THE TRACE IS THE STATE: EXACT CREDIT ASSIGNMENT FOR LLM AGENT TEAMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Credit assignment for a team of LLM agents, what each message was worth, has mostly been treated as prediction: a learned critic, a trajectory-level score, or an agent ablation stands in for the counterfactual that defines credit, which is rarely run. Teams that communicate through a shared context are different: when everything a downstream agent reads is written into the trace, the trace is the state, and that counterfactual can be executed. A credit signal can then be judged as any estimator is: by bias, variance, and agreement with an independent reference. C3, credit assignment by counterfactual continuation, substitutes one message at a decision point and continues the run to the terminal reward, so its credit is unbiased, exact up to Monte Carlo error. Given the sampled alternatives, that error's variance follows a derived law with no term for the number of agents, and the observed noise follows the law on 6 workflows of 2 to 10 decision points. On the 2-agent chain, from 4 continuations, C3 ranks alternatives at 0.69 rank correlation against a 16-continuation reference, near the 0.73 at which 2 such references agree; a critic trained on the same rollouts reaches 0.29. Used as the advantage in training a 2-agent team, C3 beats MAGRPO, the stronger baseline in our comparison, and spends 37% fewer training tokens than MAPPO, since only the messages downstream of a decision point are regenerated. When the trace is the state, credit need not be predicted; it can be exact.

1 INTRODUCTION

A team of LLM agents cooperates by reading from and writing to a shared context (Li et al., 2023; Qian et al., 2024; Wu et al., 2024): one agent drafts, another executes, a third checks, a router decides who speaks next, and the wiring may be a chain, a tree, a graph with parallel branches, or a loop that runs until some agent stops it. A run produces a long trace and one outcome, and training the team from that outcome requires answering, for every message in the trace, how much it was worth: credit assignment, the central obstacle to training cooperative LLM systems from a single terminal reward. The deeper the team, the more decisions must be priced from the same outcome.

Reinforcement learning has a standard answer: compare the action taken with the actions the agent could have taken instead, its counterfactual advantage. An LLM's action is an entire message, so alternatives cannot be enumerated, only sampled, and the question becomes what each is worth. There are two ways to find out. One can fit a model that predicts the value of an alternative, a critic (Yu et al., 2022; Foerster et al., 2018), or one can carry the alternative forward: let the rest of the team run to the end and observe the reward. Most work on LLM agent teams predicts rather than runs, or sidesteps the question with a trajectory-level score shared by every decision (Ma et al., 2024; Liu et al., 2025) or with agent ablation, comparing what remains (Li et al., 2026d). The choice is inherited from multi-agent reinforcement learning, where it is correct: in a physical environment revisiting a decision point means resetting the world, which is costly or impossible, so the outcome of a different action is estimated instead (Wolpert & Tumer, 2001; Foerster et al., 2018). The assumption traveled with the tools into LLM teams, even though single-agent LLM training already values a step by re-running from its prefix (Kazemnejad et al., 2024). For a team, two questions follow: when is a re-run from the prefix exact, and how does its credit compare, as an estimator, with predicted credit?

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.

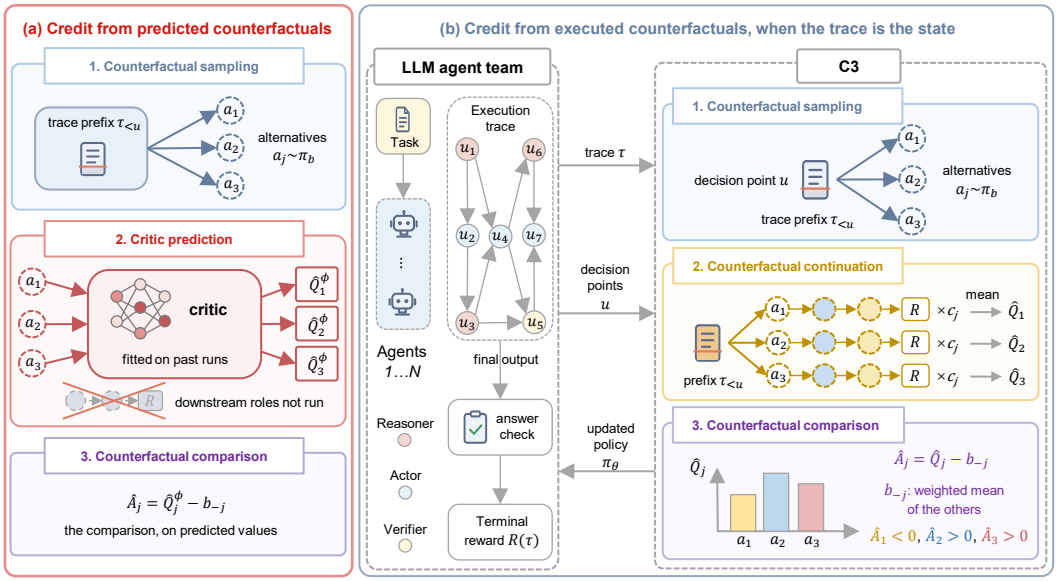


Figure 1: Two ways to value an alternative message. (a) Inherited: the decision point cannot be revisited, so a critic with parameters ϕ predicts each alternative's value $\hat{Q}_j^{\phi}$. (b) When every role reads and writes only the trace, C3 resets to the prefix $\tau_{<u}$, substitutes an alternative a_j, runs the downstream roles to the reward c_j times, and compares the mean reward $\hat{Q}_j$ with the continuation-weighted mean b_{-j} of the other alternatives: executed, not predicted. Rank correlation against a 16-continuation reference: critic 0.29, C3 from 4 continuations 0.69 (Section 4).

Agent teams that communicate through text can revisit any decision point whose inputs the trace records. When every message, tool call, memory read, and routing decision is written into the execution trace, the trace is the complete state of the system. Reset the system to any point in the trace, replace one message, re-run everything downstream: the outcome of the alternative is executed, not predicted, and what remains is sampling noise, which can be quantified. Credit is exact in this sense. The condition that makes a re-run exact, that the trace is the state, holds by construction for teams whose components read only the trace, and only unrecorded hidden state voids it (Section 2.3). C3, credit assignment by counterfactual continuation, is the estimator built on this fact (Figure 1). At each decision point it samples alternatives from the behavior policy π_b, runs each to the terminal reward, and compares each with the continuation-weighted mean return of the other alternatives (Section 2.4).

Reinforcement learning trades the bias of a critic against the variance of Monte Carlo returns; where the counterfactual cannot be run, that trade can be checked only in aggregate and only on the actions taken, and a credit signal is judged by the team it trains. Where it can be run, the value of every alternative at one history is observable. There, every credit signal, whether a critic, a trajectory-level score, an agent ablation, or C3, is used to estimate the same advantage and can be held to the standard of any estimator: its bias, its variance and reliability (Zhang, 2026a), and its agreement with a reference that shares none of its rollouts (Zhang, 2026b). Our contributions answer the three criteria in order and then put C3 to use:

1. *Bias.* When the trace is the state, C3 estimates the per-decision counterfactual advantage without bias and without learned parameters (Sections 2.3 and 2.4). Agent ablation instead estimates a different quantity, the value of the agent's presence: it shifts the downstream observation law and moves every credit at that decision point by one offset that more rollouts do not shrink; a trajectory-level score gives every decision in a run the same number (Sections 2.2 and 2.4).
2. *Variance and reliability.* Given the sampled alternatives, the variance of each advantage estimate follows a law set by that decision point's own budget and return variance, with no term for the number of agents; at the opening decision point of 6 workflows of 2 to 10 decision points the observed variance is 0.90 to 1.07 times the law, and split-half reliability on the same workflows follows the signal share of the variance (Section 3).

3. *Agreement with a reference.* On held-out decision points of the 2-agent chain, from 4 continuations per alternative, C3 ranks alternatives at 0.69 rank correlation against a 16-continuation reference, near the 0.73 at which 2 such references agree; a critic trained on the same rollouts reaches 0.29 (Section 4).
4. *In use.* As the advantage in policy optimization on a 2-agent workflow, C3 beats MAGRPO, the stronger baseline, by $+8.28$ points on MATH500 over 5 training seeds (Holm-corrected $p < 0.001$), and at a 2048-token generation limit spends 37% fewer training tokens than MAPPO (Section 5; Table 2; Appendix B.4).

2 Credit as an Executed Counterfactual

The opening criterion is bias: this section defines the per-decision advantage every credit signal is used to estimate, shows that ablating an agent whose message is read downstream shifts the downstream observation law by an amount more rollouts do not shrink (Proposition 1), and gives the condition under which C3 is unbiased (Definition 1, Proposition 2).

2.1 A Run Is a Sequence of Decision Points

Several LLM agents cooperate by reading from and writing to a shared context, and one run is a sequence of decision points $u = 1, \dots, K$ over the trace they build together. At decision point u a single role reads a context h_u and writes one message a_u. The workflow assembles that context from the trace prefix $\tau_{<u}$, the record of everything written before u, so $h_u = \psi(\tau_{<u})$ for a deterministic assembly map ψ. Which role acts next, and which part of the trace it is shown, are themselves entries in the trace. Chains, trees, directed graphs with parallel branches, dynamic dispatch, and loops are instances of this object.

This is a cooperative partially observable stochastic game (POSG) (Bernstein et al., 2002) played in sequence: at u, agent $i = \mathrm{role}(u)$ draws $a_u \sim \pi_\theta^i(\cdot \mid h_u)$, and after K decision points a scalar terminal reward $R(\tau)$ is scored from the final output (notation in Appendix B.1). A decision is a whole message rather than a token, so K is small (2 to 10 here).

2.2 The Counterfactual and Two Ways to Evaluate It

What a message was worth is a counterfactual question. Credit assignment estimates the per-decision advantage

$$A(u) \;=\; \mathbb{E}_{D_b}\big[R(\tau) \mid h_u,\, a_u\big] \;-\; \mathbb{E}_{D_b}\big[R(\tau) \mid h_u\big], \tag{1}$$

what a_u contributed beyond what h_u already determined, with the continuation drawn under the behavior policy π_b (distribution D_b), the snapshot of π_θ that generated the run; π_b is the sampling policy in the PPO ratio of Section 2.4, not the reference policy of the KL penalty. Alternatives to a whole message can be sampled but not enumerated, and there are two ways to evaluate one (Figure 1): predict its value or execute it.

Work on LLM agents inherited the constraint of physical environments, where only prediction is available once the moment has passed, and three approximations follow. A critic predicts the value of an alternative, and nothing in the run supplies a reference against which its error could be read (Yu et al., 2022; Foerster et al., 2018). A trajectory-level score hands every decision in a run the same number, whatever each decision was worth (Ma et al., 2024; Liu et al., 2025). Agent ablation values an agent by deleting it, which changes what every downstream role that reads it receives (Li et al., 2026d). That change can be quantified. Write $a_\emptyset$ for the null message that ablation substitutes at decision point u_i, and for a downstream decision point j let P^j_{pres} and P^j_{abl} be the laws of the observation j receives with agent i present and ablated. Let S_j collect the messages at u_i that force a downstream observation the null message could never produce, and let $V_j(o) = \mathbb{E}[R(\tau) \mid o_j = o]$ have oscillation $\mathrm{osc}(V_j)$.

Proposition 1 (Distribution shift under ablation)**.** *Assume (A1) the null message has probability below one under $\pi_b^i(\cdot \mid h_{u_i})$ and (A2) downstream contexts are assembled by substitution, so a message that enters a context verbatim can be read back from it (Appendix A.1). Then for every downstream decision point j, $\mathrm{TV}(P^j_{\mathrm{pres}}, P^j_{\mathrm{abl}}) \geq \pi_b^i(S_j \mid h_{u_i})$, and the ablation bias $B^j_i = \mathbb{E}_{P^j_{\mathrm{pres}}}[V_j] - \mathbb{E}_{P^j_{\mathrm{abl}}}[V_j]$*

satisfies $\left|B_i^j\right| \le \mathrm{osc}(V_j) \cdot \mathrm{TV}(P^j_{\mathrm{pres}}, P^j_{\mathrm{abl}})$*; both are fixed by the two laws and* V_j*, so drawing more rollouts does not reduce them.*

The proof is in Appendix A.1. An agent whose message reaches no downstream observation leaves the two laws equal, so ablation is unbiased for an agent that changes nothing; where the message is read verbatim, as here, the shift is at least $1 - \pi_b^i(a_\emptyset \mid h_{u_i})$. B_i^j is the same for every alternative at u_i: it prices the agent's presence rather than the sampled message, so it leaves the order of the alternatives at u_i untouched and moves every credit there by one offset; offsets differ across decision points, which matters when credit is compared across agents or decision points.

2.3 The Trace Is the State

Agents that communicate through text can be re-run from any decision point whose inputs the trace records, and the re-run is exact when the trace is the whole state. Let S_u be the complete state at u, everything that bears on the law of what follows, and let $\tau_{<u}$ be the trace prefix the system records.

Definition 1 (The trace is the state)**.** *The trace is the state at decision point* u *if, for every message* a *to be evaluated,*

$$\mathcal{L}\big(Y_{>u},\, R \,\big|\, \mathrm{do}(a_u = a),\, \tau_{<u}\big) \;=\; \mathcal{L}\big(Y_{>u},\, R \,\big|\, \mathrm{do}(a_u = a),\, S_u\big), \tag{2}$$

where $Y_{>u}$ *collects the downstream messages and* $\mathcal{L}(\cdot)$ *denotes a law.*

Under (2), resetting to $\tau_{<u}$ and injecting a yields a draw from the same interventional law as resetting the true state would, so (1) is identified, under the behavior policy's continuation, with no fitted model in the path. That identification is what *exact* means here: the counterfactual is executed and observed, so credit computed from it is exact up to Monte Carlo error, whose variance, given the sampled alternatives, Section 3 derives. Only the prefix is held fixed; downstream contexts diverge freely after different messages, since that divergence is the effect being priced.

