COUNTERFACTUAL EDIT CREDIT FOR LONG-HORIZON SOFTWARE AGENTS

Anonymous Authors

Draft: policy-effectiveness measurements are not yet available.

ABSTRACT

Repository-repair agents produce long interaction traces, while their tests usually score only the completed patch. Counterfactual edit credit connects these objects through an executable intervention on the final artifact. The method selects one surviving edit block without consulting the verifier outcome, restores its touched paths to their pre-block contents, and tests the resulting patch in a fresh environment. The signed change in test utility becomes an additive advantage on the generated tokens of the turns that produced the block. This construction requires at most one additional counterfactual verifier trial per eligible rollout and no additional policy continuation. A broadcast control receives the same signed and absolute auxiliary coefficient mass on a common reference trajectory, enabling a controlled comparison of allocation rules. We also specify a terminal-only control and a terminal-advantage edit-focus control that uses no counterfactual score. The accompanying protocol fixes training endpoints and separates tasks used for reinforcement learning from its evaluation complement. On 320 development trajectories, 244 removal queries are usable and 92 yield a nonzero contrast, including 17 from failed originals. These measurements support signal availability; policy-effectiveness measurements remain pending.

1 INTRODUCTION

A software agent can spend many turns inspecting a repository, proposing edits, running commands, and revising its patch before receiving a final test verdict. A trajectory-level reward summarizes the completed artifact, but does not identify which surviving edits contributed to that outcome. This distinction matters even for unsuccessful trajectories: an incomplete repair can preserve useful changes alongside changes that break other behavior. Repository benchmarks provide executable tests (Jimenez et al., 2023), making it possible to ask a more specific question about the artifact the agent produced.

Our question is whether a single, directly tested intervention can supply useful feedback to the turns that produced an edit. Existing approaches already provide local feedback through learned critics, execution-derived reward models, evaluator diagnostics, or policy continuations (Zhou et al., 2025; Li et al., 2026; Jin et al., 2026; Kazemnejad et al., 2025). The opportunity considered here is narrower: use the completed repository itself to construct one removal experiment, without fitting an additional scoring model or sampling how the policy would continue after an intervention.

Counterfactual edit credit selects a block of consecutive editing turns whose touched paths are not edited later. Starting from the final repository, it restores those paths to their contents immediately before the block and retains every other final file. Testing the resulting artifact yields a signed utility difference. A positive difference means that removing the selected file effects worsens this completed artifact; a negative difference means that the removal improves it. The method adds this signal to the advantages of the producing turns, including their reasoning and tool-call tokens.

The intervention and its allocation are distinct design decisions. An improvement over terminal-only training could come from additional test information, from its concentration, or from where that concentration is placed. We therefore define a broadcast control with the same query and eligibility rules. On any shared reference rollout, broadcast and localized credit have equal signed and absolute

coefficient mass before the policy loss. A second control concentrates the ordinary terminal group advantage on the selected turns without using a counterfactual score. These comparisons distinguish the proposed allocation from two concrete alternatives; they keep its information source and optimizer effects explicit.

The contribution developed in this draft has three parts:

1. an executable final-artifact intervention that removes one surviving edit block with at most one additional verifier trial;
2. an additive turn-credit construction and a pointwise, coefficient-mass-matched broadcast control;
3. a fixed experimental protocol comparing those rules with terminal-only and terminal edit-focus training, supported by factual replay and implementation checks.

The central empirical question is whether localized credit improves held-out repair performance relative to broadcast delivery of this signal. Section 3 defines the intervention and coefficients; Sections 4 and 5 specify the comparisons and evaluation units.

2 Related Work

Counterfactual utility and executable reduction. Difference-reward constructions precede language-model agents: wonderful-life utility subtracts utility after clamping part of a realized system (Wolpert & Tumer, 2002). COMA instead marginalizes one agent's action through a centralized critic while holding others fixed (Foerster et al., 2018). In software, delta debugging repeatedly tests reduced inputs or differences to isolate failure-inducing elements (Zeller & Hildebrandt, 2002). Our adaptation uses one selected final-artifact intervention to produce a training coefficient and maps it to the recorded producing turns. It does not perform iterative minimization, estimate an action-marginal baseline, or inherit guarantees from effect-set clamping.

Execution feedback and multi-turn learning. SWE-bench evaluates repository changes against task-specific tests (Jimenez et al., 2023). Such executable feedback supplies an outcome for a complete repair, whereas optimization must act on the sequence of decisions that produced it. Group-based policy optimization offers a simple outcome-based starting point: the reward of a sampled solution is compared with rewards from the same prompt group (Shao et al., 2024). This establishes a training signal for the sampled response, but leaves a separate question about allocation within a long interaction. SWEET-RL studies collaborative, multi-turn tasks and trains a critic that can use additional information available during training (Zhou et al., 2025). Its local feedback is learned from that information. Counterfactual edit credit instead obtains a particular local signal by executing a modified completed artifact. The distinction concerns how the feedback is constructed, rather than whether multi-turn agents can already receive intermediate credit.

Local feedback for generated software. BoostAPR combines execution-verified demonstrations with sequence-level and line-level reward models, then uses the latter to allocate repair feedback to edit regions (Li et al., 2026). This makes execution-derived local credit a direct precedent for software training. RCCA connects functional rubric feedback for interactive applications to responsible source spans and generated tokens (Jin et al., 2026). Both motivate a distinction between assessing an artifact and assigning optimization weight inside its generation. Our construction uses a different assessment: the difference between the final repository and an artifact with selected file effects removed. Its receiving region is also different. We allocate to the complete generating turns of an edit block, rather than asserting a precise mapping from a diagnostic or reward-model prediction to individual code tokens. This choice keeps the intervention and its recorded producer aligned at the granularity supported by the trace.

