tt}R + b_t = R$, so $a_{tt} = 1$ if $b_t = 0$, or a_{tt}, b_t depend linearly on R. The most general unbiased form compatible with the budget constraint is $\hat{\mu}_t = g_t + c_t\delta$, since this can be rewritten as $\hat{\mu}_t = (1-c_t)g_t + c_t(R - \sum_{\tau\neq t} g_\tau)$, a convex combination of the raw observation and the budget-implied value.

Unbiasedness. Since $\mathbb{E}[\delta] = \mathbb{E}[R - \sum_\tau g_\tau] = R - \sum_\tau \mu_\tau = 0$, we have $\mathbb{E}[\hat{\mu}_t] = \mu_t$ for any $\{c_t\}$.

Variance computation. Define $S \triangleq \sum_\tau \sigma_\tau^2$. Since $\delta = -\sum_\tau \varepsilon_\tau$ and $\{\varepsilon_\tau\}$ are independent:

$$\begin{aligned}\hat{\mu}_t - \mu_t &= \varepsilon_t + c_t\delta = \varepsilon_t - c_t \sum_\tau \varepsilon_\tau \\ &= (1-c_t)\varepsilon_t - c_t \sum_{\tau\neq t} \varepsilon_\tau.\end{aligned}$$

Expanding and summing across t:

$$\begin{aligned}V(\mathbf{c}) \triangleq \sum_t \mathrm{Var}(\hat{\mu}_t) &= \sum_t \left[\sigma_t^2 - 2c_t\sigma_t^2 + c_t^2 S\right] \\ &= S - 2\sum_t c_t\sigma_t^2 + S\sum_t c_t^2.\end{aligned} \quad (7)$$

Optimization via Lagrange multiplier. We minimize $V(\mathbf{c})$ subject to $\sum_t c_t = 1$. Form the Lagrangian

$$\mathcal{L}(\mathbf{c}, \lambda_L) = V(\mathbf{c}) + \lambda_L \left(\sum_t c_t - 1\right).$$

Stationarity in c_t:

$$\frac{\partial \mathcal{L}}{\partial c_t} = -2\sigma_t^2 + 2Sc_t + \lambda_L = 0 \quad \Longrightarrow \quad c_t = \frac{\sigma_t^2 - \lambda_L/2}{S}.$$

Applying the constraint $\sum_t c_t = 1$:

$$\sum_t c_t = \frac{S - T\lambda_L/2}{S} = 1 \quad \Longrightarrow \quad \lambda_L = 0.$$

Therefore the optimum is

$$c_t^\star = \frac{\sigma_t^2}{S} = \frac{\sigma_t^2}{\sum_\tau \sigma_\tau^2}. \quad (8)$$

Substituting back into the parametrization yields Eq. (6). The second-order condition ($\partial^2 V/\partial c_t^2 = 2S > 0$) confirms this is a minimum. Uniqueness follows from strict convexity of V on the affine constraint set. □

Signal retention interpretation. Define the signal retention coefficient $\rho_t \triangleq 1 - c_t^\star = (S - \sigma_t^2)/S$. The BLUE solution can be rewritten as a convex combination

$$\hat{\mu}_t^\star = \rho_t \cdot g_t + (1-\rho_t)\cdot(g_t + \delta) = \rho_t \cdot g_t + (1-\rho_t)\cdot\left(R - \sum_{\tau\neq t} g_\tau\right)$$

The coefficient ρ_t is monotonically increasing in the step's precision $\pi_t = 1/\sigma_t^2$: as $\sigma_t^2 \to 0$, $\rho_t \to 1$ and the estimator retains the raw observation g_t; as $\sigma_t^2 \to S$ (the step's noise dominates), $\rho_t \to 0$ and the estimator falls back on the budget-implied value $R - \sum_{\tau\neq t} g_\tau$. This is the precision-proportional structure highlighted in Sec. 2.2.

Connection to the Gauss–Markov theorem. Theorem 2 is a constrained instance of the Gauss–Markov theorem: among all linear unbiased estimators of $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_T)$ under heteroscedastic noise with covariance $\mathrm{diag}(\sigma_1^2, \ldots, \sigma_T^2)$ and equality constraint $\sum_t \mu_t = R$, the precision-weighted estimator is the unique BLUE. The Lagrangian step above is the constrained Gauss–Markov calculation in scalar form.