In the workflows of this paper the condition holds by construction, because every role's input is assembled by substitution from the task text, the messages upstream of it, and fixed templates, all recorded in the trace. Elsewhere it is a recording requirement that sorts systems into three cases. *Recorded*: every causally relevant variable, tool returns and memory contents and routing decisions included, is written into the trace and restored on reset, so (2) holds. *Resampled*: a genuinely random component is drawn afresh from its law given the recorded prefix, and the estimate stays unbiased at a variance cost. *Hidden state*: something bearing on the continuation is neither recorded nor resampled, as when a decision point depends on an external world that text cannot restore (Gonuguntla, 2026), and no guarantee applies. A system that cannot be shown to be recorded or resampled is treated as hidden state.

2.4 The Estimator

C3 turns (1) into three operations, none of which adds a learned parameter (Figure 1b).

Counterfactual sampling. At each iteration a snapshot $\pi_b \leftarrow \pi_\theta$ of the current policy is taken and held fixed for the iteration, and base runs are executed under it, recording at every decision point the acting role, the context h_u, and the trace prefix $\tau_{<u}$. For each recorded context, $n \ge 2$ alternatives $a_j \sim \pi_b(\cdot \mid h_u)$ are drawn, forming a group of alternatives that share one context exactly; only the message under evaluation differs across the group, so the shift of Proposition 1 never enters.

Counterfactual continuation. The system is reset to $\tau_{<u}$, a_j is injected in place of the recorded message, and the downstream roles finish the run under π_b. Each alternative receives $c_j \ge 1$ independent rollouts, giving

$$\hat{Q}^{D_b}(h_u, a_j) \;=\; \frac{1}{c_j} \sum_{m=1}^{c_j} R_m, \tag{3}$$

an unbiased estimate of $Q^{D_b}(h_u, a_j) = \mathbb{E}_{D_b}[R(\tau) \mid h_u, a_j]$, under a per-instance budget that fixes the total rollouts at B across decision points and alternatives. Because the prefix is reused rather than regenerated, an evaluation costs only the workflow downstream of u.

Algorithm 1 C3.

Require: Policy π_θ, branching factor n, budget B rollouts per instance, clip threshold ϵ

for each training iteration **do**
  $\pi_b \leftarrow \pi_\theta$ ▷ Snapshot, held fixed for the iteration
  Execute base runs under π_b; record the trace prefixes $\{\tau_{<u}\}$
  for each decision point $(i, h_u, \tau_{<u})$ **do**
    Sample n actions: $a_j \sim \pi_b^i(\cdot \mid h_u)$ for $j = 1, \ldots, n$
  end for
  for each action a_j in each group of alternatives **do** ▷ B rollouts per instance; nested groups (Section 5) share them
    Reset the system to $\tau_{<u}$; inject a_j; roll out under π_b to terminal
    Record R_m for c_j independent rollouts; compute $\hat{Q}^{D_b}(h_u, a_j)$ via Equation (3)
  end for
  for each group of alternatives **do**
    Compute b_{-j} via Equation (4) for all j
  end for
  Update π_θ by maximizing the clipped PPO objective $\mathcal{L}(\theta)$ over all $(h_u, a_j, \hat{A}_j)$ triples
end for

Counterfactual comparison. C3 compares an alternative against its peers: its baseline is the mean return of the other alternatives at the same decision point, weighted by their rollout counts (with equal counts the leave-one-out baseline of RLOO (Kool et al., 2019; Ahmadian et al., 2024), related to difference rewards (Wolpert & Tumer, 2001)); the baseline is subtracted, and the advantage enters the clipped objective of Section 2.4 unchanged in sign and order (Appendix B.2). A within-group baseline cancels the between-group variation that a global baseline would carry in, since different groups face different task instances (Greensmith et al., 2001). For a_j in a group of n alternatives with rollout counts $\{c_k\}$ and $C = \sum_k c_k$,

$$b_{-j} \;=\; \frac{1}{C - c_j} \sum_{k \neq j} c_k \, \hat{Q}^{D_b}(h_u, a_k), \qquad \hat{A}_j \;=\; \hat{Q}^{D_b}(h_u, a_j) \;-\; b_{-j}. \tag{4}$$

Proposition 2 (Unbiasedness)**.** *When the trace is the state at u (Section 2.3), with alternatives drawn independently from π_b, rollouts independent across alternatives under D_b, and non-adaptive rollout counts, $\hat{A}_j$ is, conditional on a_j, an unbiased estimate of $A^*_{D_b}(h_u, a_j) = Q^{D_b}(h_u, a_j) - V^{D_b}(h_u)$, the expectation being over the rollouts and the other alternatives at the recorded prefix $\tau_{<u}$, of which h_u is a function (proof in Appendix A.2).*

Excluding a_j from its own baseline stops an alternative from raising the bar it is judged against, and the context-level term cancels exactly (Appendix A.3).

The advantages enter the clipped surrogate objective of PPO (Schulman et al., 2017) with importance ratio $r(\theta) = \pi_\theta(a \mid h)/\pi_b(a \mid h)$; any policy gradient method that accepts per-sample advantages can take its place. Algorithm 1 lists the loop: at every decision point n alternatives are sampled, each is continued c_j times and compared within its group, and the policy is updated on every (context, alternative, advantage) triple.

3 Variance and Reliability from 2 to 10 Agents

The second criterion is variance and reliability: how much an advantage estimate moves when its rollouts are drawn again, and so how many rollouts one decision point needs. Two quantities, computed from the groups collected anyway and needing no per-decision label, answer this, and a third says whether there is anything to credit.

Reliability is whether a group orders its own alternatives the same way twice: split its rollouts into 2 halves, the even-numbered and the odd-numbered continuations, build an advantage vector from each half by (4), and take the Spearman rank correlation between the 2 vectors, averaged over the groups whose halves both order the alternatives (Appendix B.5). **Within-group variance**, the sample variance of the advantages across a group, is their total spread, signal and noise together (Appendix A.3). **Influence**, the mutual information $I(J; Y \mid h_u)$ between the injected alternative

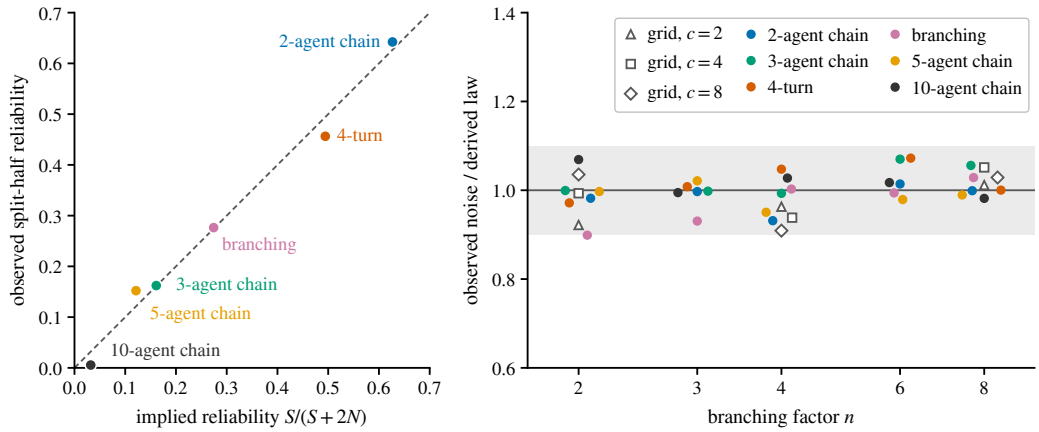


Figure 2: Reliability follows the signal share of the variance, and the noise follows the law. Left: split-half reliability of 6 workflows on a policy held fixed, against the ratio $S/(S+2N)$ of the signal variance S to the variance $S+2N$ of a 2-continuation half, with S and N from Table 3 and the reliability from Table 1; the dashed line is equality. Right: the ratio of the observed noise on an advantage to the derived law, over a grid of branching factors and continuation counts on the 2-agent chain (markers) and over the 6 workflows (colors); the band marks $\pm 10\%$.

J and the downstream response Y (Jaques et al., 2019; Appendices A.4 and A.6), says whether there is any coupling for credit to describe: near zero the downstream role is ignoring the upstream message (Zhang et al., 2025).

This section and the next use Qwen3-4B-Instruct-2507 (Yang et al., 2025a), written Qwen3-4B below, and Qwen3-8B where named, held fixed at their released weights, so that what they report describes the estimator rather than a training outcome. The 2-agent workflow is the Reasoner $\rightarrow$ Actor pair of MARFT (Liao et al., 2025): the Reasoner writes a plan, the Actor writes the solution that gets graded; the 3-agent workflow adds a Verifier that checks the answer, and deeper workflows reach a 10-agent chain (Appendix B.5).

The noise on each advantage is set by the rollouts one decision point receives and by how much the returns after it vary, not by the number of agents. Let B_u be the budget spent at a decision point, spread over n alternatives with $c = B_u/n$ rollouts each. Conditional on the sampled alternatives, the rollout noise on one advantage estimate has variance $\sigma^2 n^2/(B_u(n-1))$, with σ^2 the variance of the return given the alternative, taken equal across alternatives (Appendices A.3 and A.5), and the observed noise matches this expression to within 9% across a 3×3 grid of branching factors and continuation counts (Appendix A.5 and Table 3). A fixed per-instance budget B spread evenly over K decision points gives $B_u = B/K$, so depth enters through that division: at one rollout per alternative the branching factor is $n = B/K$, and below $n = 2$ there is no other alternative to compare against, which $B = 8$ reaches at 5 decision points. Across 6 workflows of 2 to 10 decision points and every branching factor tested, the ratio of the observed noise on an advantage to the derived law stays between 0.90 and 1.07 on Qwen3-4B (Table 3), with no trend in the branching factor (rank correlation of the ratio with n over the 43 cells of the branching scan, 30 on Qwen3-4B and 13 on Qwen3-8B: $+0.11$, 95% interval -0.20 to $+0.39$), so the noise is set by the per-decision budget and the return variance.

Table 1 reports split-half reliability, at the opening decision point of each workflow and on a policy held fixed, for 6 workflows of 2 to 10 decision points at a constant per-decision budget of 4 alternatives and 4 continuations each (Appendix B.5): from 0.64 on the 2-agent chain and 0.46 on the 4-turn workflow to 0.15 on a 5-agent chain and 0.01 on a 10-agent chain, with a 4-role workflow with 2 parallel branches at 0.28. At this design point these values agree with what the signal and the noise imply. Each half of a split averages 2 continuations, so its noise variance is $2N$ for N the sampling variance of a 4-continuation mean, and under an additive model the correlation between the 2 halves is $S/(S+2N)$, with S the variance of the alternatives' true values at a decision point, the spread the ranking rests on (Appendix A.5). Both variances are read from Table 3, and this ratio matches the observed reliability of all 6 workflows to within 0.04 (Figure 2, Table 1). What separates the workflows is therefore the signal, not the noise: S falls from 0.077 on the 2-agent chain to 0.003 on

Table 1: Reliability follows the signal share of the variance. Constant per-decision budget: 4 alternatives per decision point, 4 continuations each, 500 groups collected per cell. Observed: split-half reliability on a policy held fixed, the rank agreement between the even-numbered and the odd-numbered continuations of each group. Implied: $S/(S+2N)$, the share of a 2-continuation half's variance that is signal, from the signal variance S and the noise variance N of Table 3 (Appendix A.5). Groups: those whose halves both order the alternatives, out of 500 (Appendix B.5). The branching workflow, the one here that is not a chain, is a planner, 2 parallel solvers and an integrator.

Workflow (K)	Alternatives, $4K$	Observed	Implied	Within-group variance	Groups
2-agent chain (2)	8	0.64	0.63	0.179	326
3-agent chain (3)	12	0.16	0.16	0.092	333
4-turn workflow (4)	16	0.46	0.49	0.082	163
Branching workflow (4)	16	0.28	0.27	0.103	316
5-agent chain (5)	20	0.15	0.12	0.092	328
10-agent chain (10)	40	0.01	0.03	0.093	346

the 10-agent chain, whose opening decision point, the Reader, leaves little to rank, while N rises from 0.023 to 0.049, and the noise on each advantage follows the law throughout. When an even split of 8 over the decision points sets the alternatives per decision point, 4 on the 2-agent chain and 3 on the 3-agent chain (8 over 3, rounded), reliability is 0.23 on the 3-agent chain against 0.64 on the 2-agent chain, and Qwen3-8B reproduces the pattern at 0.34 against 0.71; Figure 3 in Appendix B.5 sweeps the branching factor for every workflow.

4 Prediction and Ablation against the Executed Counterfactual

The third criterion is agreement with a reference that shares none of the estimator's rollouts, and fresh continuations of the same decision points supply one; executed returns also give the realized bias of ablation.

From 4 continuations, C3 ranks alternatives at 0.69 rank correlation against a 16-continuation reference; a critic trained on the same rollouts reaches 0.29. A state-value critic $V(h_u)$ shifts every alternative at a decision point by the same constant and leaves their order untouched, so the comparison that matters is a decision-value critic $\hat{Q}(h_u, a)$ that scores each alternative on its own, trained on the rollouts C3 collected on the 2-agent workflow of Section 3, with held-out points split from them by decision point (Appendix B.7). Both are scored on held-out decision points against a Monte Carlo reference of 16 fresh continuations per alternative, which is unbiased under the condition of Section 2.3; C3 uses the 4 continuations per alternative it already had, and the gap is $+0.40$ in rank correlation (paired, $p < 0.001$). At the reference's own budget, 16 fresh continuations per alternative agree with the reference at 0.73, so the 4-continuation estimate already sits near the level at which 2 such references agree (on the points all three order at every budget, 0.70 against 0.76; Appendix B.7). At the scored points the critic spends no rollouts, the saving for which a critic accepts its bias, and retraining it on 2 and 4 times the rollouts does not close the gap (Appendix B.7).