Separating credit magnitude and allocation. FACTOR explicitly separates action-credit estimation from token allocation. Its action credits use checkpoint-calibrated temporal differences, while feedback-conditioned likelihood gaps distribute those credits within actions (Ma et al., 2026). CoRT contrasts the likelihood of a realized response under rubric-conditioned and criteria-free prompts, then uses bounded, normalized token weights to redistribute the response advantage (Zhang et al.,

2026). The conserved quantity and the loss reduction matter. FACTOR normalizes within each action and couples that normalization to an action-mean surrogate. Its checkpoint-calibrated action credits telescope to the trajectory advantage, with sparse continuation estimates used for calibration. Our equality instead concerns the trajectory-level sum of an additive coefficient over generated positions. We retain prompt-mean reduction and add the terminal advantage before DPPO masking, whereas FACTOR preserves each action's own TD credit through within-action normalization and an action-mean reduction. These constructions therefore control different quantities even though both conserve a form of credit. The broadcast control here uses the same removal utility and cap shrinkage as localization on a shared trace, so its comparison is tied to this exact estimator and reduction. Conservation alone cannot transfer an optimization interpretation from one construction to the other.

These works establish that conserving or redistributing a scalar credit is not itself a new contribution. Our comparison uses one artifact-derived *additive* signal with two receiving masks. Because the signal is added to the original group advantage, it can remain nonzero when the terminal group advantage vanishes. The shared mass constraint controls one aspect of this comparison; it leaves concentration, sign interactions with terminal advantages, and subsequent trust-region masking visible in the analysis.

Continuation-based and reversible execution. VinePPO estimates intermediate values from Monte Carlo continuations, providing a route to credit assignment without a learned value network (Kazemnejad et al., 2025). Shepherd represents agent execution as a reversible trace and supports fork-based operations; its training application uses forks to obtain alternative rollout feedback (Yu et al., 2026). These approaches can inspect what follows an intermediate state under further execution. Our intervention holds the other final files fixed and changes only the selected block's surviving file effects. It therefore measures a conditional artifact contrast, not the expected outcome of letting the agent continue after omitting an action. The absence of an additional policy continuation is a property of this construction. This ties the feedback to the final file effects and their recorded producing turns, with verification cost accounted for separately.

3 Counterfactual Edit Credit

Figure 1 shows the artifact intervention and the two coefficient allocations on a shared reference rollout.

3.1 From interaction turns to surviving edits

A rollout τ_i contains assistant turns interleaved with tool observations and ends in a repository patch P_i. We record a repository tree after each turn and the paths changed since the previous tree. An *edit block* is a maximal consecutive sequence of turns with nonempty path changes. The tree immediately before the block and the tree after its last turn define its net file effects.

A block is eligible if its net effect is nonzero, it touches no test paths under the rule in Section B, and none of its touched paths is modified by a later turn. "Surviving" refers to this last-touch condition: a block can also undo an earlier mistake and leave no difference from the task baseline on those paths. One eligible block is selected by the smallest SHA-256 hash of the rollout key and block index, without consulting verifier outcomes. An unavailable selection is retained rather than replaced. Section B specifies snapshots, key fields, and token masks.

3.2 A conditional final-artifact intervention

Let e_i be the selected block and $S(e_i)$ its touched paths. The intervention artifact $P_i^{-e_i}$ is constructed from the final tree by restoring each path in $S(e_i)$ to its pre-block contents, including its prior absence when the block created that path. All other final files remain fixed. The implementation constructs this tree directly and produces a patch against the task baseline. It does not use a fuzzy reverse-patch fallback.

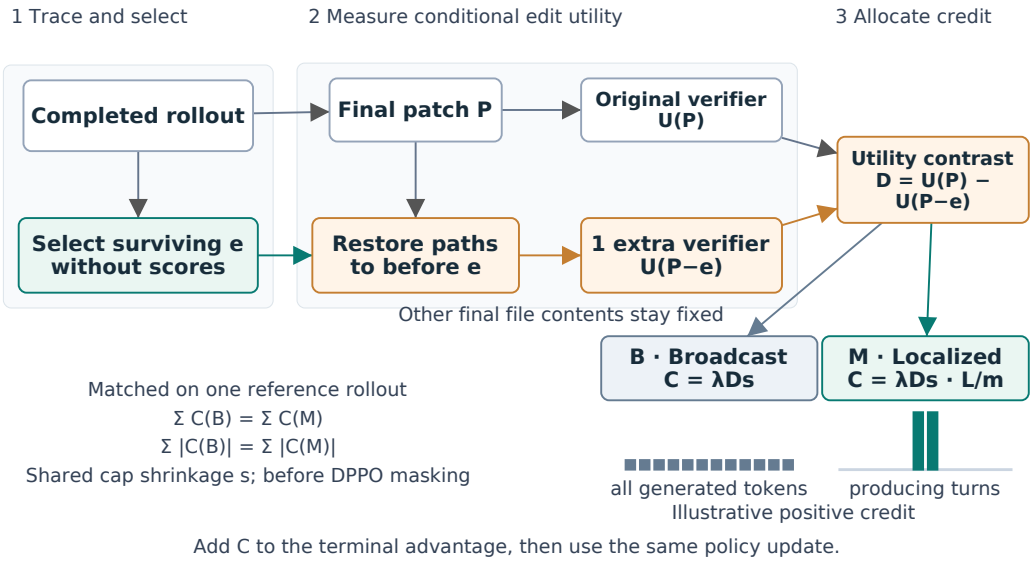


Figure 1: **Counterfactual edit credit on one reference rollout.** Restoring the selected block's paths to their pre-block contents produces an artifact for one extra verifier call. Its utility contrast supplies an additive coefficient: B distributes it over all generated tokens; M concentrates it on the producing turns with shared cap shrinkage. The illustrated positive bars have equal signed and absolute coefficient mass, not necessarily equal gradient influence. They are a schematic on one shared trajectory, not experimental measurements.

The original and intervention artifacts are assessed in matched task contexts. For a test report, define

$$U(P) = Y(P) + \tfrac{1}{4}F(P) - \tfrac{1}{4}G(P), \qquad D_i = U(P_i) - U(P_i^{-e_i}). \tag{1}$$

Here Y is the resolved verdict, F is the fraction of a fixed, deduplicated fail-to-pass test list that passes, and G is a binary indicator that any expected pass-to-pass test does not pass. Missing or skipped expected tests are non-passing in both lists. An empty expected fail-to-pass list, or a missing or unparseable report, makes the utility unavailable rather than zero. An empty expected pass-to-pass list gives $G = 0$. Under these definitions, $U \in [-1/4, 5/4]$ and $D_i \in [-3/2, 3/2]$.