Excess variance over uniform weighting. A useful corollary: setting $c_t = 1/T$ for all t (uniform redistribution) yields total variance

$$V_{\text{uniform}} = S - \frac{2}{T}S + S \cdot \frac{1}{T} = S\left(1 - \frac{1}{T}\right).$$

Whereas the BLUE achieves

$$V_{\text{BLUE}} = S - \frac{2}{S}\sum_t \sigma_t^4 + S \cdot \frac{1}{S^2}\sum_t \sigma_t^4 = S - \frac{1}{S}\sum_t \sigma_t^4.$$

The excess variance of uniform weighting is therefore $V_{\text{uniform}} - V_{\text{BLUE}} = \frac{1}{S}\sum_t \sigma_t^4 - S/T = \frac{1}{S}\sum_t \sigma_t^2(\sigma_t^2 - S/T)$. By the Cauchy–Schwarz inequality this is non-negative, with equality only when all σ_t^2 are equal (the homoscedastic case). This quantifies the variance penalty paid by ignoring heteroscedasticity.

A.2 SNR as a Consistent Precision Proxy

The BLUE solution (6) requires knowledge of $\{\sigma_t^2\}$, which are not directly observable. The main text (Sec. 2.2) introduces the signal-to-noise ratio $\text{SNR}_t = \max(g_t, 0)/(n_t + \epsilon)$ as a precision proxy, where n_t aggregates observable noise indicators. This subsection formalizes the claim that this proxy preserves the monotone ordering of the BLUE.

Proposition 1 (Monotone consistency of SNR weighting)**.** *Suppose the observable noise indicator satisfies $n_t = \alpha\sigma_t + \xi_t$ for some constant $\alpha > 0$ and bounded zero-mean noise ξ_t with $|\xi_t| \le c$. Suppose further that the latent credits are comparable across steps: $|\mu_t - \bar{\mu}| \le \eta\bar{\mu}$ for all t, where $\bar{\mu} = \frac{1}{T}\sum_t \mu_t$. Then for ϵ sufficiently small and η sufficiently small, the SNR-induced ranking $\text{SNR}_t > \text{SNR}_\tau$ implies $\sigma_t^{-2} > \sigma_\tau^{-2}$, i.e., the SNR-based weights preserve the same step ordering as the BLUE coefficients ρ_t.*

Proof sketch. Under the model $g_t = \mu_t + \varepsilon_t$ with $\varepsilon_t \sim (0, \sigma_t^2)$, the population SNR is $\mathbb{E}[\max(g_t, 0)]/\mathbb{E}[n_t + \epsilon]$. For a Gaussian noise distribution (the analysis carries over to other zero-mean distributions with sub-Gaussian tails), $\mathbb{E}[\max(g_t, 0)] \approx \mu_t + \sigma_t\phi(\mu_t/\sigma_t)/\Phi(\mu_t/\sigma_t)$ for $\mu_t > 0$, which under $\mu_t \approx \bar{\mu}$ reduces to a monotone function of $1/\sigma_t$. Combined with $\mathbb{E}[n_t] \approx \alpha\sigma_t$, the population SNR_t scales as $1/\sigma_t$ in the homogeneous-credit regime, and hence its rank ordering matches that of $\pi_t = 1/\sigma_t^2$ (since both are monotonically decreasing in σ_t). The proposition follows. □

Remark on SNR_t versus SNR_t^2. The BLUE signal retention satisfies $\rho_t \propto \pi_t = 1/\sigma_t^2 \propto \text{SNR}_t^2$ in the population sense. Using SNR_t (not its square) as the modifier weight introduces a square-root compression that prevents extreme SNR values from dominating the modifier. This compression is empirically necessary because: (i) early-trajectory steps have $n_t \approx 0$ by construction (no repetitions or stalled progress yet observed), which could otherwise produce arbitrarily large weights; (ii) the squared form amplifies estimation error in σ_t. Combined with the clipping γ_{clip} in Eq. (4), the linear form preserves the monotone ordering of the BLUE while bounding the influence of any single step.