Ablating an agent shifts what the roles downstream of it read. We computed how much of that shift reaches the reward by replacing one role's message with a null action in 150 groups of the 3-agent workflow, one per MATH500 question, and comparing the return against the group's mean under π_b. The per-point values, stratified by answer-level influence (Appendix A.6) and by whether the sampled alternatives received different values, are summarized in Appendix B.6. The empty-message form of the null action is the agent-ablation counterfactual that CCPO (Li et al., 2026d) subtracts as its baseline.

5 C3 as the Advantage in Training

C3 is put to use as the advantage in policy optimization, and the 3 methods share everything but how rollouts are arranged and credited. The tasks are math benchmarks whose final answer is checked against a reference: MATH500 (Hendrycks et al., 2021), AIME (Mathematical Association of America, 2025), CMATH (Wei et al., 2023) and GSM8K (Cobbe et al., 2021) in the 5-seed run, and MATH500, Minerva-Math (Lewkowycz et al., 2022), AMC23 (Mathematical Association of America, 2023) and AIME with its 2024 and 2025 sets merged in the long-generation run. The

Table 2: The 5-seed run on Qwen3-4B: greedy accuracy (%) under a 512-token generation limit, mean ± std over 5 training seeds on the 2-agent workflow; best value per column in bold, including ties at the printed precision. †Avg. is the mean of MATH500, CMATH and GSM8K; AIME 2025 (30 problems) is reported beside it, not inside it.

Method	MATH500	AIME 2025	CMATH	GSM8K	Avg.†
MAPPO	69.3 ± 0.9	3.3 ± 0.0	95.3 ± 0.2	92.9 ± 0.1	85.8 ± 0.3
MAGRPO	74.5 ± 0.4	5.3 ± 1.8	96.1 ± 0.2	$\mathbf{93.4 \pm 0.3}$	88.0 ± 0.1
C3	$\mathbf{82.8 \pm 0.6}$	$\mathbf{8.0 \pm 1.8}$	$\mathbf{96.3 \pm 0.2}$	$\mathbf{93.4 \pm 0.2}$	$\mathbf{90.9 \pm 0.3}$

3 methods share the 2-agent workflow of Section 3, the released Qwen3-4B as starting point (the starting model row of Table 7), and the rollout budget, and differ in where the advantage comes from: a centralized critic's value per decision point (MAPPO (Yu et al., 2022)), one trajectory-level score normalized within a group of trajectories and handed to every decision point in them (MAGRPO (Liu et al., 2025)), or the estimator of Section 2.4 (C3). The behavior policy π_b, held fixed for the iteration, generates all rollouts at $B = 8$ rollouts per instance, which C3 splits into 2 Reasoner alternatives, each continued by 4 Actor alternatives, where the baselines draw 8 independent runs. Hyperparameters are in Appendix B.2.

The comparison is run under 2 settings. Table 2 is the 5-seed run: 5 training seeds per method, greedy decoding under a 512-token generation limit on MATH500, AIME 2025, CMATH and GSM8K; its paired differences are tested in Appendix B.8, and the MATH500 gain of $+8.28$ points over MAGRPO, the stronger baseline, comes from it. The long-generation run trains one model per method under a 2048-token limit and evaluates it 5 times with sampled decoding on MATH500, Minerva-Math, AMC23 and AIME (Table 7; the training tokens are in Appendix B.4). MAGRPO diverged late in that run, its KL to the reference policy rising sharply and its responses running to the 2048-token cap; its final model and the last model saved before the divergence, at step 250, are reported beside the others in Table 7. On the 4-benchmark average C3 gains $+2.26$ points over the starting model against $+1.17$ for MAPPO, on AMC23 it gains $+5.28$ points, $+2.78$ more than MAPPO, and it spends 37% fewer training tokens than MAPPO.

6 What Exactness Requires

What a system has to record. Section 2.3 reduces the condition that the trace is the state, on which the unbiasedness of C3 rests, to a list of what a trace must contain. Tool and retrieval calls log their inputs, outputs, versions, and side effects, or are resampled from their law given the recorded prefix at the variance cost of Section 3; persistent memory is serialized into the trace rather than kept beside it; orchestration writes its routing decisions into the trace as text. The guarantee requires all three.

Cost. The price of exactness is the continuations, n alternatives with c rollouts each at every credited decision point, each rollout a run of the downstream roles only. That is why C3's training tokens come in below MAPPO's in Section 5 although it executes every counterfactual it credits: only the messages downstream of a decision point are regenerated, where a full-episode baseline regenerates every message of every rollout; on the 2-agent workflow the 2×4 tree writes 10 messages per instance where 8 independent runs write 16, the ratio behind the saving reported there.

Tasks and rewards. C3 is exact for the terminal reward it is given. Every task here ends in a checkable final answer, so a return varies only through the continuations, which is the noise the law of Section 3 describes; a reward scored by a judge adds the judge's variance to σ^2, and the same law covers it.

7 Related Work

Credit assignment in cooperative multi-agent RL. This literature developed under the premise that executing a counterfactual requires privileged access to the environment. MAPPO (Yu et al., 2022) centralizes a critic, value factorization (Sunehag et al., 2017; Rashid et al., 2020; Wang et al., 2020) decomposes joint values under Markov and discrete-action assumptions, difference rewards (Wolpert & Tumer, 2001) replace one agent's action with a fixed default, and COMA (Foerster et al., 2018) marginalizes over enumerable discrete actions at a single timestep. Where the trace is the state,

that premise fails: the counterfactual COMA reaches for can be executed, downstream continuation included.

Credit assignment for multi-agent LLM systems. MARFT (Liao et al., 2025) brings reinforcement fine-tuning to LLM agent systems. MAGRPO (Liu et al., 2025) and CORY (Ma et al., 2024) assign credit at the trajectory level, and Dr. MAS (Feng et al., 2026) traces the instability of such group-normalized objectives to a baseline shared across agents whose reward distributions differ. SHARP (Li et al., 2026c) attributes credit by Shapley values, MAPPA (Li et al., 2026a) by per-action process rewards from AI feedback, and CCPO (Li et al., 2026d) by ablating an agent, which introduces the shift of Section 2.2. CCPO is defined for a 2-agent chain, re-prompting the single downstream agent without the ablated message; Yang et al. (2025b) propose, as a framework, ablating a whole agent from the recorded prefix onward, with judge scores spreading its Shapley share over its messages. Appendix B.6 prices the shift. COSAC (Deshmukh et al., 2026) fits an additive per-agent decomposition without new reward calls. MALT (Motwani et al., 2024) labels each role's outputs from a tree of sampled continuations and trains on the labels off-policy; AT-GRPO (Zhao et al., 2026) forms groups of same-context samples for one agent at one turn, as C3 does, but scores each by an immediate verifier reward and continues only the best, where C3 continues every alternative to the terminal reward. Concurrent work budgets executed counterfactual evaluations against learned signals (Ben-Gal & Tong, 2026), allocates credit from actions down to tokens while conserving the total (Ma et al., 2026), and prices what individual messages contribute by re-running a trajectory without one of its messages (Yang et al., 2026). Appendix B.9 tabulates what each of these signals restores, re-runs and compares. Relative to these, this paper holds a team's credit to the standard of an estimator: it states when re-running from the prefix is unbiased, derives how the rollout noise depends on the per-decision budget, scores learned and ablation-based credit against executed counterfactuals, and uses the continuations as on-policy advantages.

Counterfactuals from the same prefix, and whether restoring it is exact. At the turn level of single-agent RL, estimating counterfactuals from the same prefix is standard (Kazemnejad et al., 2024; Ji et al., 2025; Liao et al., 2026; Meng, 2026; Li et al., 2026b; Shah, 2026; Tan et al., 2026); a report on tool-using agents instead spreads a discounted terminal return over the tool-call chunks of one trajectory (Wang et al., 2025). Process reward models (Lightman et al., 2024) and turn-level reward designs (Wei et al., 2025) supply dense supervision; C3 needs only a terminal reward. Whether restoring a text history restores the decision point has been examined directly: a survey (Zhang, 2026a) discusses unbiasedness for the same-prefix estimator when the pre-intervention state is restored and both actions are supported, non-identifiability for text matching without restoration, and asks continuation-based estimators to report a test-retest floor (Section 3). Matteson (2026) shows that re-feeding a prefix on batch-variant inference kernels can differ from resuming the live decoder state by a near mean-zero perturbation, so our split-half check is taken at the level of returns (Section 3); Gonuguntla (2026) shows that restoring text does not restore an external environment, the hidden-state case of Section 2.3. Zhang (2026b) notes that a criterion computed against the auditing tool's own estimates certifies only self-agreement; Section 4 therefore scores C3 and the critic against 16 fresh continuations drawn under a separate seed, independent of their own rollouts.

8 Conclusion

Credit assignment for LLM agent teams has mostly been treated as prediction because in physical environments counterfactuals cannot be run. For agents that communicate through a shared context and read only what the trace records, the trace is the state, so they can: reset, replace one message, re-run. C3 turns this into a per-decision estimator that is unbiased, whose rollout noise follows the derived law from 2 to 10 agents, and whose agreement with an independent reference sits near the level at which two such references agree; used in training, it produces a better team and spends fewer training tokens. The prescription: log everything the roles read, spend rollouts per decision point, and check every credit signal against the counterfactual it replaces. The three criteria are not specific to C3: a critic, a trajectory-level score and an agent ablation each stand in for the same per-decision advantage. Where the trace is not the state, Section 6 lists what to record; where it is, the executed counterfactual is the reference to score them against before any is trusted to train a team. When the trace is the state, credit need not be predicted; it can be exact.

AI USE STATEMENT

We used generative AI tools to implement methods, to provide feedback on the experimental design, to help interpret results, to write and edit code, and to polish the text. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

All training and evaluation hyperparameters are listed in Appendix B.2; the complete prompt templates for the 2-agent and 3-agent workflows are in Appendix B.3, and the 4 deep workflows and the reliability measurements are described in Appendix B.5; the proofs of Propositions 1 and 2 and the variance and influence derivations are in Appendix A; the procedures behind Section 4 (the ablation-bias map and the learned-critic comparison) are specified in Appendices B.6 and B.7; and the significance testing procedure is in Appendix B.8. Every arm of the long-generation run on every benchmark, the 2 saturation controls and the 3-agent workflow of the 5-seed run included, is in Appendix B.4. Code will be released upon publication.

REFERENCES

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

Elai Ben-Gal and Stela Tong. Marginal-contribution policy gradients under filtered feedback for multi-agent LLMs. *ArXiv preprint*, abs/2604.22785, 2026. URL https://arxiv.org/abs/2604.22785.

Daniel S Bernstein, Robert Givan, Neil Immerman, and Shlomo Zilberstein. The complexity of decentralized control of Markov decision processes. *Mathematics of Operations Research*, 27(4): 819–840, 2002.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *ArXiv preprint*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

Shripad Deshmukh, Jayakumar Subramanian, Raghavendra Addanki, and Nikos Vlassis. COSAC: Counterfactual credit assignment in sequential cooperative teams. *ArXiv preprint*, abs/2604.17693, 2026. URL https://arxiv.org/abs/2604.17693.

Lang Feng, Longtao Zheng, Shuo He, Fuxiang Zhang, and Bo An. Dr. MAS: Stable reinforcement learning for multi-agent LLM systems. *ArXiv preprint*, abs/2602.08847, 2026. URL https://arxiv.org/abs/2602.08847.

Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In Sheila A. McIlraith and Kilian Q. Weinberger (eds.), *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pp. 2974–2982. AAAI Press, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17193.

Ashritha Gonuguntla. The replay gap: Static evaluation of model switching in LLM agents scores the wrong world. *ArXiv preprint*, abs/2608.08239, 2026. URL https://arxiv.org/abs/2608.08239.

Evan Greensmith, Peter L. Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani (eds.), *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pp. 1507–1514. MIT Press, 2001. URL https://proceedings.neurips.cc/paper/2001/hash/584b98aac2dddf59ee2cf19ca4ccb75e-Abstract.html.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. *ArXiv preprint*, abs/2103.03874, 2021. URL https://arxiv.org/abs/2103.03874.

Natasha Jaques, Angeliki Lazaridou, Edward Hughes, Çaglar Gülçehre, Pedro A. Ortega, DJ Strouse, Joel Z. Leibo, and Nando de Freitas. Social influence as intrinsic motivation for multi-agent deep reinforcement learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 3040–3049. PMLR, 2019. URL http://proceedings.mlr.press/v97/jaques19a.html.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. Tree search for LLM agent reinforcement learning. arXiv:2509.21240, 2025.

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

Wouter Kool, Herke van Hoof, and Max Welling. Buy 4 REINFORCE samples, get a baseline for free! In *ICLR 2019 Workshop on Deep Reinforcement Learning Meets Structured Prediction*, 2019. URL https://openreview.net/forum?id=r1lgTGL5DE.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. *ArXiv preprint*, abs/2206.14858, 2022. URL https://arxiv.org/abs/2206.14858.

Ed Li, Junyu Ren, and Cat Yan. Scaling multiagent systems with process rewards. *ArXiv preprint*, abs/2601.23228, 2026a. URL https://arxiv.org/abs/2601.23228.

Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for "mind" exploration of large language model society. *Advances in Neural Information Processing Systems*, 36:51991–52008, 2023.

Mukai Li, Qingcheng Zeng, Tianqing Fang, Zhenwen Liang, Linfeng Song, Qi Liu, Haitao Mi, and Dong Yu. Verified critical step optimization for LLM agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, 2026b. URL https://arxiv.org/abs/2602.03412.

Yanming Li, Xuelin Zhang, WenJie Lu, Ziye Tang, Maodong Wu, Haotian Luo, Tongtong Wu, Zijie Peng, Hongze Mi, Yibo Feng, Naiqiang Tan, Chao Huang, Lian Peng, and Li Shen. Who deserves the reward? SHARP: Shapley credit-based optimization for multi-agent system. *ArXiv preprint*, abs/2602.08335, 2026c. URL https://arxiv.org/abs/2602.08335.