Only the intervention requires an additional counterfactual (CF) verifier trial: the original rollout already receives its ordinary final verification. Each query-eligible rollout issues at most one such trial, with no additional policy continuation. A failed query gives no auxiliary credit and remains in the availability denominator. The measured contrast fixes the other final files; it consequently describes the contribution of these file effects conditional on the completed artifact.

3.3 Localized and broadcast coefficients

Let $\mathcal{G}_i$ contain the valid generated-token positions in the flattened rollout, excluding observations and padding, and let $L_i = |\mathcal{G}_i|$. The selected block's producing turns define $\mathcal{E}_i \subseteq \mathcal{G}_i$, with $m_i = |\mathcal{E}_i|$. For a block spanning several turns, this is the union of those turns' generated spans. Every selected span must be present and complete after truncation; a missing, clipped, duplicated, or masked span makes the selection unavailable.

For an available query with $L_i, m_i > 0$, coefficient scale $\lambda > 0$, and token cap $c > 0$, define

$$s_i = \begin{cases} 1, & D_i = 0, \\ \min\left(1, \dfrac{c\, m_i}{\lambda |D_i| L_i}\right), & D_i \neq 0. \end{cases} \tag{2}$$

The localized arm M and broadcast arm B use

$$C_{it}^{M} = \lambda D_i s_i \frac{L_i}{m_i} \mathbf{1}[t \in \mathcal{E}_i], \qquad C_{it}^{B} = \lambda D_i s_i \, \mathbf{1}[t \in \mathcal{G}_i]. \tag{3}$$

When the cap is active, these coefficients simplify to $c\,\mathrm{sign}(D_i)$ on M's receiving tokens and $c\,\mathrm{sign}(D_i)m_i/L_i$ on B's. The magnitude of D_i and the scale λ then cancel: delivery depends on sign and support size. Both are zero when the selection or utility is unavailable, the trajectory is excluded from the policy loss, or there is no valid receiving token. Format-error trajectories receive no auxiliary credit. The terminal reward rule is shared across arms.

On a common reference rollout, summing Equation (3) gives

$$\sum_t C_{it}^M = \sum_t C_{it}^B = \lambda D_i s_i L_i, \qquad \sum_t |C_{it}^M| = \sum_t |C_{it}^B| = \lambda |D_i| s_i L_i. \tag{4}$$

These are identities of the raw auxiliary coefficients. For nonzero credit, their squared masses have ratio $\sum_t (C_{it}^M)^2 / \sum_t (C_{it}^B)^2 = L_i/m_i$. Thus mass matching controls the total signed and absolute signal without imposing the same concentration.

3.4 Policy optimization and terminal edit focus

For a prompt group of n sampled rollouts, let V index members retained by the common trajectory masks. The shared terminal group advantage is

$$A_i = \begin{cases} r_i - \dfrac{1}{|V|}\sum_{j\in V} r_j, & i \in V, \\ 0, & i \notin V, \end{cases} \qquad \widetilde{A}_{it} = A_i + C_{it}. \tag{5}$$

The common terminal reward r_i includes the format-error and pass-to-pass regression rules specified in Section 4. An entirely excluded group contributes zero. Substituting the valid-member mean for excluded rewards before centering gives the same valid-member advantages. Group advantages are not divided by their standard deviation. We add the auxiliary tensor after group centering, leave terminal returns unchanged, and use Divergence Proximal Policy Optimization (DPPO) (Qi et al., 2026). For N_g prompt groups, with T_g valid generated tokens in group g, the policy surrogate is

$$\mathcal{L}(\theta) = -\frac{1}{N_g}\sum_{g=1}^{N_g} \frac{1}{\max(1, T_g)} \sum_{i\in g}\sum_{t\in\mathcal{G}_i} \rho_{it}(\theta)\, q_{it}(\theta, \widetilde{A}_{it})\, \widetilde{A}_{it}, \tag{6}$$

where ρ_{it} is the current-to-behavior token likelihood ratio and q_{it} is the stop-gradient binary-TV mask defined in Section B. This prompt-mean reduction averages tokens within each prompt group, then groups; it is not an equal-weight mean of per-rollout means. The combined advantage's sign can affect q_{it}, so Equation (4) does not guarantee equal realized gradient norms or masking rates.

The terminal-only arm O sets $C_{it} = 0$. The terminal edit-focus arm E uses the same structural selection and receiving mask, but derives its auxiliary coefficient solely from the original terminal group advantage:

$$C_{it}^E = \mathrm{clip}(\lambda A_i L_i/m_i, -c, c)\,\mathbf{1}[t \in \mathcal{E}_i]. \tag{7}$$

Unavailable selections give zero, and E issues no counterfactual query. Its term is additive rather than a relocation that removes the original terminal term. In particular, if all terminal group advantages are zero, E contributes no new signal, whereas M/B can still receive an available nonzero D_i.

4 Experimental Design

Comparisons. The primary planned contrast is M versus B at step 80 on the designated 100-task holdout. It tests the allocation rule for artifact-derived feedback, including the concentration that localization creates. M versus O tests its value beyond the common terminal-only recipe; M42 versus E42 provides a single-training-seed comparison with terminal edit focus. Table 1 specifies the four arms. B and M use identical eligibility, query-rate, utility, coefficient, and cap rules. Their online policies generate different trajectories after training diverges; pointwise coefficient equivalence is checked on a common frozen trajectory set, rather than assumed between unpaired online rollouts.

Tasks and splits. The reinforcement-learning pool contains 149 tasks and 2,384 aliases, retaining 16 aliases per task from the original 150-task candidate pool. A task whose expected scoring list

Table 1: Planned terminal-only (O), broadcast (B), localized (M), and terminal edit-focus (E) controls. Entries specify the per-token auxiliary coefficient, using the symbols in Equations (3) and (7); all arms share terminal reward and policy optimizer. The counterfactual (CF) trial bound is per eligible rollout. Unavailable feedback contributes zero.