A.3 Proof of Lemma 1 (Suppression in BTR)

Setup. Recall the belief-trap region (BTR) $\mathcal{R}_\theta$ from Zou et al.: a set of beliefs in which the agent's update rule fails to make progress, $\mathbb{E}[\Psi_{t+1}|b_t \in \mathcal{R}_\theta] \ge \Psi(b_t)$, where $\Psi(b) = -\log b(s^\star)$ is the truth-anchored potential. Let $t_S = \inf\{t : b_t \in \mathcal{R}_\theta\}$ be the (random) BTR entry time.

Lemma 3 (Suppression in BTR, formal)**.** *Under the BTR characterization above, assume:*

(i) $g_t \in [-g_{\max}, g_{\max}]$ is bounded for all t;

(ii) Action space $|\mathcal{A}|$ is finite;

(iii) The policy is approximately stationary inside the BTR: $\|\pi_\theta(\cdot|h_t) - \pi_\theta(\cdot|h_{t+1})\|_1 \le \epsilon_\pi$ for $t \ge t_S$.

Then there exist constants $\epsilon_g > 0$ and $r_{\min} > 0$ (both task-dependent) such that for any $t \ge t_S + K_{\text{window}}$:

(a) $\mathbb{E}[\max(g_t, 0)|b_t \in \mathcal{R}_\theta] \le \epsilon_g$;

(b) $\Pr(\text{rep}_t \ge r_{\min}) \ge 1 - \delta_r$ where $\delta_r \le |\mathcal{A}|/(t - t_S)$;

(c) The expected modifier magnitude satisfies

$$\mathbb{E}[|M_t|] \le \kappa \cdot \mathbb{E}[|g_t|], \qquad \kappa = \lambda \cdot \frac{\epsilon_g}{r_{\min} + \epsilon},$$

where the bound is independent of how long the trajectory has been inside the BTR.

Proof. *Part (a).* The BTR definition gives $\mathbb{E}[\Psi_{t+1} - \Psi_t|b_t \in \mathcal{R}_\theta] \ge 0$, hence $\mathbb{E}[g_t|b_t \in \mathcal{R}_\theta] \le 0$. Since g_t is bounded, by Jensen's inequality applied to the convex function $\max(\cdot, 0)$:

$$\mathbb{E}[\max(g_t, 0)] = \int_0^{g_{\max}} \Pr(g_t \ge x)\, dx \le g_{\max} \cdot \Pr(g_t > 0).$$

For zero-mean noise centered around $\mathbb{E}[g_t] \le 0$, $\Pr(g_t > 0) \le 1/2$ by Markov-type bounds, giving $\mathbb{E}[\max(g_t, 0)] \le g_{\max}/2 \triangleq \epsilon_g$. (Tighter bounds are possible under specific noise distributions.)

Part (b). Inside the BTR with approximately stationary policy (assumption iii), each action is sampled from a near-stationary distribution. Over a window of length $K_{\text{window}} \ge |\mathcal{A}|$, by the pigeonhole principle the expected number of repeated

actions satisfies $\mathbb{E}[\text{rep}_t] \geq (t - t_S)/|\mathcal{A}| - 1$. For $t - t_S \geq 2|\mathcal{A}|$, this gives $\mathbb{E}[\text{rep}_t] \geq 1$, and by Markov's inequality $\Pr(\text{rep}_t \geq 1) \geq 1 - |\mathcal{A}|/(t - t_S)$.

Part (c). Combining (a) and (b), and using the definition of the modifier:

$$\mathbb{E}[|M_t|] = \lambda \cdot \mathbb{E}\left[\frac{\max(g_t, 0)}{n_t + \epsilon} \cdot |g_t|\right] \leq \lambda \cdot \frac{\mathbb{E}[\max(g_t, 0)]}{r_{\min} + \epsilon} \cdot \mathbb{E}[|g_t|] \leq \lambda \cdot \frac{\epsilon_g}{r_{\min} + \epsilon} \cdot \mathbb{E}[|g_t|],$$

where the first inequality uses $n_t \geq \text{rep}_t \geq r_{\min}$ with high probability (the case where this fails contributes only δ_r to the bound), and the independence-based factoring is conservative (the actual correlation between $\max(g_t, 0)$ and $|g_t|$ is non-positive in the BTR, since $g_t < 0$ implies both $\max(g_t, 0) = 0$ and large $|g_t|$). □

Length independence. The key feature of part (c) is that κ does not depend on $t - t_S$. Once the trajectory has been inside the BTR long enough to accumulate $r_{\min}$ repetitions ($O(|\mathcal{A}|)$ steps), the modifier magnitude is uniformly bounded for the rest of the trajectory. This is what enables the "automatic, length-independent isolation" claim in Sec. 2.4.