Zhongyi Li, Wan Tian, Jinju Chen, Huiming Zhang, Yang Liu, Yikun Ban, and Fuzhen Zhuang. Counterfactual credit policy optimization for multi-agent collaboration. *ArXiv preprint*, abs/2603.21563, 2026d. URL https://arxiv.org/abs/2603.21563.

Junwei Liao, Muning Wen, Jun Wang, and Weinan Zhang. MARFT: Multi-agent reinforcement fine-tuning. *ArXiv preprint*, abs/2504.16129, 2025. URL https://arxiv.org/abs/2504.16129.

Sicong Liao, Zhi Chen, and Yaohua Tang. TCPO: Turn-level credit policy optimization. *ArXiv preprint*, abs/2608.01667, 2026. URL https://arxiv.org/abs/2608.01667.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Shuo Liu, Tianle Chen, Zeyu Liang, Xueguang Lyu, and Christopher Amato. LLM collaboration with multi-agent reinforcement learning. *ArXiv preprint*, abs/2508.04652, 2025. URL https://arxiv.org/abs/2508.04652.

Hao Ma, Tianyi Hu, Zhiqiang Pu, Boyin Liu, Xiaolin Ai, Yanyan Liang, and Min Chen. Coevolving with the other you: Fine-tuning LLM with sequential cooperative multi-agent reinforcement learning. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024. URL https://arxiv.org/abs/2410.06101.

Lichao Ma, Yang Sun, Shuaitao Zhao, Yangyi Fang, Cong Qin, Xiaoliang Fu, Yuhang Tian, Yuchen Wei, Junbo Zhu, Yang Wei, Lu Pan, and Jiaye Lin. How much, then where: Credit-conserving action-to-token allocation for multi-turn agent reinforcement learning. *ArXiv preprint*, abs/2608.07118, 2026. URL https://arxiv.org/abs/2608.07118.

Mathematical Association of America. American mathematics competitions (AMC 12), 2023. URL https://artofproblemsolving.com/wiki/index.php/2023_AMC_12A.

Mathematical Association of America. American invitational mathematics examination (AIME), 2025. URL https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

Nils Matteson. Re-feeding is not replaying: Measuring replay noise in counterfactual token-credit estimation. *ArXiv preprint*, abs/2606.15621, 2026. URL https://arxiv.org/abs/2606.15621.

Renwei Meng. Policy-conditioned counterfactual credit for verifiable reinforcement learning of long-horizon language agents. *ArXiv preprint*, abs/2606.05263, 2026. URL https://arxiv.org/abs/2606.05263.

Sumeet Ramesh Motwani, Chandler Smith, Rocktim Jyoti Das, Rafael Rafailov, Ivan Laptev, Philip H. S. Torr, Fabio Pizzati, Ronald Clark, and Christian Schroeder de Witt. MALT: Improving reasoning with multi-agent LLM training. *arXiv preprint arXiv:2412.01928*, 2024.

Liam Paninski. Estimation of entropy and mutual information. *Neural Computation*, 15(6):1191–1253, 2003.

Chen Qian, Wei Liu, Hongzhang Liu, Nuo Chen, Yufan Dang, Jiahao Li, Cheng Yang, Weize Chen, Yusheng Su, Xin Cong, et al. ChatDev: Communicative agents for software development. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15174–15186, 2024.

Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. Monotonic value function factorisation for deep multi-agent reinforcement learning. *Journal of Machine Learning Research*, 21(178):1–51, 2020.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *ArXiv preprint*, abs/1707.06347, 2017. URL https://arxiv.org/abs/1707.06347.

Jaineet Shah. Causal agent replay: Counterfactual attribution for LLM-agent failures. *ArXiv preprint*, abs/2606.08275, 2026. URL https://arxiv.org/abs/2606.08275.

Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning. *ArXiv preprint*, abs/1706.05296, 2017. URL https://arxiv.org/abs/1706.05296.

Hui-Ze Tan, Xiao-Wen Yang, Hao Chen, Jie-Jing Shao, Yi Wen, Yuteng Shen, Weihong Luo, Xiku Du, Lan-Zhe Guo, and Yu-Feng Li. Hindsight credit assignment for long-horizon LLM agents. *ArXiv preprint*, abs/2603.08754, 2026. URL https://arxiv.org/abs/2603.08754.

Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. QPLEX: Duplex dueling multi-agent Q-learning. *ArXiv preprint*, abs/2008.01062, 2020. URL https://arxiv.org/abs/2008.01062.

Weixun Wang, XiaoXiao Xu, et al. Let it flow: Agentic crafting on rock and roll, building the ROME model within an open agentic learning ecosystem. arXiv:2512.24873, 2025.

Quan Wei, Siliang Zeng, Chenliang Li, Zhongruo Wang, William Brown, Oana Frunza, Wei Deng, Anderson Schneider, Yuriy Nevmyvaka, Yang Katie Zhao, Alfredo Garcia, and Mingyi Hong. Reinforcing multi-turn reasoning in LLM agents via fine-grained reward structure and credit assignment. *ArXiv preprint*, abs/2505.11821, 2025. URL https://arxiv.org/abs/2505.11821.

Tianwen Wei, Jian Luan, Wei Liu, Shuang Dong, and Bin Wang. CMATH: Can your language model pass Chinese elementary school math test? *ArXiv preprint*, abs/2306.16636, 2023. URL https://arxiv.org/abs/2306.16636.

David H Wolpert and Kagan Tumer. Optimal payoff functions for members of collectives. *Advances in Complex Systems*, 4(02n03):265–279, 2001.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling*, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *ArXiv preprint*, abs/2505.09388, 2025a. URL https://arxiv.org/abs/2505.09388.

Chih-Hsuan Yang, Tanwi Mallick, Le Chen, Krishnan Raghavan, Amal Gueroudji, Ian T. Foster, and Rajeev Thakur. Who gets the reward & who gets the blame? evaluation-aligned training signals for multi-LLM agents. *ArXiv preprint*, abs/2511.10687, 2025b. URL https://arxiv.org/abs/2511.10687.

Chih-Hsuan Yang, Anjir Ahmed Chowdhury, Cheng-Hau Yang, Weijian Zheng, Fernando Llorente, Xiaolong Ma, Xinyang Li, Eliu A. Huerta, Ian T. Foster, and Rajeev Thakur. Wrong but useful: Trajectory value beyond answer correctness in multi-agent messages. *ArXiv preprint*, abs/2608.14375, 2026. URL https://arxiv.org/abs/2608.14375.

Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. The surprising effectiveness of PPO in cooperative multi-agent games. *Advances in Neural Information Processing Systems*, 35:24611–24624, 2022.

Chenchen Zhang. From reasoning to agentic: Credit assignment in reinforcement learning for large language models. *ArXiv preprint*, abs/2604.09459, 2026a. URL https://arxiv.org/abs/2604.09459.

Haiyue Zhang. Credit without ground truth: Auditing step-level credit assignment in LLM agents against executed replay. *ArXiv preprint*, abs/2608.19760, 2026b. URL https://arxiv.org/abs/2608.19760.

Zhiwei Zhang, Xiaomin Li, Yudi Lin, Hui Liu, Ramraj Chandradevan, Linlin Wu, Minhua Lin, Fali Wang, Xianfeng Tang, Qi He, and Suhang Wang. Unlocking the power of multi-agent LLM for reasoning: From lazy agents to deliberation. *ArXiv preprint*, abs/2511.02303, 2025. URL https://arxiv.org/abs/2511.02303.

Yujie Zhao, Lanxiang Hu, Yang Wang, Minmin Hou, Hao Zhang, Ke Ding, and Jishen Zhao. StrongerMAS: Multi-agent reinforcement learning for collaborative LLMs. In *The Fourteenth International Conference on Learning Representations (ICLR)*, 2026. URL https://arxiv.org/abs/2510.11062.

A Theoretical Proofs and Derivations

When the trace is the state (Section 2.3), counterfactual credit assignment is executed rather than predicted; the same property makes credit quality checkable against executed counterfactuals (Section 4). This appendix provides the complete proofs underlying both consequences. Appendix A.1 shows that agent ablation induces a systematic shift in the downstream observation law that cannot be reduced by additional sampling; Appendix A.2 establishes the unbiasedness of the C3 advantage estimator for counterfactual rollouts from the same prefix; Appendix A.3 derives the variance reduction properties of within-group comparison; Appendix A.4 develops the conditional mutual information estimator for the dependence between an injected action and the downstream response; Appendix A.5 collects the budget arithmetic behind Section 3; Appendix A.6 defines the response-level mutual information of Section 3 and the answer-level quantity used in Section 4 and Appendix B.6.

A.1 Distribution Shift Induced by Agent Ablation

This subsection restates Proposition 1 from Section 2.2 in quantitative form and gives its proof. The statement separates two quantities that the qualitative reading runs together: how far ablating an agent moves the downstream observation law, and how much reward bias that movement produces. The separation is what makes the second quantity an empirical question, answered in Appendix B.6.

Setup. Agent i acts at decision point u_i with history h_{u_i}, and j is a downstream decision point with $u_j > u_i$. Ablation replaces the action at u_i by the null action $a_\emptyset$. Write O_j for the serialized observation that decision point j receives, and let

$$P^j_{\text{pres}} \;=\; \mathcal{L}\big(O_j \mid a_{u_i} \sim \pi^i_b(\cdot \mid h_{u_i})\big), \qquad P^j_{\text{abl}} \;=\; \mathcal{L}\big(O_j \mid a_{u_i} = a_\emptyset\big)$$

be its law with the agent present and with the agent ablated. Let $V_j(o) = \mathbb{E}[R(\tau) \mid O_j = o]$ be the continuation value as a function of that observation, with oscillation $\operatorname{osc}(V_j) = \sup V_j - \inf V_j$. Total variation distance is $\operatorname{TV}(P, Q) = \sup_A |P(A) - Q(A)| = \frac{1}{2}\sum_o |P(o) - Q(o)|$ on the discrete observation space at hand. Two standing assumptions carry over from Proposition 1:

- (A1) The behavior policy is non-degenerate at u_i: $\pi^i_b(a_\emptyset \mid h_{u_i}) < 1$.
- (A2) Serialization: the observation o_j at a downstream decision point is assembled by substitution from the task text and the messages upstream of it, so that when the message produced at u_i enters o_j verbatim there is a map φ with $\varphi(o_j) = a_{u_i}$.

Let $P(o_j \mid a)$ denote the law of O_j when the action at u_i is a and everything downstream is sampled as usual. Define

$$S_j \;=\; \big\{\, a : \operatorname{supp} P(o_j \mid a) \cap \operatorname{supp} P(o_j \mid a_\emptyset) = \emptyset \,\big\},$$

the actions at u_i whose downstream observations can never coincide with one produced under the null action. The definition is marginal: no conditioning on the downstream sampling stream is involved, and the bound below is a statement about the two marginal laws.

Proposition 3 (Distribution Shift Induced by Agent Ablation, quantitative form)**.** *Under (A1) and (A2), for every downstream decision point j with $u_j > u_i$:*

- *(i)* $\operatorname{TV}\big(P^j_{\text{pres}}, P^j_{\text{abl}}\big) \;\geq\; \pi^i_b(S_j \mid h_{u_i})$. *In workflows where the message produced at u_i appears verbatim in the prompt read at j, $S_j \supseteq \mathcal{A} \setminus \{a_\emptyset\}$ and therefore $\operatorname{TV}\big(P^j_{\text{pres}}, P^j_{\text{abl}}\big) \geq 1 - \pi^i_b(a_\emptyset \mid h_{u_i}) > 0$. If $S_j = \emptyset$, this lower bound is vacuous.*
- *(ii) The ablation bias $B^j_i = \mathbb{E}_{P^j_{\text{pres}}}[V_j] - \mathbb{E}_{P^j_{\text{abl}}}[V_j]$ satisfies $\big|B^j_i\big| \;\leq\; \operatorname{osc}(V_j) \cdot \operatorname{TV}\big(P^j_{\text{pres}}, P^j_{\text{abl}}\big)$.*
- *(iii) B^j_i is a functional of the pair of laws and of V_j alone. It does not depend on how many rollouts are drawn, so increasing the rollout count does not reduce it.*

Proof. Part (i). Let $E = \big(\operatorname{supp} P(o_j \mid a_\emptyset)\big)^c$, the set of downstream observations that cannot arise under the null action. By definition $P^j_{\text{abl}}(E) = 0$. Under P^j_{pres} the action is drawn from $\pi^i_b(\cdot \mid h_{u_i})$,

and for every $a \in S_j$ the law $P(o_j \mid a)$ puts all its mass in E, so

$$P^j_{\text{pres}}(E) \;=\; \sum_a \pi^i_b(a \mid h_{u_i})\, P(o_j \mid a)(E) \;\geq\; \pi^i_b(S_j \mid h_{u_i}).$$

Since total variation distance is the supremum of $|P(A) - Q(A)|$ over events,

$$\mathrm{TV}\big(P^j_{\text{pres}}, P^j_{\text{abl}}\big) \;\geq\; \big|P^j_{\text{pres}}(E) - P^j_{\text{abl}}(E)\big| \;=\; P^j_{\text{pres}}(E) \;\geq\; \pi^i_b(S_j \mid h_{u_i}).$$

For the verbatim case, (A2) gives a map φ with $\varphi(o_j) = a_{u_i}$ on the support of every $P(o_j \mid a)$; observations produced under distinct actions therefore lie in disjoint sets, so $S_j \supseteq \mathcal{A} \setminus \{a_\emptyset\}$ and $\pi^i_b(S_j \mid h_{u_i}) \geq 1 - \pi^i_b(a_\emptyset \mid h_{u_i})$, which is strictly positive by (A1). This holds for every downstream decision point whose observation contains the message, whatever sampling occurs in between, because the argument uses only the marginal laws. At the opposite extreme, an agent whose message enters no downstream observation leaves $P(o_j \mid a) = P(o_j \mid a_\emptyset)$ for every action, so the two laws coincide, the shift is zero, and by part (ii) so is the bias; a redundant agent is exactly this case.