Arm	Coefficient	Allocation	Extra CF trials
O	None	None	0
B	$\lambda D_i s_i$	All generated tokens	≤ 1
M	$\lambda D_i s_i L_i / m_i$	Producing-turn tokens	≤ 1
E	$\mathrm{clip}(\lambda A_i L_i / m_i, -c, c)$	Producing-turn tokens	0

omitted its issue-specific failing test was excluded prospectively from all successor training arms. The development subset therefore contains 39 tasks, all inside the training 149. Each alias is a repeated entry for the same task, not an additional unique issue. The original 40-task pilot and its fixed factual replay sample remain intact; their denominators are not reduced retrospectively. The final evaluation protocol prescribes the original 500-task list and designated 100-task holdout with their original grading labels.

For the new training runs, the evaluation complement contains 351 tasks: the designated holdout 100, the other original 250, and the excluded candidate task. This denotes exclusion from successor RL training, not a guarantee of absence from base-model pretraining. The designated holdout of 100 tasks is reported separately and is not used to choose successor checkpoints or coefficients. For comparison with the original partition, supplementary tables will also expose the original training 150, the remaining 250, and the excluded task's membership explicitly.

Model, budget, and execution. All planned arms start from the same Qwen3.5-9B base checkpoint. The fixed primary endpoint is 80 training steps. O/B/M have planned training seeds 42, 314159, and 271828; E has a separately allocated seed-42 run. Evaluation repetitions are distinct from these training replications. The policy comparisons remain incomplete at this manuscript snapshot.

The common recipe uses 16 prompt groups per update and eight rollouts per group, a learning rate of 10^{-6}, prompt-mean loss reduction, and the binary-TV DPPO threshold 0.15. Equal-reward groups are retained. The credit scale is $\lambda = 0.1$, the cap is $c = 1$, and the query rate among eligible M/B rollouts is one. The common allocation is 10 inference GPUs and four policy GPUs, with 64 concurrent primary sandboxes. Sampling uses temperature 1, top-$p = 1$, and no top-k or min-p restriction; the context budget is 98,304 tokens. Task timeouts use a multiplier of two and the budget reminder ratio is 0.35.

Shared terminal reward and environment. The resolved verdict supplies the ordinary success reward. Any format error overrides the terminal reward to -0.5; an unresolved trajectory with a detected pass-to-pass regression receives -0.25. These rules are identical for O/B/M/E and are separate from the auxiliary utility in Equation (1). Counterfactual credit is added to advantages, not to this logged terminal reward.

The agent sees a repository restricted to the task baseline Git state and runs without network access. Original and counterfactual verification use the same task preparation. Package versions normally inferred from Git history are supplied from the original task image's package metadata. Two task-specific resource corrections provide two CPUs to a test that requires them and bound numerical-library threading under an existing one-CPU quota. The same profiles apply across arms; no expected test label is rewritten.

Evaluation and accounting. The planned main evaluation samples two rollouts per task on all 500 tasks. The higher-repetition holdout evaluation samples 16 per task and includes base, O, B, M, and E. The 16-rollout holdout cohort is independent of the two-rollout cohort. Both report mean single-rollout resolution, never the best outcome among repetitions. The primary estimator averages the three paired M–B training-seed differences on the holdout; the analysis and missing-outcome rules were fixed before evaluation and are specified in Section C. Each evaluation records its model, task list, decoding settings, and attempt identities; missing attempts remain visible in the planned

denominator. Actual allocated GPU time, generated tokens, verifier trials, and unavailable feedback will be reported alongside the outcome comparisons.

5 EVALUATION

Policy effectiveness: pending. Table 2 reserves the prescribed models, training seeds, and task partitions. None of these policy comparisons has yet been measured. The primary M–B contrast tests allocation of the extra feedback on the designated holdout; M–O and the single-seed M42–E42 comparison address the corresponding terminal-feedback alternatives.

Table 2: Reserved evaluation table. Every score dash means *not measured*, not zero. Columns report mean single-rollout resolution, never the best outcome among repetitions. Here n is the number of rollouts per task and the suffix 80 denotes the prescribed training-step endpoint. The three O/B/M training seeds are separate experimental units; E has one preallocated training seed.

Model	Seed	Verified 500 $n = 2$	Train 149 $n = 2$	RL complement 351 $n = 2$	Holdout 100 $n = 16$
Base	N/A	–	–	–	–
O80	42	–	–	–	–
O80	314159	–	–	–	–
O80	271828	–	–	–	–
B80	42	–	–	–	–
B80	314159	–	–	–	–
B80	271828	–	–	–	–
M80	42	–	–	–	–
M80	314159	–	–	–	–
M80	271828	–	–	–	–
E80	42	–	–	–	–

Does artifact removal produce a usable signal? The base-model development pilot generated eight trajectories on each of the original 40 tasks. Of these 320 planned trajectories, 244 met the training query rules and all 244 intervention trials returned usable reports. Here, usability follows the training adapter's fixed-list utility contract. The exclusions were 41 format-error trajectories, 34 without an eligible block, and one without an original utility after the agent deleted installation files. This last trajectory is retained as a model failure; it is not resampled.

Table 3: Artifact-signal measurements on base-model development trajectories. The original 40-task pilot is retained in full; the 39-task column identifies the prospective training-admission stratum. Counts are trajectories or queries, not learned-policy resolution gains.

Measurement	Original 40 tasks	Admitted 39 tasks
Planned trajectories	320	312
Unavailable original utility	1	1
Issued / usable CF queries	244 / 244	238 / 238
Positive / negative contrasts	87 / 5	84 / 5
Nonzero contrasts	92	89
Queried failed original trajectories	111	111
Nonzero contrasts on failed originals	17	17
Capped nonzero contrasts	83	80

The contrast is nonzero for 92 queries (37.7%), including 17 from failed original trajectories: 12 positive and five negative, spanning 14 tasks and six repositories. These are 17 of the 111 queried failed trajectories, within 150 original trajectories with a known failed verdict. The twelve positive contrasts split evenly between a positive fail-to-pass (F2P) component and the regression indicator alone (Section D). These observations establish information availability on failures, rather than a benefit from learning with that information. Table 3 preserves the original pilot denominator and separately exposes the 39-task stratum admitted to training.