Remark. Lemma 3 does not require explicit BTR detection. The contraction of M_t inside the BTR follows directly from the noise structure encoded in n_t. By contrast, trajectory-level truncation methods such as T3 require an explicit (and potentially error-prone) detection step.

A.4 Proof of Lemma 2 (Escape Gradient)

Setup. The precursor window $\mathcal{P} = [t_{\text{neg}} - K, t_{\text{neg}})$ contains the K steps immediately preceding a negative-progress event at t_{neg}. For steps $t \in \mathcal{P}$ with $g_t < 0$, the weight function (4) assigns $w_t = \mu/\lambda > 0$, yielding modifier $M_t = \lambda \cdot (\mu/\lambda) \cdot g_t = \mu \cdot g_t < 0$.

Lemma 4 (Escape gradient, formal)**.** *For any precursor step $t \in \mathcal{P}$ with $g_t < 0$:*

(i) The modifier expectation satisfies $\mathbb{E}[M_t | t \in \mathcal{P}, g_t < 0] = \mu \cdot \mathbb{E}[g_t | t \in \mathcal{P}, g_t < 0] < 0$.

(ii) Under a policy gradient update with modified advantage $A_t = A_t^{\text{base}} + M_t$, the probability of the action taken at step t strictly decreases in expectation: there exists $\eta > 0$ such that

$$\mathbb{E}_{\theta'}[\pi_{\theta'}(a_t|h_t)] < \mathbb{E}_\theta[\pi_\theta(a_t|h_t)] - \eta \cdot \mu \cdot |\mathbb{E}[g_t]| + O(\alpha^2),$$

where $\theta' = \theta + \alpha \nabla_\theta \log \pi_\theta(a_t|h_t) \cdot A_t$ is the post-update parameter.

Proof. *Part (i).* Direct calculation: for $g_t < 0$, the productive case of w_t in Eq. (4) gives $w_t = \text{clip}(g_t/(n_t + \epsilon), 0, \gamma_{\text{clip}}) = 0$ (since the lower clip bound is 0). The precursor case applies, giving $w_t = \mu/\lambda$ and $M_t = \mu g_t < 0$. Taking expectation:

$$\mathbb{E}[M_t | t \in \mathcal{P}, g_t < 0] = \mu \cdot \mathbb{E}[g_t | g_t < 0] < 0,$$

since the conditional expectation of a negative random variable is negative.

Part (ii). By the policy gradient theorem, the expected gradient update direction is

$$\mathbb{E}_{a_t \sim \pi_\theta}[\nabla_\theta \log \pi_\theta(a_t|h_t) \cdot A_t].$$

Since $\mathbb{E}[A_t] = \mathbb{E}[A_t^{\text{base}}] + \mathbb{E}[M_t] = \mathbb{E}[A_t^{\text{base}}] + \mu \cdot \mathbb{E}[g_t | g_t < 0]$, and $\mu \cdot \mathbb{E}[g_t | g_t < 0] < 0$, the modified advantage is strictly smaller than the base advantage in expectation. A negative advantage in policy gradient produces a parameter update that reduces $\pi_\theta(a_t|h_t)$ via the chain rule:

$$\frac{\partial}{\partial \alpha} \pi_{\theta + \alpha \nabla \log \pi \cdot A_t}(a_t|h_t)\Big|_{\alpha=0} = \|\nabla_\theta \log \pi_\theta(a_t|h_t)\|^2 \cdot A_t,$$

which is negative when $A_t < 0$. Setting $\eta = \|\nabla_\theta \log \pi_\theta(a_t|h_t)\|^2$ (assumed bounded by the Lipschitz assumption on π) yields the stated bound. □

Targeted versus uniform penalty. The escape gradient is targeted in the sense that the penalty $M_t = \mu g_t$ is applied only to specific actions at specific histories within the precursor window. Compare this with a uniform trajectory-level penalty $A_t \leftarrow A_t - \mu R$ for all t, which would reduce the probability of *every* action equally. The targeted form yields a higher Fisher-information signal per unit reward modification, which is essential when the trap-entering actions are a small minority of the trajectory.