Part (ii). Write $P = P^j_{\text{pres}}$ and $Q = P^j_{\text{abl}}$, and let

$$m \;=\; \tfrac{1}{2}\big(\sup V_j + \inf V_j\big), \qquad W \;=\; V_j - m.$$

Shifting by the midpoint centers the function: $\sup W = \frac{1}{2}\mathrm{osc}(V_j)$ and $\inf W = -\frac{1}{2}\mathrm{osc}(V_j)$, so $\|W\|_\infty = \frac{1}{2}\mathrm{osc}(V_j)$, which is the smallest sup-norm any constant shift of V_j can achieve. Because P and Q are both probability measures, the constant contributes nothing to the difference of the two expectations,

$$\sum_o m\,\big(P(o) - Q(o)\big) \;=\; m\,(1-1) \;=\; 0,$$

and therefore

$$B^j_i \;=\; \sum_o V_j(o)\,\big(P(o) - Q(o)\big) \;=\; \sum_o W(o)\,\big(P(o) - Q(o)\big).$$

Bounding the right-hand side by Hölder's inequality and substituting the definition of total variation distance,

$$\big|B^j_i\big| \;\leq\; \|W\|_\infty \sum_o \big|P(o) - Q(o)\big| \;=\; \tfrac{1}{2}\mathrm{osc}(V_j) \cdot 2\,\mathrm{TV}(P, Q) \;=\; \mathrm{osc}(V_j) \cdot \mathrm{TV}(P, Q).$$

The midpoint shift is what makes the bound depend on the oscillation of V_j rather than on its magnitude: without it the same argument would give the weaker $\|V_j\|_\infty$ in place of $\frac{1}{2}\mathrm{osc}(V_j)$.

Part (iii). B^j_i is defined as the difference of two expectations of V_j, one under each law. No sample size appears in that expression. Drawing more rollouts shrinks the Monte Carlo error of an *estimate* of either expectation, but the quantity being estimated, and hence their difference, is unchanged. □

Remark. *The two parts bound different things and can separate. The shift can be at its largest while the bias is zero: if V_j is constant on the observations that ablation moves mass between, then* $\mathrm{osc}(V_j)$ *restricted to that mass is zero and $B^j_i = 0$ regardless of how large* TV *is. In the workflows of this paper the policy does not produce the null action, so the lower bound of part (i) is close to one at every decision point and the bound in part (ii) is driven by reward sensitivity alone. Whether that sensitivity is present is checked rather than assumed: Appendix B.6 reports the realized bias by stratum of decision points.*

A.2 Unbiasedness of the C3 Advantage

We restate and prove Proposition 2 from Section 2.4. Every expectation is taken at the recorded prefix $\tau_{<u}$, of which h_u is a function; since the base run draws $\tau_{<u}$ from its law given h_u and the alternatives are drawn from $\pi_b(\cdot \mid h_u)$ independently of the rest of the prefix, an estimate that is unbiased at the prefix is unbiased for the h_u-conditioned advantage of (1) as well.

Proposition 4 (Unbiasedness, restated)**.** *When the trace is the state at u, with alternatives drawn independently from π_b, rollouts independent across alternatives under D_b, and non-adaptive rollout counts, the C3 advantage $\hat{A}_j$ is, conditional on a_j, an unbiased estimate of $A^*_{D_b}(h_u, a_j) =$*

$Q^{D_b}(h_u, a_j) - V^{D_b}(h_u)$, *the expectation being over the rollouts and the other alternatives at the recorded prefix* $\tau_{<u}$.

Setup. Fix a decision point with history h_u. A group of alternatives at that decision point contains n actions $a_1, \ldots, a_n$ sampled independently from $\pi_b(\cdot \mid h_u)$. Each action a_k receives $c_k \geq 1$ independent rollouts under D_b, producing rewards $\{R_{k,1}, \ldots, R_{k,c_k}\}$. The rollout counts $c_1, \ldots, c_n$ do not depend on any sampled action or reward (non-adaptive allocation). Define:

$$\hat{Q}^{D_b}(h_u, a_k) = \frac{1}{c_k} \sum_{m=1}^{c_k} R_{k,m}, \qquad C = \sum_{k=1}^{n} c_k.$$

Proof. We show $\mathbb{E}[\hat{A}_j \mid h_u, a_j] = A^*_{D_b}(h_u, a_j)$, where the expectation is over rollout outcomes and the sampling of actions a_k for $k \neq j$ at the recorded prefix $\tau_{<u}$; by (2), a rollout reset to $\tau_{<u}$ follows the continuation law of the true state.

Step 1: Action value estimate. Since each $R_{k,m}$ is drawn independently from the terminal reward distribution conditioned on (h_u, a_k) with continuation under π_b, and rollouts for distinct actions are independent:

$$\mathbb{E}\big[\hat{Q}^{D_b}(h_u, a_k) \mid a_k\big] = Q^{D_b}(h_u, a_k).$$

Step 2: Baseline expectation. The baseline for action j is $b_{-j} = \frac{1}{C - c_j} \sum_{k \neq j} c_k \, \hat{Q}^{D_b}(h_u, a_k)$. Taking expectation over rollout outcomes conditional on all actions:

$$\mathbb{E}_{\text{rollouts}}\big[b_{-j} \mid a_1, \ldots, a_n\big] = \frac{1}{C - c_j} \sum_{k \neq j} c_k \, Q^{D_b}(h_u, a_k).$$

Next, take the expectation over the independent sampling of $a_k \sim \pi_b(\cdot \mid h_u)$ for $k \neq j$. Since c_k are non-adaptive (fixed independently of the sampled actions), the factor $c_k/(C - c_j)$ is a constant that passes through the expectation:

$$\begin{aligned}
\mathbb{E}_{a_k, k \neq j}\Big[\frac{1}{C - c_j} \sum_{k \neq j} c_k \, Q^{D_b}(h_u, a_k)\Big] &= \frac{1}{C - c_j} \sum_{k \neq j} c_k \, \mathbb{E}_{a_k \sim \pi_b}\big[Q^{D_b}(h_u, a_k)\big] \\
&= \frac{1}{C - c_j} \sum_{k \neq j} c_k \, V^{D_b}(h_u) \\
&= V^{D_b}(h_u). \qquad (5)
\end{aligned}$$

The second equality uses the definition $V^{D_b}(h_u) = \mathbb{E}_{a \sim \pi_b}[Q^{D_b}(h_u, a)]$.

Step 3: Combining.

$$\begin{aligned}
\mathbb{E}[\hat{A}_j \mid h_u, a_j] &= \mathbb{E}\big[\hat{Q}^{D_b}(h_u, a_j) \mid a_j\big] - \mathbb{E}[b_{-j} \mid a_j] \\
&= Q^{D_b}(h_u, a_j) - V^{D_b}(h_u) \\
&= A^*_{D_b}(h_u, a_j).
\end{aligned}$$

□

A.3 Variance Reduction via Within-Group Comparison

Section 3 introduces within-group variance as a diagnostic metric. Here we derive how the within-group baseline eliminates between-group variation and express the residual variance as a function of the branching factor n and the rollout count c_j.

Decomposition. At a decision point with history h_u, the empirical action value admits the decomposition:

$$\hat{Q}^{D_b}(h_u, a_j) = \underbrace{V^{D_b}(h_u)}_{m(h_u)} + \underbrace{A^*_{D_b}(h_u, a_j)}_{\delta_j} + \underbrace{\hat{Q}^{D_b}(h_u, a_j) - Q^{D_b}(h_u, a_j)}_{\varepsilon_j},$$

where $m(h_u)$ is the history-level mean (constant within a group), δ_j is the true action effect, and ε_j is zero-mean rollout noise with $\text{Var}(\varepsilon_j \mid a_j) = \sigma^2(a_j)/c_j$.

Cancellation of between-group variation. The C3 advantage is:

$$\begin{aligned}\hat{A}_j &= \hat{Q}^{D_b}(h_u, a_j) - b_{-j} \\ &= \big[m(h_u) + \delta_j + \varepsilon_j\big] - \frac{1}{C - c_j}\sum_{k\neq j} c_k\big[m(h_u) + \delta_k + \varepsilon_k\big] \\ &= \delta_j + \varepsilon_j - \frac{1}{C - c_j}\sum_{k\neq j} c_k(\delta_k + \varepsilon_k).\end{aligned}$$

The term $m(h_u)$ cancels exactly because $\frac{1}{C-c_j}\sum_{k\neq j} c_k = 1$. This eliminates the dominant source of variance in multi-task training, where different task instances induce large variation in $V^{D_b}(h_u)$ across groups.

Residual variance. Conditional on all actions (hence all δ_k fixed), the variance of $\hat{A}_j$ arises solely from rollout noise. Since rollouts for different actions are independent:

$$\begin{aligned}\text{Var}(\hat{A}_j \mid \text{actions}) &= \text{Var}(\varepsilon_j) + \frac{1}{(C - c_j)^2}\sum_{k\neq j} c_k^2\,\text{Var}(\varepsilon_k) \\ &= \frac{\sigma^2}{c_j} + \frac{1}{(C - c_j)^2}\sum_{k\neq j} c_k^2 \cdot \frac{\sigma^2}{c_k} \\ &= \frac{\sigma^2}{c_j} + \frac{\sigma^2}{(C - c_j)^2}\sum_{k\neq j} c_k \\ &= \frac{\sigma^2}{c_j} + \frac{\sigma^2}{C - c_j},\end{aligned} \tag{6}$$

assuming homogeneous noise variance $\sigma^2(a_k) \approx \sigma^2$. Under heterogeneous variance, the expression generalizes to $\sigma^2(a_j)/c_j + (C - c_j)^{-2}\sum_{k\neq j} c_k\,\sigma^2(a_k)$.

Equal-allocation case. When $c_k = c$ for all k (uniform budget allocation), $C = nc$ and:

$$\text{Var}(\hat{A}_j \mid \text{actions}) = \frac{\sigma^2}{c} + \frac{\sigma^2}{(n-1)c} = \frac{\sigma^2 \cdot n}{c(n-1)}.$$

For a fixed budget $B_u = n \cdot c$ at the decision point (i.e., $c = B_u/n$):

$$\text{Var}(\hat{A}_j \mid \text{actions}) = \frac{\sigma^2 \cdot n^2}{B_u(n-1)}.$$

The variance is minimized at $n = 2$ (yielding $4\sigma^2/B_u$) and increases monotonically for $n > 2$, since $\frac{\partial}{\partial n}\big[n^2/(n-1)\big] = n(n-2)/(n-1)^2 > 0$ for $n > 2$.

A.4 Estimating Influence from Groups of Alternatives

Section 3 defines influence as the conditional mutual information $I(J; Y \mid h_u)$ between the injected action and the downstream response, where $J \in \{1, \ldots, n\}$ indexes the injected action and Y denotes the downstream response under D_b, the token sequence produced after the injection. Here we derive a plug-in estimator from the groups of alternatives already stored.

Definition. For a group of n alternatives at history h_u, define J as a random variable with distribution $P(J = k) = c_k/C$ (proportional to rollout allocation). The conditional mutual information is:

$$I(J; Y \mid h_u) = H(Y \mid h_u) - H(Y \mid J, h_u),$$

where H denotes Shannon entropy. When the downstream agent's response is deterministic given its full input (i.e., greedy decoding), $Y = f(J, h_u)$ and $H(Y \mid J, h_u) = 0$, so influence reduces to $H(Y \mid h_u)$.

Discrete approximation. The terminal reward $R \in \{0, 1\}$ is a coarsened proxy for Y. The data processing inequality requires that R depends on the upstream action J only through the downstream response Y; formally, $J \to Y \to R$ forms a Markov chain conditioned on h_u. This condition holds when the reward function scores only the final output without access to intermediate actions. Under this condition, $I(J; R \mid h_u) \leq I(J; Y \mid h_u)$, so the reward-level quantity is a lower bound on the response-level influence. For binary R:

$$I(J; R \mid h_u) = H(R \mid h_u) - H(R \mid J, h_u),$$

where

$$H(R \mid h_u) = H\Big(\frac{1}{C}\sum_{k=1}^{n} c_k Q^{D_b}(h_u, a_k)\Big)$$

with $H(p) = -p \log p - (1-p)\log(1-p)$, and

$$H(R \mid J, h_u) = \frac{1}{C}\sum_{k=1}^{n} c_k \, H\big(Q^{D_b}(h_u, a_k)\big).$$

Plug-in estimator. Substituting empirical success rates $\hat{Q}^{D_b}(h_u, a_k)$ from the stored group:

$$\hat{I}(J; R \mid h_u) = H\Big(\frac{1}{C}\sum_{k} c_k \hat{Q}_k\Big) - \frac{1}{C}\sum_{k} c_k \, H(\hat{Q}_k), \tag{7}$$

where $\hat{Q}_k = \hat{Q}^{D_b}(h_u, a_k)$. This estimator reuses the group data already collected for advantage estimation, requiring no additional sampling.

Finite-sample bias. The plug-in entropy estimator is negatively biased for small sample sizes (Paninski, 2003). For binary outcomes with c_k trials per action, the leading-order bias of each conditional entropy term is $-1/(2c_k)$, while the marginal entropy (estimated from the pooled C observations) has bias $-1/(2C)$. Since $\hat{I} = \hat{H}(\text{marginal}) - \hat{H}(\text{conditional})$ and the conditional term is the more negatively biased of the two, the mutual information estimator is positively biased (overestimates dependence). The leading-order residual is:

$$\mathbb{E}[\hat{I}] - I \;\approx\; -\frac{1}{2C} + \frac{1}{C}\sum_{k=1}^{n} c_k \cdot \frac{1}{2c_k} \;=\; \frac{n-1}{2C}.$$

This positive bias means $\hat{I}$ may slightly overestimate $I(J; R \mid h_u)$, but since the true quantity is itself a lower bound on $I(J; Y \mid h_u)$ via the data processing inequality, the combined estimate remains informative.