The cap activates for 83 of the 92 nonzero contrasts (90.2%), placing most observed nonzero cases in the sign-and-support regime of Equation (3). Among these 92 cases, the median selected-token fraction m/L is 2.10% and median shrinkage s is 0.181; Section E gives their distributions. Before format-error and original-utility query exclusions, 282 trajectories have a constructed intervention artifact; of these, 47 become an empty patch. The median fraction of final changed paths touched by the selected block is 0.5. One of the 92 nonzero cases instead selects paths absent from the final patch: the block repairs earlier agent damage. This illustrates why last-touch eligibility is defined relative to the pre-block tree rather than only to the final net diff. These structural measurements do not certify semantic attribution accuracy.

Is verification repeatable, and what does it cost? All 120 prescribed cross-machine factual replays match their originals at the level of reported test labels. The initial 244 intervention trials take 237.3 seconds on average, with a 416.9-second empirical 95th percentile. A separate deployment repeat of the same 244 artifacts on the designated verifier host, at concurrency 24 with images retained between trials, yields identical per-test reports for all cases and takes 115.1 seconds on average, with a 189.9-second 95th percentile. Service time includes setup, verification, and cleanup, and excludes semaphore waiting. Host placement, concurrency, and cleanup differ together; the comparison does not isolate a cache effect. Production queueing and total training cost remain unmeasured.

Does the coefficient interface preserve its intended behavior? The admitted 149-task profile has 298 gold/unchanged-tree controls with the expected resolved outcomes. Twenty CPU checks exercise the auxiliary interface, including batch concatenation and padding, group advantages, prompt-mean reduction, and zero-credit DPPO loss/gradient equivalence. Separate nonzero coefficient and backward checks validate the M/B construction. Section A specifies this evidence. Distributed nonzero-credit learning and the fixed-endpoint policy comparisons remain to be evaluated.

6 LIMITATIONS

The intervention's estimand. Final-artifact removal conditions on the other completed files. Later edits on other paths may depend on the removed block, even when no later turn overwrites its paths. The resulting contrast is not an unbiased action value, a Shapley allocation, or the outcome of a naturally continued trajectory. Last-touch and test-path eligibility also restrict which edits can receive this feedback. Snapshot attribution is temporal: background writes captured after a turn can be assigned to that turn, and restoring a last-touch block can reintroduce damage from an earlier block.

Allocation and optimization. Equal signed and absolute coefficient mass does not isolate semantic placement from every optimization effect. Localization changes concentration; adding the terminal advantage can change signs and DPPO masking. O/B/M/E evaluate a specific practical allocation rule. Claims about attribution accuracy require a suitably matched relocation control, and claims that removal information is necessary require controls using simpler factual test feedback. Those conclusions are outside the present comparison.

Empirical and benchmark scope. The planned study uses one base model and a limited RL task pool. It does not establish model-family-wide effectiveness, and base-model pretraining overlap is not measured here. An earlier adapter admitted zeros without grading reports; the affected run was excluded from the comparison. Successor training requires complete grading evidence, but this can also withhold negative feedback for model-broken installation; retry and masking rates must be reported by arm (Section A). The official evaluation's expected-test lists can omit behavior relevant to an issue; retaining its labels and denominator preserves the benchmark definition rather than repairing that limitation. Matching reports establish reproducibility of the recorded score. Missing test observations can be classified as failures, and reused descriptions can overwrite earlier test statuses. Final evaluation therefore retains raw observations and separates observed failures from outcomes requiring attribution.

7 Conclusion

Counterfactual edit credit turns one tested removal of surviving file effects into an additive signal for their producing turns. The construction makes the intervention, query budget, and receiving region explicit, while its broadcast and terminal edit-focus controls separate concrete alternative training rules. The method and implementation checks provide a concrete basis for testing whether this allocation improves repository repair.

References

Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. doi: 10.1609/aaai.v32i1.11794. URL https://ojs.aaai.org/index.php/AAAI/article/view/11794.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can Language Models Resolve Real-World GitHub Issues? *arXiv preprint arXiv:2310.06770*, 2023. URL https://arxiv.org/abs/2310.06770.

Rui Jin, Jikai Chen, Yihan Chen, Hao Zhou, Demin Zhu, Kaichen Yang, Dong Wang, Linjian Mo, and Chenyi Zhuang. Rubric-to-Code Credit Assignment for Reinforcement Learning. *arXiv preprint arXiv:2608.27906*, 2026. URL https://arxiv.org/abs/2608.27906.

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. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29557–29590. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/kazemnejad25a.html.

Yuanhao Li, Hongbo Wang, Xiaotang Shang, Xunzhu Tang, Yiming Cao, and Xuhong Chen. Boost-APR: Boosting Automated Program Repair via Execution-Grounded Reinforcement Learning with Dual Reward Models. *arXiv preprint arXiv:2605.09134*, 2026. URL https://arxiv.org/abs/2605.09134.

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 arXiv:2608.07118*, 2026. URL https://arxiv.org/abs/2608.07118.

Penghui Qi, Xiangxin Zhou, Zichen Liu, Tianyu Pang, Chao Du, Min Lin, and Wee Sun Lee. Rethinking the Trust Region in LLM Reinforcement Learning. *arXiv preprint arXiv:2602.04879*, 2026. URL https://arxiv.org/abs/2602.04879.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024. URL https://arxiv.org/abs/2402.03300.

David H. Wolpert and Kagan Tumer. Collective intelligence, data routing and braess' paradox. *Journal of Artificial Intelligence Research*, 16:359–387, 2002. doi: 10.1613/jair.995. URL https://arxiv.org/abs/1106.1821.

Simon Yu, Derek Chong, Ananjan Nandi, Dilara Soylu, Jiuding Sun, Christopher D Manning, and Weiyan Shi. Shepherd: Enabling Programmable Meta-Agents via Reversible Agentic Execution Traces. *arXiv preprint arXiv:2605.10913*, 2026. URL https://arxiv.org/abs/2605.10913.