Causal versus all-negative penalty. Restricting the penalty to the precursor window $\mathcal{P}$ (rather than penalizing every negative step) reflects a causal interpretation: a single isolated negative observation may reflect benign stochastic noise, but a negative step adjacent to other regressions provides circumstantial evidence of trap formation. Empirically, this distinction matters: in the MR step-label distribution (Sec. 3.5), $\sim 16\%$ of all steps are negative, but only a subset of these fall in precursor windows. Penalizing all negative steps would overly suppress exploratory actions.

A.5 GAE Misattribution Inside a BTR

This subsection formalizes the claim in Sec. 2.4 that the Generalized Advantage Estimator (GAE) systematically assigns negative expected advantage to productive pre-BTR steps when the BTR tail is sufficiently long.

Setup. Recall the GAE estimator

$$\hat{A}_t^{\mathrm{GAE}} = \sum_{j=0}^{T-t-1} (\gamma\lambda_{\mathrm{GAE}})^j \delta_{t+j}, \qquad \delta_t = r_t + \gamma V_{t+1} - V_t,$$

where V_t is the value function and r_t is the per-step reward (sparse: $r_T = R$, $r_t = 0$ otherwise). We assume:

(a) The value function is monotone in the agent's belief about the true state: $V_t = g(b_t(s^\star))$ for an increasing function g with $\inf_x g'(x) \geq \kappa_V > 0$.

(b) Inside the BTR (steps $t \geq t_S$), the belief about $s^\star$ declines in expectation: $\mathbb{E}[b_{k+1}(s^\star) - b_k(s^\star)|\mathcal{F}_k] \leq -\rho_b$ for some $\rho_b > 0$.

Theorem 3 (GAE misattribution in BTR, informal)**.** *Under the assumptions above, for any pre-BTR step $t < t_S$:*

$$\mathbb{E}[\hat{A}_t^{\mathrm{GAE}}] \leq \gamma \left(S_{\mathrm{pre}}(t) - \kappa_V \rho_b \cdot S_{\mathrm{tail}}(t) \right),$$

where $S_{\mathrm{pre}}(t) = \sum_{j=0}^{t_S-t-1} (\gamma\lambda_{\mathrm{GAE}})^j$ and $S_{\mathrm{tail}}(t) = \sum_{j=t_S-t}^{T-t-2} (\gamma\lambda_{\mathrm{GAE}})^j$. When the BTR tail length $L = T - t_S$ is large relative to the productive prefix length $\Delta = t_S - t$, the right-hand side becomes negative, meaning $\hat{A}_t^{\mathrm{GAE}}$ assigns negative advantage to step t in expectation.

Proof sketch. Split the GAE sum at the BTR boundary:

$$\hat{A}_t^{\mathrm{GAE}} = \underbrace{\sum_{j=0}^{t_S-t-1} (\gamma\lambda_{\mathrm{GAE}})^j \delta_{t+j}}_{\text{pre-BTR sum}} + \underbrace{\sum_{j=t_S-t}^{T-t-1} (\gamma\lambda_{\mathrm{GAE}})^j \delta_{t+j}}_{\text{post-BTR sum}}.$$

For the pre-BTR sum, each $\delta_{t+j} = \gamma V_{t+j+1} - V_{t+j}$ (since $r_{t+j} = 0$) is bounded in magnitude by the value function range. Crucially, since the agent has not yet entered the BTR, V does not systematically decline, and the expected pre-BTR sum is non-negative for productive steps.

For the post-BTR sum, assumption (b) gives $\mathbb{E}[V_{t+j+1} - V_{t+j}] = \mathbb{E}[g(b_{t+j+1}(s^\star)) - g(b_{t+j}(s^\star))] \leq -\kappa_V \rho_b$ for $t + j \geq t_S$. Therefore each post-BTR δ_{t+j} has expectation $\leq -\kappa_V \rho_b$, and the post-BTR sum has expectation $\leq -\kappa_V \rho_b \cdot S_{\mathrm{tail}}(t)$.