A.5 Budget Allocation and Reliability

Section 3 states that the noise on an advantage in deeper workflows is set by the budget spent at that decision point and by the return variance rather than by the agent count. This subsection collects the arithmetic behind that claim from the variance result of Appendix A.3 and states what it does and does not determine.

Variance at one decision point. Fix a decision point and let B_u denote the budget available *there*, spread over n alternatives with $c = B_u/n$ rollouts each. Specializing the equal-allocation case of Appendix A.3 to this budget,

$$\mathrm{Var}\big(\hat{A}_j \mid \text{actions}\big) \;=\; \frac{\sigma^2 n}{c\,(n-1)} \;=\; \frac{\sigma^2 n^2}{B_u\,(n-1)}. \tag{8}$$

As a function of n at fixed B_u this is minimized at $n = 2$, where it equals $4\sigma^2/B_u$, and rises for $n > 2$, since $\partial_n\big[n^2/(n-1)\big] = n(n-2)/(n-1)^2 > 0$ there.

Depth divides the budget. Under a fixed total budget spread over the K decision points of an instance, each decision point receives, at one rollout per alternative, $n = B/K$ alternatives, so workflow depth enters only through this division. Two consequences follow directly. Below $n = 2$ there is no contrast at all: the baseline b_{-j} of (4) averages over the peers of a_j within its group, and with a single alternative there are none, so the estimator is undefined rather than merely noisy. And a total budget of $B = 8$ crosses that line once a workflow has 5 or more decision points, which is the case the constant per-decision budget of Section 3 is introduced to avoid.

Why the variance expression is not the objective. Equation (8) is not the objective. It gives the noise on a single advantage estimate, and it is monotone in n above 2, so on its own it would always recommend the smallest admissible branching factor. What it omits is that the estimate is used as a ranking: at $n = 2$ the group offers one comparison and one peer to form the baseline from, and each additional alternative adds both comparisons to be settled and peers to average the baseline over. The two effects run against each other as n grows at fixed B. How the reliability of the ranking depends on n at a fixed budget is therefore an empirical question that (8) does not settle. The return variance σ^2 and the continuation count c of (8) also enter the reliability the additive model implies. Write S for the variance of the alternatives' true values at the decision point and $N = \sigma^2/c$ for the sampling variance of a c-continuation mean; a split-half estimate averages $c/2$ continuations per half, so each half carries noise variance $2N$, and the Pearson correlation between 2 halves whose noise is independent is $S/(S + 2N)$ under an additive model. Table 1 compares this prediction with the measured reliability of the 6 workflows.

A.6 Influence at Three Levels: Reward, Answer, Response

Appendix A.4 derives a plug-in estimator of the mutual information between the injected action and the downstream response, in which that response is coarsened to the terminal reward. The coarsening is what makes the estimator cheap, and it is also what limits what the estimator can see. This subsection states the uncoarsened quantity, and the answer-level quantity between the two, which the bias map of Appendix B.6 reports.

Definition. Let $J \in \{1, \ldots, n\}$ index the alternative injected at decision point u, and let Y be the downstream response itself, the full token sequence produced after the injection rather than any summary of it. *Response-level influence* is the conditional mutual information

$$I(J; Y \mid h_u) \;=\; H(Y \mid h_u) \;-\; H(Y \mid J, h_u), \tag{9}$$

computed over the distribution of Y under D_b. It is the reduction in uncertainty about what is written downstream that is attributable to knowing which upstream alternative it read. The name records the level at which the mutual information is taken: over the response distribution, rather than over the reward.

Relation to the reward-level estimator. The reward-level quantity of Appendix A.4 replaces Y by the terminal reward R. Conditioned on h_u, the chain $J \to Y \to R$ is Markov whenever the reward function scores only the final output, which is the case throughout this paper, so the data processing inequality gives

$$I(J; R \mid h_u) \;\le\; I(J; Y \mid h_u).$$

The reward-level value is therefore a lower bound on the response-level value, and the two are not on a common scale. A binary reward caps its own mutual information at one bit no matter how strongly the upstream choice determines the downstream text, so the reward-level estimator is blind to any change in downstream behavior that does not change whether the answer is right. The three levels have separate names; the reported values are answer-level, and they are not compared with the other two.

Estimation at the answer level. The bias map of Appendix B.6 reports neither of the two quantities above but the one between them: Y is taken to be the downstream final answer, a categorical variable read off the response, and $I(J; Y \mid h_u)$ is estimated by the plug-in of Appendix A.4 over the empirical answer distribution of each group, with the Miller-Madow correction applied to each entropy term, in nats. Because the answer is a function of the response and the reward a function of the answer, the

Table 3: Observed over predicted variance of the rollout noise on a C3 advantage, the observation being half the variance, over groups, of the difference between 2 advantage vectors computed from independent continuations of the same alternatives, and the prediction being $\sigma^2 n^2/(B_u(n-1))$ of Appendix A.5 at the budget behind one vector, with σ^2 estimated from the same cell's continuation returns. Top: the 2-agent workflow on Qwen3-4B over a 3×3 grid of branching factors n and continuations per alternative c, the 2 vectors coming from 2 independent sets of c continuations, so $B_u = nc$; the grid is collected apart from the rows below, so its $c = 4$ column and the 2-agent row measure the same design twice. Middle: the 6 workflows of Section 3 at every branching factor tested, 4 continuations per alternative, on Qwen3-4B, the 2 vectors coming from the odd and the even continuations, so $B_u = nc/2$; K is the number of decision points. Bottom: at $n = 4$, the variance across a decision point's 4 alternatives of their true values (signal, estimated as the variance of their 4-continuation mean returns minus the noise) and the sampling variance of a 4-continuation mean return (noise, the within-alternative variance of the 4 continuations divided by 4), each over the workflow's groups at its opening decision point; their ratio orders the 6 reliabilities of Table 1. The rank correlation of the ratio with n in Section 3 pools the 30 Middle cells with 13 on Qwen3-8B: the 2-agent and 3-agent chains at every branching factor, and the 4-turn, branching and 5-agent workflows at $n = 4$.

n	$c = 2$	$c = 4$	$c = 8$
2	0.92	0.99	1.04
4	0.96	0.94	0.91
8	1.01	1.05	1.03

Workflow	$n = 2$	$n = 3$	$n = 4$	$n = 6$	$n = 8$
2-agent chain, $K = 2$	0.98	1.00	0.93	1.01	1.00
3-agent chain, $K = 3$	1.00	1.00	0.99	1.07	1.06
4-turn workflow, $K = 4$	0.97	1.01	1.05	1.07	1.00
Branching workflow, $K = 4$	0.90	0.93	1.00	0.99	1.03
5-agent chain, $K = 5$	1.00	1.02	0.95	0.98	0.99
10-agent chain, $K = 10$	1.07	0.99	1.03	1.02	0.98

Workflow ($n = 4$)	Signal: variance of true values	Noise: variance of a mean
2-agent chain, $K = 2$	0.077	0.023
3-agent chain, $K = 3$	0.014	0.038
4-turn workflow, $K = 4$	0.031	0.016
Branching workflow, $K = 4$	0.025	0.033
5-agent chain, $K = 5$	0.011	0.041
10-agent chain, $K = 10$	0.003	0.049

chain $J \to Y_{\text{response}} \to Y_{\text{answer}} \to R$ is Markov given h_u, so the answer-level value sits between the reward-level and the response-level values.

B Supplementary Experiments and Reproducibility

The main text states when the trace of a cooperative LLM system is its state, which makes interventional credit assignment exact and lets credit quality be checked against executed counterfactuals. The complete reproducibility materials and full experimental results validating these properties are organized as follows. Appendix B.1 summarizes the main notation in a single reference table. Appendix B.2 specifies hyperparameters and training configuration. Appendix B.3 gives the complete prompt templates for the 2-agent and 3-agent workflows. Appendix B.4 gives every arm of the long-generation run, MAGRPO included, on the 4 main benchmarks and the 2 saturation controls, with the training tokens each arm consumed, and the 3-agent workflow of the 5-seed run. Appendix B.5 describes the 4 deep workflows and the reliability measurements of Section 3. Appendices B.6 and B.7 give the procedures behind Section 4, the ablation-bias map and the learned-critic comparison. Appendix B.8 gives the significance testing procedure behind Table 2.

B.1 NOTATION

Table 4 summarizes the notation used throughout the paper, enabling quick reference without paging back to the definition site. Symbols are grouped by topic.

Table 4: Summary of notation.

Symbol	Description
Problem Setting (Section 2)	
N	Number of agents (Figure 1); in Section 3, Appendix A.5 and Table 1, N is the sampling variance of a c-continuation mean ($c = 4$ in every reported value) and S the variance of the alternatives' true values, not the state S_u of Section 2.3 or the message set S_j of Proposition 1
K	Number of decision points of a workflow
u	Decision point index
h_u	Context the acting role reads at decision point u
a	Action (complete textual message)
τ	Execution trace of one run
$R(\tau)$	Terminal reward (scalar)
$\tau_{<u}$	Trace prefix: everything recorded before decision point u
Method (Section 2.4)	
π_b	Behavior policy: the snapshot of π_θ taken at each iteration and held fixed for the iteration
π_θ	Current learnable policy
D_b	Continuation distribution under π_b
n	Branching factor: alternatives sampled per decision point
c_j	Number of Monte Carlo rollouts for action j
$Q^{D_b}(h_u, a)$	Action value under continuation distribution D_b
$V^{D_b}(h_u)$	State value under continuation distribution D_b
$A^*_{D_b}(h_u, a)$	Target advantage under D_b
$\hat{A}_j$	C3 advantage estimate for action j
b_{-j}	Baseline for action j: the continuation-weighted mean return of the other alternatives
Diagnostics (Section 3)	
$I(J; Y \mid h_u)$	Mutual information between the injected action and Y
J	Index of injected upstream action
Y	Downstream token sequence after the injected action
$H(Y \mid h_u)$	Conditional entropy of downstream response
Training (Section 2.4)	
$r(\theta)$	Importance ratio of PPO, taken per token (Appendix B.2)
ϵ	PPO clipping parameter
$\mathcal{L}(\theta)$	Policy gradient objective (clipped surrogate)
B	Budget: rollouts per instance; B_u is the budget at one decision point, and neither is the ablation bias B_i^j of Proposition 1

B.2 HYPERPARAMETERS AND TRAINING CONFIGURATION

The complete hyperparameter specification for all experiments in Section 5 is listed below. Exact values are provided so that practitioners can replicate results without guessing unreported settings.

Shared Configuration. All methods (C3, MAPPO, MAGRPO) share the fixed settings listed in Table 5.

Table 6 lists the settings specific to C3.

Policy objective as implemented. Each message is one training sample, a triple $(h_u, a_j, \hat{A}_j)$ whose tokens are $a_{j,1}, \ldots, a_{j,|a_j|}$. The importance ratio is taken per token, $r_{j,t}(\theta) = \pi_\theta(a_{j,t} \mid h_u, a_{j,<t}) / \pi_b(a_{j,t} \mid h_u, a_{j,<t})$, and on a micro-batch $\mathcal{M}$ of triples the objective maximized is

$$\mathcal{L}(\theta) \;=\; \frac{\sum_{(h_u, a_j, \hat{A}_j) \in \mathcal{M}} \sum_{t=1}^{|a_j|} \min\big(r_{j,t}(\theta)\, \hat{A}_j,\; \mathrm{clip}(r_{j,t}(\theta), 1-\epsilon, 1+\epsilon)\, \hat{A}_j\big)}{\sum_{(h_u, a_j, \hat{A}_j) \in \mathcal{M}} |a_j|}, \qquad \epsilon = 0.2. \tag{10}$$

Table 5: Training hyperparameters.

Parameter	Value
Base model (trained workflows)	`Qwen3-4B-Instruct-2507`
Starting model	`Qwen3-4B-Instruct-2507` (the released instruct model, no further fine-tuning)
Learning rate (actor)	1×10^{-6}
Learning rate (critic, MAPPO only)	5×10^{-5}
Learning rate schedule	cosine, 3% warmup ratio
Batch size (messages per optimizer step)	256
Rollout batch size (instances)	128
Micro-batch (train / rollout)	64 / 32 (5-seed run); 4 / 8 (long-generation run)
PPO epochs per batch	1
Passes over the training problems	5 (5-seed run); 2 (long-generation run)
PPO clip ratio ϵ	0.2
KL penalty coefficient (init)	0.01
KL target	0.1
KL estimator	`k3`
Prompt limit	2560 tokens
Generation limit	512 tokens (5-seed run); 2048 (long-generation run)
Sampling	temperature 0.7, top-p 0.8, top-k 20
Optimizer	AdamW
Weight decay	0.0
Gradient clipping norm	1.0
Warmup steps	3% of total training steps
Precision	`bf16`
Rollout budget B per instance	8
Training problems (5-seed run)	MATH training split, 7,500 problems, disjoint from every evaluation set
Training problems (long-generation run)	8,096 prompts, the MATH training split and the CMATH validation split, both disjoint from every evaluation set; 5,000 sampled per pass, 2 passes
Reward	1 if the final answer extracted from the last message matches the reference answer, else 0
Model reported (long-generation run)	the model saved at the end of training
Evaluations per trained model	1 per seed (5-seed run, greedy decoding); 5 (long-generation run)
Eval interval	every 5% of training steps (5-seed run); every 25% (long-generation run)
Hardware (5-seed run)	8×NVIDIA A800-80GB PCIe, single node
Hardware (long-generation run)	4×NVIDIA A100-80GB containers

Table 6: C3-specific hyperparameters.