Andreas Zeller and Ralf Hildebrandt. Simplifying and isolating failure-inducing input. *IEEE Transactions on Software Engineering*, 28(2):183–200, 2002. URL https://www.st.cs.uni-saarland.de/papers/tse2002/tse2002.pdf.

Bo-Wen Zhang, Junwei He, Wen Wang, Song-Lin Lv, Wentao Ma, Rongyi Lin, Shuhan Zhong, and Lan-Zhe Guo. CoRT: Counterfactual Replay for Token-Level Rubric-Guided Policy Optimization. *arXiv preprint arXiv:2607.25659*, 2026. URL https://arxiv.org/abs/2607.25659.

Yifei Zhou, Song Jiang, Yuandong Tian, Jason Weston, Sergey Levine, Sainbayar Sukhbaatar, and Xian Li. SWEET-RL: Training Multi-Turn LLM Agents on Collaborative Reasoning Tasks. *arXiv preprint arXiv:2503.15478*, 2025. URL https://arxiv.org/abs/2503.15478.

A Implementation and Replay Evidence

Factual reconstruction. The original development pilot contains 40 tasks with eight planned rollouts per task. An index-only rule selects repetitions 0, 3, and 6 of every task for cross-machine replay, giving 120 fixed cases. Every expected test's reported success/failure label and the task identifier are compared. All 120 cases match; unavailable cases are not replaced. Factual replay and intervention validation are separate checks.

Task controls. Each of the 149 admitted tasks has a gold-patch control with resolved verdict one and an unchanged-tree control with verdict zero under a matching profile: 298 controls. These check the configured expected-test score, not every unscored suite test. One excluded candidate's official lists omit the issue-specific failing test, so its unchanged-tree fixture scores as resolved locally; the official command-line evaluator separately filters empty submissions. The original 500-task evaluation membership and labels are retained by the protocol. Two tasks have common resource corrections: two CPUs for a required two-core test, and one numerical-library thread under an existing one-CPU quota.

Evaluation-profile controls. Two additional reference pairs illustrate different measurement limits. For `psf__requests-1724`, both the gold patch and unchanged tree pass all 85 fixed expected tests and receive native score one; the sole suite failure is outside those lists. This pair does not distinguish the patches under the recorded profile. For `psf__requests-2317`, the gold patch passes all eight fail-to-pass tests, but an expected pass-to-pass test raises a connection-reset error during an external HTTP request, producing native score zero. The unchanged tree also fails the targeted encoded-method test. Each of these four reference controls has one recorded execution. These records establish neither network reliability nor a repaired evaluation environment; their original scores and the fixed benchmark membership are retained. The logs, expected-test observations, and source hashes are indexed in `requests_control_observations_20260927.json`.

Fixed lists and empty submissions. Each gold and unchanged-tree control in the historical and common-profile pairs for `django__django-10097` reports native resolution one. In both common-profile controls, all 1,870 fixed native identifiers are recorded as `PASSED`; six full `URLValidator` test cases pass for gold and fail for the unchanged tree, and none belongs to the grading lists. These are reference test-report scores: the final evaluator's separate empty-prediction rule remains zero. The historical pair lacks original observation artifacts, so its test reports remain unresolved under the strict evidence check. The task and grading lists remain fixed; the controls and source comparisons are indexed in `django10097_control_observations_20260928.json`.

Test phases and native aliases. A recorded call phase does not establish execution of the intended assertions. In a gold/empty reference pair for `scikit-learn__scikit-learn-15100`, whitespace truncation merges three full parameterized test identifiers into one fixed native identifier. The `CountVectorizer` and `TfidfVectorizer` instances pass, while the `HashingVectorizer` instance produces a call-phase `XFAIL`. The pinned test source calls `pytest.xfail` for that estimator before the subsequent assertion block. Its reason mentions PyPy, but the branch does not check the runtime platform. The native gold score remains one; its strict admission remains unresolved. Complete identifiers, marker declarations, phase reports, and source bindings are indexed in `sklearn15100_xfail_attribution_20260928.json`.

Observed exceptions and unresolved evidence. An audit of nine selected SymPy task pairs reproduces all 18 saved native grading results. Each unchanged-tree control has one expected fail-to-pass test-body exception, every expected pass-to-pass test passes, and no expected identifier is missing. The native zeros remain unresolved under the strict admission rule. Qualified `file:test` entries marked `FAILED` alias the same exceptions recorded as bare-name `ERROR`; they are not independent failed-test witnesses. The same two additional errors outside the fixed lists occur in each SymPy19637 control and trace to Python AST compatibility. These reference diagnoses do not classify failures from future model patches. Exact evidence and unchanged decisions are indexed in `batch17_18_sympy_attribution_20260928.json`.

Implementation checks and records. The 20 reviewed CPU checks comprise four integration probes, eight readiness tests, four terminal-focus algebra tests, and four generator/trainer-hook tests. They cover alignment, observation/padding masks, terminal-return preservation, and zero-credit equivalence through GRPO, prompt-mean, and DPPO. Separate coefficient-array and synthetic backward checks cover nonzero M/B tensors. None of these is a distributed GPU optimizer test. Two original controls lack grading reports; the archived control and repair configurations reuse their project names. The original records are retained, and replacement controls use unique identities.

The reproducibility records are `reconstruction120_20260927.json`, `training_profile_validation_20260927.json`, `clean_pilot_validation_20260927.json`, and `pilot_mechanism_strata_20260927.json`, with paths and hashes of their raw inputs. Service-repeat records preserve every issued artifact. The empirical 95th percentile is sorted element $\lfloor 0.95(N-1) \rfloor$ with zero-based indexing. The original service measurement exceeds the development targets of mean 180 seconds and 95th percentile 360 seconds; the deployment repeat is a separate operational measurement, not a retrospective replacement.