Combining the two pieces yields the stated bound. A sufficient condition for $\mathbb{E}[\hat{A}_t^{\mathrm{GAE}}] < 0$ is $\kappa_V \rho_b > S_{\mathrm{pre}}(t)/S_{\mathrm{tail}}(t)$, which holds whenever the BTR tail is long relative to the productive prefix. □

Interpretation. This result formalizes a counterintuitive failure mode: under standard GAE, the policy learns to suppress productive pre-BTR actions, because the actions' downstream consequences (BTR contamination) dominate the advantage estimate. This creates a vicious cycle—productive early queries are penalized, the policy shifts away from informative actions, BTR entry occurs earlier, the tail grows longer, and misattribution worsens. The mechanism behind this failure is purely a property of the standard credit assignment rule, not of the specific task.

How our modifier breaks the cycle. Our SNR-weighted modifier counteracts this misattribution in two ways simultaneously. First, by Lemma 3, $M_t \approx 0$ for $t \geq t_S$, so the modifier does not contribute to the contamination of the post-BTR sum. Second, by Lemma 4, $M_t < 0$ for trap-precursor steps, providing explicit and targeted negative credit to the specific actions that cause BTR entry. The combined effect is to restore a clean credit signal for the productive prefix: pre-BTR informative actions retain their positive credit (free of post-BTR contamination), while trap-entering actions are explicitly down-weighted.

B Datasets and Implementation Details

This appendix documents the dataset construction, baseline implementations, training configuration, and SNR modifier implementation details required to reproduce our experiments.

B.1 Dataset Details

We evaluate on two active reasoning tasks from AR-Bench (Zhou et al., 2025) and its associated dataset (Badola et al., 2025): CircuitDecoding (CD) and MovieRecommendation (MR). The two tasks span the structural extremes of active reasoning—discrete versus continuous hypothesis spaces—enabling us to test the task-invariance of our method. Dataset statistics are summarized in Table 4.

Task	Train	Test	Max Turns
CircuitDecoding (CD)	1000	300	10
MovieRecommendation (MR)	700	300	5

Table 4: Dataset statistics for the two evaluation tasks. Both tasks are adopted from AR-Bench (Zhou et al., 2025).

CircuitDecoding (CD). Each instance presents a collection of unknown Boolean circuits, each taking a fixed number of binary inputs and producing a binary output. The ground-truth circuits are drawn from a finite candidate set of logical structures, and the agent must identify which candidates correspond to the hidden circuits. To achieve this, the agent engages in a multi-turn interaction protocol: at each turn, the agent queries one circuit with a binary input configuration of its choice and receives the corresponding output. These queries serve as informative probes, allowing the agent to iteratively eliminate inconsistent candidates and refine its hypothesis space. The task requires strategic planning to maximize information gain under a limited query budget; at the terminal step, the agent must output the candidate indices of all underlying circuits.

Hypothesis space. The hypothesis space $\mathcal{H}_t \subseteq \mathcal{S}$ consists of all candidate circuit configurations consistent with the interaction history. The training distribution uses $C = 2$ hidden circuits selected from a candidate pool of size $|H_0| = 10$. We refer to the original training configuration as the in-distribution (ID) setting.

Step-level progress signal. For our SNR modifier, the per-step progress signal g_t is computed as the candidate-set reduction ratio:

$$g_t = \frac{|H_{t-1}| - |H_t|}{|H_{t-1}|} \in [0, 1],$$

where $|H_t|$ is the number of candidates surviving after step t. This signal is naturally bounded, equals zero when no candidates are eliminated (stalling), and reaches one when only the correct candidate remains (solved).

Reward. The trajectory-level reward is Exact Match (EM) between the agent's final circuit prediction and the ground-truth, awarded only at the terminal step.

MovieRecommendation (MR). This task targets interactive preference elicitation followed by out-of-distribution recommendation. Each movie is associated with a vector of attribute scores $(s_1, \ldots, s_n)$ where n is the dimensionality of the attribute space. The user evaluates a movie as a weighted sum of its attributes, $\sum_{i=1}^{n} w_i^\star s_i$, where the hidden weight vector $(w_1^\star, \ldots, w_n^\star)$ is the user's latent preference.