Parameter	Value
Reasoner group per instance	1, with $n = 2$ alternatives, each continued by its 4 Actor alternatives ($c_j = 4$)
Actor groups per instance	2, one per Reasoner alternative
Branching factor n per Actor group	4
Rollouts per Actor alternative c_j	1
Reset strategy	Record the trace prefix $\tau_{<u}$ at each decision point u; reset the system to it and re-run for counterfactual rollouts.
Baseline mode	`loo` (the mean of the other alternatives, the action's own return excluded)
Credit variant	`reward_only` (no critic-based value subtraction)

The advantage of a message is assigned to every token of a_j with no division by the message length; prompt and padding positions are masked out, and the mean runs over all unmasked tokens of the micro-batch. The log-probabilities under π_b and all advantages are computed once per iteration, before any gradient step, and each batch is used for one PPO epoch; π_θ, and with it $r_{j,t}(\theta)$, is re-evaluated on every micro-batch. The KL penalty of Table 5 is estimated with k_3 against the starting model, with a coefficient that starts at 0.01 and is adapted once per iteration toward a target of 0.1. Both roles share the parameters θ and this single objective, with no role-dependent weight.

Baseline Implementation Details. MAPPO uses a centralized critic (`mmBERT-base`, separate model) that receives the concatenation of all agents' observations, with `mappo_normalize_scope=global`, `state_max_len=2560`, and a critic warmup of 64 steps; its per-decision advantages are standardized over the batch and divided by the message's token count. MAGRPO computes group-relative advantages with `baseline=group_mean`, `adv_unit=joint_action`, and $n_{\text{samples}} = 8$. Both baselines draw 8 independent full-episode rollouts per instance, the budget $B = 8$ that C3 spends; all three arms share the 256 messages per optimizer step.

Hardware. The 5-seed run used a single node with 8×NVIDIA A800-80GB PCIe GPUs; the long-generation run ran on cloud containers with 4 A100-80GB GPUs, and the critic comparison on containers with 2 GPUs.

B.3 Workflow Specification

Complete prompt templates for the 2-agent workflow of Section 3 and the 3-agent workflow used in Sections 3 and 4 are reproduced below, enabling exact reproduction of each multi-agent interaction and facilitating adaptation to new workflow designs.

2-agent workflow (Reasoner → Actor). The 2-agent workflow assigns 2 roles to 2 separate model instances. The Reasoner receives the task input and produces a step-by-step solution plan. The Actor receives both the task input and the Reasoner's plan, then produces the final answer.

Reasoner System Prompt (2-agent workflow)

Two LLM agents (Reasoner → Actor) collaborate step-by-step to solve math problems. You are the **Reasoner**: Analyze the original problem, historical actions, and reflection data (if provided) to determine the critical next step. Guide the Actor by providing concise reasoning for the optimal operation.

Configuration: `with_answer=false` (Reasoner does not emit a final answer).

Actor System Prompt (2-agent workflow)

Two LLM agents (Reasoner → Actor) collaborate step-by-step to solve math problems. You are the **Actor**: Execute operations using the original problem, action history, and the Reasoner's guidance. Provide the final answer within `\boxed{}`.

Configuration: `with_answer=true`; `depends_on=[reasoner]`.

3-agent workflow (Reasoner → Actor → Verifier). The 3-agent workflow extends the 2-agent workflow with a third agent. The Reasoner prompt is unchanged. The Actor prompt is modified to defer the final answer to the Verifier:

Actor System Prompt (3-agent workflow)

Three LLM agents (Reasoner → Actor → Verifier) collaborate step-by-step to solve math problems. You are the **Actor**: Produce a candidate step-by-step solution using the original problem, action history, and the Reasoner's guidance. Focus on correctness and leave the final checked answer to the Verifier.

Configuration: `with_answer=false`; `depends_on=[reasoner]`.

Verifier System Prompt (3-agent workflow)

Three LLM agents (Reasoner → Actor → Verifier) collaborate step-by-step to solve math problems. You are the **Verifier**: Review the Actor's candidate solution, cross-check it by alternative methods or by plugging the result back in, and then provide the final checked answer. Output the final answer within `\boxed{}`.

Configuration: `with_answer=true`; `depends_on=[actor]`.

Context Serialization. At each decision point u, the input to the downstream agent is constructed by Python format-string substitution into the role's system prompt. The following placeholders are available:

Context Serialization

`{question}` Original task input (math problem text).
`{context}` `"\n\n"`-joined outputs of all upstream roles in topological order.
`{<role>}` Direct access to a specific upstream role's output (e.g., `{reasoner}`, `{actor}`).

Execution order:
2-agent: Reasoner ← `{question}`; Actor ← `{question}` + `{reasoner}`.
3-agent: Reasoner ← `{question}`; Actor ← `{question}` + `{reasoner}`; Verifier ← `{question}` + `{reasoner}` + `{actor}`.

Missing keys are replaced with the empty string. Substitution occurs once per decision point; outputs are appended to `{context}` for downstream roles.

B.4 Extended Training Results

This subsection collects the long-generation run in full (Table 7): every arm, including MAGRPO's final model and its last model before the divergence, on the 4 main benchmarks, MATH500, Minerva-Math, AMC23 and AIME, and on the 2 saturation controls, CMATH and GSM8K, where the starting model is already within a few points of the ceiling. On the 4-benchmark average the starting model, MAPPO and C3 reach 63.0, 64.2 and 65.3. The training stage consumed 191.9M tokens for MAPPO, 213.6M for MAGRPO and 120.0M for C3, read from the training logs (2 passes over the training problems and 8 terminal runs per instance for every arm); the untrained starting model consumed none.

The 5-seed run also trained the 3-agent workflow, the Reasoner, Actor and Verifier chain of Section 3, under the same settings as its 2-agent workflow; Table 8 reports it with the training tokens of each arm. C3 leads on MATH500 and on the 3-benchmark average, and its training consumed 418.4M tokens against 616.1M for MAPPO and 703.1M for MAGRPO, 32% and 40% fewer.

Table 7: The long-generation run in full on the 2-agent workflow with Qwen3-4B: accuracy (%) under sampled decoding, avg@k with $k = 16$ on AMC23 and AIME (2024 and 2025 merged) and $k = 4$ on the other 4 benchmarks, mean $\pm$ std over 5 evaluation runs of one model, so the spread is evaluation noise, on the 4 main benchmarks and on the 2 saturation controls, CMATH and GSM8K, where the starting model already sits near the ceiling; the last column is the training tokens each arm consumed, in millions, from the training logs. †The last model MAGRPO saved before it diverged, reported for reference.

Method	**MATH500**	**Minerva**	**AMC23**	**AIME**	**CMATH**	**GSM8K**	**Tokens (M)**
Starting model	88.5 ± 0.2	59.2 ± 0.8	75.8 ± 1.3	28.6 ± 0.4	96.2 ± 0.2	93.8 ± 0.1	0
MAPPO	88.9 ± 0.3	59.6 ± 0.5	78.3 ± 0.7	29.9 ± 0.8	96.4 ± 0.1	93.6 ± 0.1	191.9
MAGRPO, final	1.9 ± 0.3	1.0 ± 0.6	1.0 ± 0.1	0.1 ± 0.1	34.4 ± 1.2	2.4 ± 0.3	213.6
step 250†	89.0 ± 0.2	60.0 ± 0.2	76.9 ± 0.4	30.1 ± 0.8	96.4 ± 0.0	93.7 ± 0.1	
C3	89.5 ± 0.1	59.8 ± 0.6	81.1 ± 1.2	30.6 ± 0.6	96.3 ± 0.2	93.8 ± 0.2	120.0

Table 8: The 5-seed run on the 3-agent workflow of Qwen3-4B, the Reasoner, Actor and Verifier chain of Section 3: greedy accuracy (%) under a 512-token generation limit, mean $\pm$ std over 5 training seeds, and the training tokens each arm consumed; best value per column in bold. †Avg. is the mean of MATH500, CMATH and GSM8K; AIME 2025 (30 problems) is reported beside it, not inside it.

Method	MATH500	AIME 2025	CMATH	GSM8K	Avg.†	Tokens (M)
MAPPO	75.8 ± 0.7	6.7 ± 1.4	95.8 ± 0.3	93.4 ± 0.3	88.3 ± 0.3	616.1
MAGRPO	62.1 ± 1.3	7.3 ± 2.5	79.6 ± 0.8	92.9 ± 0.3	78.2 ± 0.5	703.1
C3	$\mathbf{78.6 \pm 0.5}$	$\mathbf{7.8 \pm 2.5}$	$\mathbf{95.9 \pm 0.2}$	$\mathbf{93.5 \pm 0.3}$	$\mathbf{89.3 \pm 0.4}$	**418.4**

B.5 Deep Workflow Specifications and Budget Rules

The depth study of Section 3 uses 4 workflows beyond the 2-agent and 3-agent topologies of Appendix B.3. All of them reuse the context serialization of Appendix B.3 unchanged: each role's prompt is assembled by substitution of the task text and the outputs of the roles it depends on, in topological order, so the trace prefix records the entire input to every decision point.

5-agent chain ($K = 5$). A linear chain of 5 roles, Planner $\rightarrow$ Solver $\rightarrow$ Critic $\rightarrow$ Reviser $\rightarrow$ Verifier, each depending on its predecessor. The Planner produces a solution plan without solving; the Solver executes it; the Critic lists concrete defects without rewriting the solution; the Reviser corrects the defects; the Verifier cross-checks and emits the final answer. Only the Verifier emits an answer in `\boxed{}`.

10-agent chain ($K = 10$). A linear chain of 10 roles: Reader $\rightarrow$ Planner $\rightarrow$ Decomposer $\rightarrow$ SolverA $\rightarrow$ SolverB $\rightarrow$ Integrator $\rightarrow$ Critic $\rightarrow$ Reviser $\rightarrow$ Checker $\rightarrow$ Verifier. The Reader restates the problem, its givens, and what is asked; the Planner produces an ordered plan; the Decomposer splits that plan into 2 work packages, an opening package A and a remaining package B; SolverA executes package A and SolverB continues from its partial working to execute package B; the Integrator merges the 2 partial workings into one coherent candidate solution; the Critic lists concrete defects; the Reviser corrects them; the Checker cross-checks by an independent route and reports whether the check passes; the Verifier resolves any remaining discrepancy and emits the final answer.

4-turn workflow ($K = 4$). The Reasoner and the Actor alternate twice (Reasoner, Actor, Reasoner, Actor), so that the agent count is held at the 2-agent value while decision depth rises to 4. This workflow separates temporal depth from cross-agent depth: it is the control for reading the 3-agent result, since the 2-agent and 3-agent workflows differ in agent count and depth at once.

Branching workflow ($K = 4$). A Planner writes the plan; 2 Solvers work from the task text and the plan in parallel, neither seeing the other; an Integrator reads both workings and emits the final answer. It is the one workflow in this paper whose wiring is not a chain: the 2 Solver decision points share a prefix and are re-run independently of each other, and the Integrator's context is assembled from both, so the trace prefix of every decision point is still the whole of what that role reads.

The two quantities. Two quantities are reported per workflow in Table 1, both computed from the groups of alternatives already collected for advantage estimation and both on a policy held fixed, so that they describe the estimator rather than a training outcome. Every workflow is measured at its first decision point (the Reasoner, Planner or Reader), on the released instruct model held fixed, with a 512-token limit per message.

- *Split-half reliability.* Within a group, the rollouts of each alternative are partitioned into 2 halves, the even-numbered and the odd-numbered continuations, the same halves as in Table 3. Each half gives its own mean return per alternative and therefore its own advantage vector over the n alternatives of the group (Equation (4)). The split-half reliability of the group is the Spearman rank correlation between the 2 vectors, and the reported value is the mean over groups. Computing the same statistic as a Pearson rather than a Spearman correlation changes no workflow's value by more than 0.003, and the Pearson values lie within 0.04 of $S/(S+2N)$ as well: returns are 0 or 1 and each half averages 2 continuations, so the two statistics coincide on most groups. The quantity needs no separate reference rollouts, which is what makes it comparable across workflows of different depth. Groups with fewer than 3 alternatives, and groups in which either half assigns the same advantage to every alternative (for example because every rollout of that half returned the same reward) have no ranking to compare and are excluded from the mean; the statistic therefore describes the decision points at which the outcome varies across alternatives, and the excluded fraction grows with the share of instances the policy either always solves or never solves. Under the fixed budget it is 35% and 50% at 2 and 3 decision points (at 4 decision points, the fixed budget leaves 2 alternatives per group, below the 3 alternatives the statistic needs, so no value is defined there). Because subtracting the mean of the other alternatives is an increasing affine map of the per-alternative mean return within a group, the rank correlation is that of the per-alternative mean returns themselves.
- *Within-group variance.* The sample variance of the advantages across the alternatives of a group, averaged over groups, as defined in Section 3.

Two budget rules. Under a *fixed total budget*, one budget B is spent per instance regardless of depth, so each decision point receives a branching factor of $n = B/K$ (Section 3); C3's training runs in Section 5 spend their $B = 8$ differently, as one 2×4 tree whose 8 Actor rollouts are also the Reasoner's continuations (Table 6). At $K = 4$ with $B = 8$ this leaves $n = 2$, the smallest value at which another alternative exists to compare against, and at 5 or more decision points the contrast is undefined. Under a *constant per-decision budget*, the budget scales with depth as $B = 4K$, holding $n = 4$ at every decision point: $B = 16$ for the 4-turn and branching workflows, $B = 20$ for the 5-agent chain, and $B = 40$ for the 10-agent chain. Both rules count alternatives, at one rollout each; the measurements then give every alternative 4 continuations, so that the split-half statistic is defined. Under the fixed budget the 4-turn and branching workflows would have 2 alternatives per group, and the deeper chains fewer, below the 3 alternatives the statistic needs, so Table 1 reports every workflow at the constant per-decision budget and Figure 3 sweeps the branching factor.