Grading failures and training recovery. A later audit found numeric zeros without native grading reports in an O training run. A frozen consumption snapshot links one such trial to training call five and 18 to call 40. The run was stopped after 40 completed calls and excluded from the planned comparison; its checkpoints were not used to initialize the replacement. Successor training uses the same base and task membership, with an offline copy of the pinned native grading functions. It reproduces the reports and rewards of 265 eligible archived reference records, and a new `pydata__xarray-4966` gold/empty pair receives one/zero with complete expected-test observations. These checks do not repeat all 298 admission controls above. Numerical verifier rewards require matching report, observation, completion, and source records. Missing evidence remains unjudged under bounded retry and masking rules. Missing token metadata or an exception caught while processing an established terminal reward masks the trajectory without another generation. Caught auxiliary-credit exceptions retain terminal learning. This changes feedback missingness and cost, not the native scoring formula. Records and the earlier failures are retained in `O42_recovery_validation_20260927.json` and its bound evidence.

CPU quota and skipped expected tests. For `pylint-dev__pylint-6528`, a one-CPU profile skips four expected tests through a two-core eligibility hook. Same-image probes return effective CPU counts of one and two under the corresponding quotas. A separate profile changes only the quota to two CPUs. Its single gold/empty pair scores one/zero, with all 175 fixed test identifiers observed in each execution. Two failures outside the fixed lists remain in each execution. The original pair and grading lists are preserved; the candidate has not been adopted as a common evaluation profile. This is an environment control, not a policy comparison.

B Algorithm Conventions

Snapshots and eligibility. An initial snapshot precedes generation. After each assistant turn's sequential shell actions finish, a private Git index stages tracked changes and non-ignored untracked files within the repository using `git add -A`; it does not overwrite the agent's index. A Git tree records the resulting files, and a no-rename diff against the previous tree gives the changed paths. A format-error turn executes no action and appends an unchanged snapshot, ending a contiguous editing block. Attribution is to this observation boundary, not to individual processes that wrote files.

All paths touched within a block are considered, including paths changed and restored within it. Later changes invalidate the block even if those changes are subsequently undone. Test-path exclusion is the union of the following case-sensitive regular expressions, applied to repository-relative paths:

```
(^|/)(tests?|testing)(/|_); (^|/)test_[^/]*\.py$;
_tests?\.py$; (^|/)conftest\.py$.
```

All edit blocks receive chronological zero-based indices before eligibility filtering, and eligible blocks retain those indices. The selector minimizes the integer SHA-256 digest of `key|index`; `key` is `instance_id|rep|cache_salt`. Training uses the task-alias identity and sampling weight-version salt; the development pilot uses the task identity and literal pilot salt. A hash tie

retains the first block. Selection does not inspect a test verdict. The counterfactual patch is a binary diff of the constructed intervention tree against the initial baseline.

Delivery procedure. For each generated trajectory: (1) record snapshots and generated-turn spans; (2) form and select an eligible block; (3) restore its touched paths in the final tree; (4) reject a failed snapshot, construction error, unchanged intervention, format error, or incomplete receiving span; (5) for B/M, read the original utility and issue one intervention trial; (6) compute D, s, L, m and the corresponding coefficient array, or record an unavailable query; (7) apply the common trajectory masks and add surviving coefficients after group centering. E applies the same structural and complete-span checks, omits utility parsing and querying, and uses the terminal advantage at the trainer. Original and intervention utilities use the same fixed expected lists; unavailable utilities never become synthetic zero measurements.

Common trajectory masks. A failed trajectory without usable token metadata returns a zero loss mask. A prompt group containing a final error is wholly masked; two or more unjudged agent timeouts also mask the group, while a single such timeout masks only that member. Judged timeouts with complete metadata retain their actual reward. Context-truncated generated tokens remain trainable in this configuration, but incomplete selected-turn spans cannot receive auxiliary credit. Observation and padding tokens are always excluded. Format errors retain the common terminal penalty but receive no auxiliary term.

The successor training adapter uses at most two attempts for retryable, still-unjudged errors. A missing reward file remains unjudged even when verifier stdout exists; the earlier zero fallback was removed after the recovery audit. Unavailable original utility also prevents a CF query. Missing token metadata or exceptions caught in terminal-reward processing mask the trajectory without another generation; caught auxiliary-credit exceptions retain terminal learning. These training masks remain separate from the final evaluation protocol's explicit failure attribution.

Policy surrogate and budgets. Let $p_{it} = \pi_\theta(a_{it} \mid h_{it})$ and b_{it} be the recorded behavior-policy probability. The implementation uses $\rho_{it} = \exp(\mathrm{clip}(\log p_{it} - \log b_{it}, -20, 20))$. For $z = \widetilde{A}_{it}$, the binary-TV mask in Equation (6) is

$$q_{it} = \begin{cases} 0, & z > 0 \text{ and } p_{it} - b_{it} > 0.15, \\ 0, & z < 0 \text{ and } b_{it} - p_{it} > 0.15, \\ 1, & \text{otherwise.} \end{cases}$$

The mask is detached from differentiation. Advantages are divided by $N_g \max(1, T_g)$ before the token losses are summed. No additional importance-sampling correction is enabled. Empty groups contribute zero. The base checkpoint is the frozen Qwen3.5-9B snapshot identified by revision prefix `c2022362`; exact input identities are bound in the run records. Agent and verifier task timeouts are each 3,000 seconds before the common multiplier of two. There is no separate turn-count limit; context and wall-clock budgets apply. Commands time out after 30 seconds, and observations are truncated to 3,000 characters.

C Statistical and Cost Protocol

The analysis specification was fixed before policy evaluation. Let $y_{kir} \in \{0, 1\}$ be task i's resolution on repetition r for training seed k. For arm a, N tasks, n repetitions, and K training seeds, define

$$\widehat{p}_a = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{N} \sum_{i=1}^{N} \frac{1}{n} \sum_{r=1}^{n} y^a_{kir}, \qquad \widehat{\Delta}_{M-B} = \widehat{p}_M - \widehat{p}_B.$$

The primary population is the designated holdout 100 with $n = 16$ and $K = 3$ paired training seeds. M–O is secondary; M42–E42 has one training seed. The 500-task $n = 2$ cohort is independent and supplementary; its two attempts are not pooled into the primary cohort or selected for favorable scores.