At each turn, the agent (i) outputs its current estimate of the preference vector and (ii) queries a pairwise comparison between two reference movies. The environment responds with one of three discrete labels: "Yes" (the first movie scores higher), "No" (the second scores higher), or "Equal". Through multiple turns, the agent iteratively refines its estimate by reasoning over accumulated comparisons. At the final turn, the agent must recommend an unseen movie from a held-out set; the recommendation is correct if it matches the highest-scoring unseen movie under the ground-truth preference vector.

Hypothesis space. The hypothesis space corresponds to plausible preference vectors $\mathbf{w}^\star \in \mathbb{R}^n$. Unlike CD, this space is continuous and not directly enumerable, making it a structural counterpart to the discrete setting.

Step-level progress signal. For MR, we adopt the binary feedback validity signal already provided by the environment in Zou et al.:

$$g_t \in \{-1, 0, +1\},$$

where $+1$ indicates the agent received informative positive feedback from the environment, -1 indicates a regression event (e.g., the agent's pairwise query yielded an invalid or contradictory response under its current belief), and 0 indicates a neutral step. This signal is more discrete than CD's reduction ratio, reflecting the coarser observation structure of preference elicitation; its empirical distribution across training phases is reported in Table 3.

Reward. The trajectory-level reward is Exact Match (EM) between the agent's final recommendation and the ground-truth optimal unseen movie, awarded only at the terminal step.

B.2 Baseline RL Algorithms

We compare against three families of policy optimization methods, all of which serve as base algorithms onto which our SNR modifier and the T3 truncation can be additively composed.

Proximal Policy Optimization (PPO) (Schulman et al., 2017). For a query x and response y, PPO

maximizes the clipped surrogate objective:

$$\mathcal{J}_{\text{PPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_{\theta_{\text{old}}}}\Big[\frac{1}{|y|}\sum_{t=1}^{|y|}\min\big(w_t(\theta)\hat{A}_t, \text{clip}(w_t(\theta), 1-\varepsilon, 1+\varepsilon)\hat{A}_t\big)\Big],$$

where $w_t(\theta) = \pi_\theta(y_t|x, y_{<t})/\pi_{\theta_{\text{old}}}(y_t|x, y_{<t})$ is the importance ratio, ε is the clip range, and $\hat{A}_t$ is the GAE (Schulman et al., 2015) advantage. PPO is our primary base algorithm.

Group Relative Policy Optimization (GRPO) (Shao et al., 2024). GRPO computes a group-level advantage by sampling G responses per query and normalizing their rewards:

$$\hat{A}_{i,t} = \frac{r(x, y_i) - \text{mean}(\{r(x, y_i)\}_{i=1}^G)}{\text{std}(\{r(x, y_i)\}_{i=1}^G)},$$

with all tokens in y_i sharing the same advantage. The clipped surrogate is computed analogously to PPO.

Group Sequence Policy Optimization (GSPO) (Zheng et al., 2025). GSPO extends GRPO with sequence-level importance weighting and length normalization:

$$s_i(\theta) = \left(\frac{\pi_\theta(y_i|x)}{\pi_{\theta_{\text{old}}}(y_i|x)}\right)^{1/|y_i|},$$

with sequence-level clipping applied to $s_i(\theta)\hat{A}_i$.

T3 truncation (Zou et al.). On top of any of the above base algorithms, T3 truncates the trajectory at step t_S where a belief-trap is detected (operationalized as $k = 3$ consecutive non-progressive steps for CD and $k = 2$ for MR), zeroing the advantages of all subsequent steps. Our method composes additively with T3 by applying the SNR modifier to the surviving steps.

B.3 Training Configuration

Compute and framework. All experiments are conducted on a single node equipped with 8 NVIDIA A800 GPUs, using the Verl framework. We use Fully Sharded Data Parallelism (FSDP) with BFloat16 precision throughout training and evaluation. Rollouts use vLLM[1] with tensor parallel size 1.

[1] https://docs.vllm.ai/en/latest/

Base model. All RL experiments fine-tune Qwen-2.5-7B-Instruct. We do not modify the model architecture or vocabulary.

Training schedule. All methods train for 200 steps with the actor optimized at learning rate 1.0×10^{-6}. The maximum number of interaction turns per episode is 10 for CD and 5 for MR, following Zou et al..

Rollout sampling. We use temperature 0.6 and top-p 0.95 for both tasks.

Reward configuration. Rewards are sparse, provided only at the terminal step of each trajectory. For both CD and MR, the reward is Exact Match between the agent's final answer and the ground-truth, in $\{0, 1\}$.