Model scale. Reliability is recomputed on Qwen3-8B: split-half reliability 0.71 for the 2-agent workflow and 0.34 for the 3-agent workflow under the fixed total budget $B = 8$, and 0.71 and 0.26 under the constant per-decision budget.

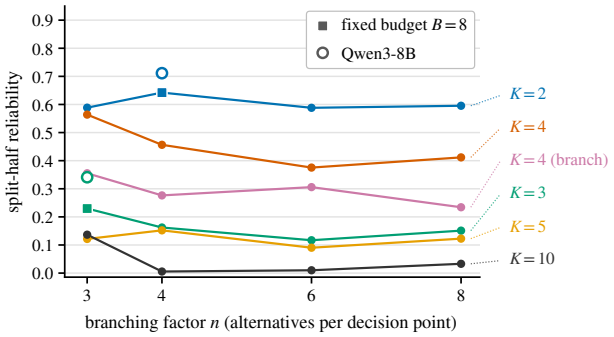


Figure 3: Split-half reliability on a policy held fixed against the branching factor n, one line per workflow labeled with its number of decision points K; every alternative receives 4 continuations so that the statistic is defined. Filled squares: the branching factor an even split of 8 alternatives over the decision points gives, 4 on the 2-agent chain and 3 on the 3-agent chain (8 over 3, rounded). Open circles: the same fixed-budget points on Qwen3-8B.

B.6 Ablation Bias at Each Decision Point

Proposition 1 bounds the ablation bias by $\mathrm{osc}(V_j) \cdot \mathrm{TV}$ but does not say how much of that bound is realized. That is an empirical question, and this subsection gives the procedure behind Table 9.

Sample. The sample is 150 decision points under the 3-agent workflow on MATH500, the Reasoner position of the 150 questions at the head of a fixed evaluation order, one per question, so that the question is the independent unit for the paired test reported below. Each point carries 4 alternatives with 4 rollouts each, plus the null-action arm. Those rollouts are the ones already stored for advantage estimation; the null-action arm is the only addition.

Two operationalizations of the null action. Ablation has to be turned into something a language model can be run on, and there are two natural choices. The null action $a_\emptyset$ is either the deletion of the role's entire segment from the downstream context (in our context assembly an empty message and a deleted segment produce the same downstream prompts, so this is the empty-message form), or a fixed placeholder message that keeps the role present in the context but contributes nothing. The two are not interchangeable in principle: one erases the fact that the role acted at all, the other leaves it present but silent. Both were run on the same decision points; a paired test between them gives $p = 0.92$, and the results below are reported under the empty-message form, which is the counterfactual CCPO (Li et al., 2026d) subtracts.

Per-decision-point bias. At each decision point the computed quantity is the mean return over the alternatives actually sampled from π_b at that point, minus the return obtained under the null action, both estimated from the stored rollouts of that decision point. This is the orientation of Proposition 1, which defines B_i^j as the continuation value with the agent present minus the value with it ablated, so a positive value means that ablation costs return.

Two diagnostics related to the bound. Two diagnostics stratify the realized bias. For reward sensitivity we use the empirical oscillation of the estimated action values within a group of alternatives,

$$\hat{\Delta}_{Q,u} \;=\; \max_k \hat{Q}^{D_b}(h_u, a_k) \;-\; \min_k \hat{Q}^{D_b}(h_u, a_k),$$

which is a within-group spread over the sampled alternatives at u. It is used only to split decision points into a stratum where the estimated action values differ and a stratum where they do not; it is not an estimator of the population quantity $\mathrm{osc}(V_j)$, and it is not reported as one. For downstream dependence we use the answer-level plug-in influence of Appendix A.6: the conditional mutual information of Section 3 with Y taken to be the downstream final answer, estimated on the same groups. Its mean over these groups is 0.36 nats.

Stratification. The 150 points are split at the median influence into a high half and a low half. Within the high half, the points are further split by whether $\hat{\Delta}_{Q,u}$ is nonzero (52 points) or zero (23

Table 9: Realized agent-ablation bias, stratified by reward sensitivity within the high-influence half of the sample (answer-level plug-in influence, Appendix A.6). The sample is 150 decision points of the 3-agent workflow on MATH500, one per evaluation question; the high-influence half is the half above the median influence, and the two strata split those points by whether the sampled alternatives received different measured values ($\hat{\Delta}_{Q,u} > 0$). The two operationalizations of the null action $a_\emptyset$ (an empty message and a fixed placeholder message) differ by no detectable amount (paired $p = 0.92$).

Stratum	Points	Mean bias	Median	IQR	Exactly zero
Alternatives differ in value ($\hat{\Delta}_{Q,u} > 0$)	52	$+0.02$	$+0.06$	$[-0.08, +0.19]$	3 / 52
Alternatives equal in value ($\hat{\Delta}_{Q,u} = 0$)	23	-0.01	$+0.00$	$[+0.00, +0.00]$	22 / 23

points). No threshold uses a bias value. Table 9 reports the two strata: 96% of the points whose alternatives received equal measured values have exactly zero bias, against 3 of the 52 points with differing values (the points with equal measured values are almost all instances that every rollout solves or fails, where both arms return the same reward). The low-influence half carries a mean absolute bias of 0.172.

Aggregate and concentration. The 52 points that are high on both factors, 35% of the sample, carry 44% of the total absolute bias over the 150 points, a mean absolute bias of 0.20 against 0.135 for the remaining 98 points.

B.7 Learned-Critic Comparisons

C3 is compared with 2 learned critics, at 2 different granularities. The distinction matters, because the two answer different questions and only one of them is a competitor for per-decision credit.

State-value critic (already in the baselines). MAPPO carries a centralized state-value critic $V(h_u)$ over the concatenated observations of all agents (Appendix B.2). As a credit estimator it has a structural limit that no training budget lifts: a state-value critic assigns one number to a decision point, so subtracting it shifts every alternative at that decision point by the same constant and leaves their relative order untouched. It therefore adds no within-group information beyond the returns of the rollouts it is given; what it changes is the between-group centering, and MAPPO's own training builds no shared-history groups at all.

Decision-value critic (trained for this comparison). The estimator that does compete at the right granularity is a decision-value critic $\hat{Q}(h_u, a)$ that scores each alternative individually. It is a regression head on a small pretrained encoder (mmBERT-base) that reads the decision point's context and the alternative as one sequence of up to 1024 tokens, and it is trained on the rollout data C3 already collects: the groups of the 2-agent workflow of Section 3 on the policy held fixed, 500 decision points with 4 alternatives and 4 continuations each, of which 400 decision points train the critic and 100 are held out, split by decision point so that no alternative of a held-out point enters training. The target is the mean return of an alternative's continuations. The family is the learning rate in $\{10^{-5}, 3 \times 10^{-5}, 10^{-4}\}$ crossed with 2 heads, a linear output and one hidden layer of 768 units; each of the 6 is trained for 3 epochs (AdamW, batch size 16, weight decay 0.01), the one with the lowest validation loss on a fifth of the training questions is retrained on all training points, and the selection never sees the held-out rank correlations.

Evaluation. Both estimators are scored on the held-out decision points against a Monte Carlo reference of 16 fresh continuations per alternative, drawn under a seed different from the run's own, which is unbiased under the condition of Section 2.3. The metric is the *rank correlation on held-out decision points*: the Spearman rank correlation between an estimator's ordering of the 4 alternatives at a decision point and the reference ordering, averaged over the points at which the reference and both estimators order the alternatives, 79 of the 100 held-out points (a point drops out when the reference or an estimator assigns all 4 alternatives the same value, the rule of the split-half statistic in Section 3). It is not the split-half reliability of Section 3, which uses the rollouts themselves as their own reference. C3's estimate at a held-out point is the advantage from the 4 continuations the run already had, so the two estimators are compared on the same points against a reference independent of both.

Result. The decision-value critic reaches a rank correlation of 0.29 on the held-out decision points; C3 on the same points reaches 0.69 from 4 continuations per alternative. The paired difference is $+0.40$ at $p < 0.001$. A second cell of 16 fresh continuations per alternative on the same held-out alternatives, drawn under a third seed, agrees with the reference at 0.73 on 88 points, which is the agreement 2 independent draws of the reference have with each other. On the 75 held-out points that the reference, C3 with 4 continuations and the second 16-continuation reference at 4, 8 and 16 continuations all order, C3 reaches 0.70 and the 2 references agree at 0.76. Retrained with the same settings on 2 and 4 times the rollouts, the critic stays well below C3's agreement with the reference: over the three data sizes the upper end of its 95% bootstrap interval over held-out points reaches 0.45 at most, while the interval of C3 from 4 continuations starts at 0.59. Trained on the same rollouts, the critic falls well short of ranking alternatives within a decision point, which is the operation per-decision credit assignment consists of.

B.8 Significance Testing Procedure

An average over benchmarks can be moved by one benchmark, so the MATH500 gain of the 5-seed run, the one the introduction quotes, is tested on its own with a paired test. This subsection gives the procedure. The long-generation run carries no test: Table 7 reports one trained model per arm and the spread of its 5 evaluation runs.

Pairing and sample. The comparison is C3 against MAGRPO, the baseline with the highest MATH500 mean on the 2-agent workflow of the 5-seed run. The test runs on the records behind Table 2, paired by seed index over the 5 training seeds the per-seed records carry for both arms.

Test. The paired differences (C3 minus the strongest baseline, in percentage points) are tested against zero with a paired two-sided t test, and the reported interval is the corresponding 95% confidence interval on the mean difference. The two-sided form is used throughout: the test is asked whether the two methods differ on that benchmark, not whether C3 is ahead.

Multiplicity. The 4 benchmarks tested at once, the MATH500, AIME 2025, CMATH and GSM8K columns of Table 2, inflate the chance that one of them clears an uncorrected threshold, so the 4 raw p values are corrected by the Holm step-down procedure within the family. Holm sorts the raw values in ascending order and scales the k-th smallest of m by $m - k + 1$, taking a running maximum so that the corrected values stay ordered. It controls the family-wise error rate under arbitrary dependence between the tests, which matters here because the 4 benchmarks share the same training runs and their differences are not independent. The MATH500 difference is $+8.28$ pp, 95% interval $+7.48$ to $+9.08$, at a Holm-corrected $p < 0.001$; Table 10 lists all 4 differences with their intervals and both p values.

Table 10: The 5-seed run: paired differences, C3 minus MAGRPO, on the 4 benchmarks tested as one family, in points, with 95% intervals and raw and Holm-corrected p values over $n = 5$ training seeds. CMATH and GSM8K are the saturation controls of Appendix B.4, on which the starting model already sits near the ceiling; AIME 2025 has 30 problems.

Benchmark	Difference	95% interval	Raw	Holm
MATH500	$+8.28$	$+7.48$ to $+9.08$	$p < 0.001$	$p < 0.001$
AIME 2025	$+2.67$	-0.80 to $+6.13$	$p = 0.10$	$p = 0.30$
CMATH	$+0.26$	-0.19 to $+0.70$	$p = 0.19$	$p = 0.38$
GSM8K	$+0.09$	-0.33 to $+0.51$	$p = 0.58$	$p = 0.58$

Training tokens. The token savings quoted in the abstract come from the long-generation run on the 2-agent workflow: Appendix B.4 lists the total training tokens of every arm from its training log (2 passes over the training problems and 8 terminal runs per instance for every arm): 120.0M for C3 against 191.9M for MAPPO, a saving of 37%.

B.9 What Each Credit Signal Restores, Re-runs and Compares

Table 11 places the credit signals of Section 7 side by side at the level of mechanism: the unit that receives a number, what is restored before an alternative is valued, how its value is obtained, and

what that value is compared against. Of these, C3 alone values every sampled alternative at every role of a team by executing its continuation to the terminal reward and compares it with the other alternatives sampled at the same decision point; the single-agent methods re-run from a prefix for one policy, and the multi-agent methods either predict the value, share one score, remove an agent or a message, or score a candidate without continuing it.

Table 11: Mechanisms of the credit signals compared in Section 7: what each restores before it values an alternative, how it obtains the value, and what it compares the value against.

Method	**Unit credited**	**Restored before valuing**	**Value obtained by**	**Compared against**
MAPPO (Yu et al., 2022)	each decision point	nothing: the value is predicted	a centralized critic on the concatenated observations	the critic's value of the state at that decision point
MAGRPO (Liu et al., 2025)	every decision of a trajectory, the same number	nothing	the trajectory's terminal score	the mean score of a group of trajectories
CCPO (Li et al., 2026d)	one agent of a 2-agent chain	the downstream agent's prompt with the ablated message removed	re-running the downstream agent to the reward	the reward with the agent present (Proposition 1)
MALT (Motwani et al., 2024)	each role's output	the prefix, branched into a tree of sampled continuations	the outcomes below the output, turned into labels	a label threshold; training is off-policy on the labels
VinePPO (Kazemnejad et al., 2024)	a reasoning step of one agent	the step's prefix	Monte Carlo completions from the prefix	the Monte Carlo value of the preceding step
Tree-GRPO (Ji et al., 2025)	a step of one agent	the step's prefix, branched into a tree	the terminal outcomes of the branches	the group mean over the tree's branches
ROME (Wang et al., 2025)	a tool-call chunk of one trajectory	nothing	the trajectory's terminal return, discounted to the chunk	no per-decision baseline
AT-GRPO (Zhao et al., 2026)	one agent's candidate at one turn	the same context, with K candidates sampled	an immediate verifiable reward of the candidate; only the best candidate is continued	the mean over the K candidates
Re-run without one message (Yang et al., 2026)	one message	the trajectory without that message	re-running the trajectory	the reward with the message present
C3 (this paper)	a decision point of any role	the trace prefix, the state under Definition 1	c_j continuations of every downstream role to the terminal reward, for each of n sampled alternatives	the continuation-weighted mean of the other alternatives at the same decision point