The appendix will report approximate two-sided 95% bootstrap intervals from 10,000 draws with random seed 20260927. Each draw independently resamples training seeds and tasks with replacement,

using the same indices across compared arms and retaining all repetitions within a sampled cell. E42 comparisons resample tasks only. Three training seeds provide limited information about training randomness, so intervals accompany individual seed differences and effect sizes. Supplementary partitions are not alternative primary populations, and repeated trajectories are not independent training runs.

Attempt identity and missingness. Execution binds checkpoint, tokenizer, service, task profile, grading lists, and runner hashes to immutable manifests. Each logical task/rep slot accepts the first valid report, regardless of its score. An explicit verifier-infrastructure failure can receive at most one re-verification of the identical patch; a generation-infrastructure failure before a complete trajectory can receive at most one replacement generation. All attempts and reasons are retained. Valid failures, model-caused budget exhaustion, and model-broken installation are not retried for a better outcome. If a model-caused termination still has a valid report, that report is used; without a judgeable final artifact an established model failure is zero. Ambiguous attribution remains unresolved.

With S observed successes, J observed outcomes and H planned slots, unresolved slots yield bounds $[S/H, (S + H - J)/H]$, not an imputed point score or a reduced denominator. Difference bounds subtract opposite arm endpoints. A comparison with unresolved outcomes is reported as incomplete, without a completed-effect point estimate or significance claim. Logical slot manifests have been fixed; checkpoint loading/precision calibration, full evaluation profiles, and the strict attempt runner remain execution prerequisites.

Cost. Accounting includes allocated GPU-hours, generated tokens, original and CF verifier trials, service and queue time, failed attempts, and unused prefetches. A single extra logical query bounds query count, not total resource cost. Historical runs with invalid or incomparable feedback are excluded from policy comparisons.

D Origin of the Intervention Signal

For the same 244 usable development queries, write $D = \Delta_Y + \Delta_F + \Delta_G$, where

$$\Delta_Y = Y(P) - Y(P^{-e}), \qquad \Delta_F = 0.25\,[F(P) - F(P^{-e})], \qquad \Delta_G = -0.25\,[G(P) - G(P^{-e})].$$

The following table accounts for all 17 nonzero contrasts among the 111 queried failed originals. The breakdown is unchanged in the admitted 39-task stratum.

Contrast on a failed original	Count	Δ_Y	Δ_F	Δ_G
Positive, F2P component	6	0	0.125 or 0.25	0
Positive, regression component	6	0	0	0.25
Negative, F2P component	1	0	−0.25	0
Negative, regression component	3	0	0	−0.25
Intervention resolves the task	1	−1	−0.25	0

For six positive contrasts, the issue-test component is unchanged and the regression indicator supplies the entire utility difference. In one negative case, repetition 6 of `sympy__sympy-15875`, restoring the selected `sympy/core/add.py` changes (Y, F, G) from $(0, 0, 0)$ to $(1, 1, 0)$, giving $D = -1.25$. All other final files remain fixed during that intervention. The recorded decomposition and per-query input hashes are provided in `pilot_credit_origin_20260927.json`.

E Observed Coefficient Geometry and Example

The following summaries use all 92 usable nonzero development queries and the lower empirical quantile convention above. Magnitudes refer to each arm's receiving tokens before adding the terminal advantage or applying DPPO. The cap removes magnitude information from 83 of these cases; this is a property of the measured setting, not evidence that saturation improves learning.

Quantity	5th percentile	Median	95th percentile
Selected generated-token fraction m/L	0.00504	0.02100	0.04823
Shared shrinkage s	0.04035	0.18067	1.00000
M receiving coefficient magnitude	0.77242	1.00000	1.00000
B receiving coefficient magnitude	0.00504	0.02077	0.04823

For a concrete recorded example, repetition 7 of `django__django-14007` selects zero-based turn 105 and path `django/db/models/query.py`. The final patch changes only `django/db/models/base.py`; the selected turn restored the former file after earlier agent edits. Reinstating its pre-block contents reintroduces an indentation error, so the intervention fails before the expected tests can pass. The original has $(Y, F, G) = (1, 1, 0)$; the intervention has $(0, 0, 1)$, giving $D = 1.5$. Here $L = 21{,}162$, $m = 186$, and $s = 0.05860$. M receives coefficient 1 on the selected tokens and B approximately 0.00879 on all generated tokens; each has total signed coefficient mass 186. This is a traceable self-repair example, not an estimate of attribution accuracy.

F Configuration Companion

The accompanying `configuration-v1.zip` preserves the earlier training configuration, task/split manifests, 149 admission-control task profiles, agent template, local tokenizer/chat-template metadata, and content hashes. It predates the grading recovery in Section A. Host addresses and workspace paths are replaced by explicit placeholders; weight hashes retain their original identities. It is a configuration companion, not a complete executable runtime release. The recovered source/runtime package and final evaluation artifacts must accompany a submission.

The four local weight shards match the public Qwen/Qwen3.5-9B revision `c202236235762e1c871ad0ccb60c8ee5ba337b9a`. Local tokenizer and template files have their own recorded hashes. Training uses the language-model-only parameters with LoRA rank zero, BF16 parameters, FP32 gradient reduction, tensor parallelism four, and one update epoch per batch. The configured microbatch is one sequence per GPU; actual packing/microbatch counts are recorded per update. Optimizer settings are betas $(0.9, 0.98)$, weight decay 0.01, gradient-norm cap 1, and constant learning rate 10^{-6} without warmup. The 14 allocated GPUs are A100-SXM4-80GB; each of the two shared hosts has two 32-core Xeon Platinum 8350C CPUs. These shared resources do not imply dedicated verifier CPU time.

The corrected task identities are `pylint-dev__pylint-7277` (two CPUs) and `scikit-learn__scikit-learn-25232` (one BLAS/OpenMP thread). The excluded training candidate is `django__django-7530`. Membership lists and original grading configurations in the companion make these changes explicit.

G Assisted Development

Language-model tools assisted implementation, code review, and manuscript drafting. Experimental records and numerical summaries are retained separately from the manuscript. Implementation reviews are same-family, provisional checks; empirical conclusions depend on executable measurements.