PPO-specific hyperparameters. We use GAE with $\lambda_{\text{GAE}} = 1$ and discount $\gamma = 1$. The KL divergence regularization coefficient β is set to 0.001 and the clip ratio ε to 0.2.

GRPO/GSPO-specific hyperparameters. GRPO samples $G = 5$ responses per prompt. GSPO additionally uses sequence-level clipping with $\varepsilon_{\text{low}} = 0.0003$ and $\varepsilon_{\text{high}} = 0.0004$, and disables the KL constraint. Other settings match PPO.

Evaluation. We evaluate on the held-out test set every 20 training steps and report best validation EM across 200 training steps as the primary metric. All EM values in the main text are scaled by 100. A summary of training hyperparameters appears in Table 5.

B.4 SNR Modifier Implementation Details

This subsection documents the implementation details of our SNR-weighted advantage modifier (Sec. 2.3).

Hyperparameters. We use a single set of modifier hyperparameters across both CD and MR, with no per-task tuning. Values are summarized in Table 6.

Step-label proxies. The progress signal g_t is computed from rollout-level observables already maintained by the environment under Zou et al.:

- For CD: the candidate-set reduction ratio defined above.

Hyperparameter	Value
Base model	Qwen-2.5-7B-Instruct
Training framework	Verl
GPUs	8 × A800
Precision	BFloat16
Parallelism	FSDP, tensor parallel = 1
Training steps	200
Learning rate	1.0×10^{-6}
Max interaction turns (CD)	10
Max interaction turns (MR)	5
Rollout temperature	0.6
Rollout top-p	0.95
PPO GAE λ	1.0
PPO discount γ	1.0
PPO KL coefficient β	0.001
PPO clip ratio ε	0.2
GRPO group size G	5
GSPO ε_{low}	0.0003
GSPO $\varepsilon_{\text{high}}$	0.0004
Evaluation frequency	every 20 steps

Table 5: Training hyperparameters used across all RL experiments. Where settings differ between base algorithms, the algorithm-specific value is listed.

Hyperparameter	Value
λ	1.0
μ	0.5
γ_{clip}	5.0
K	3
ϵ	10^{-8}

Table 6: SNR modifier hyperparameters, identical across CD and MR.

- For MR: the binary feedback validity signal $g_t \in \{-1, 0, +1\}$ extracted from environment responses.

No additional model components are required to compute g_t. The signal is collected at the end of each interaction turn and stored alongside the rollout transition.

Noise indicator computation. The noise indicator $n_t = \text{rep}_t + \text{prox}_t$ is computed from the interaction history as follows:

- $\text{rep}_t = |\{\tau < t : a_\tau = a_t\}|$, the count of past actions identical to the current action. Action equality is determined by exact string match on the agent's query.
- $\text{prox}_t = \max(0,\ K - |\{j \in [t-K, t) : g_j > 0\}|)$, the count of non-progress steps within the most recent $K = 3$ steps.

Both indicators are non-negative integers computed in $O(K)$ time per step.

Modifier and advantage update. The modifier M_t (Eq. (4) in Sec. 2.3) is applied to each step's advantage:

$$A_t \leftarrow A_t^{\text{base}} + M_t,$$

where A_t^{base} is the GAE estimate produced by PPO (or the group-relative estimate for GRPO/GSPO). The modifier is broadcast uniformly to all tokens within each interaction step, since active reasoning operates at the step granularity rather than at the token level. No changes are made to the value function, reward model, or training loop beyond the additive modification to A_t.

Computational overhead. The modifier adds $O(T)$ computation per trajectory, where $T \leq 10$ for CD and $T \leq 5$ for MR. In practice, this overhead is negligible compared to the rollout and forward-pass costs. The modifier requires no additional GPU memory beyond a per-step scalar buffer.

Integration with T3. When combined with T3, the modifier is applied only to the steps that survive truncation. Specifically, for trajectories truncated at step t_S, we set $M_t = 0$ for all $t \geq t_S$, and apply the modifier only for $t < t_S$. This preserves T3's trajectory-level cleaning while enabling step-level redistribution within the surviving prefix.

Ack: We use AI to polish our work. But we do every exp by ourself. ALL code are check they are all available for research use.