Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Learning the Causal Value of Reflection: Counterfactual Credit Assignment for Large Reasoning Models

Ziwei Wang

wang.ziwei2@northeastern.edu

Northeastern University

Yan-Cheng Hsu

University of California San Diego

Jialun Wu

Johns Hopkins University

Duo Xu

Northeastern University

Research Article

Keywords: Counterfactual credit assignment, self-reflection, reasoning models, efficient inference

Posted Date: November 23rd, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11805442/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: The authors declare no competing interests.

Learning the Causal Value of Reflection: Counterfactual Credit Assignment for Large Reasoning Models

Ziwei Wang[1],*, Yan-Cheng Hsu[2], Jialun Wu[3], Duo Xu[4]
[1] Northeastern University (Seattle Campus), Seattle, 98109, USA
[2] University of California San Diego, La Jolla, 92093, USA
[3] Johns Hopkins University, Baltimore, 21218, USA
[4] Northeastern University (Silicon Valley Campus), San Jose, 95113, USA
*wang.ziwei2@northeastern.edu, bill.yancheng.hsu@gmail.com, jwu168@alumni.jh.edu, xu.duo3@northeastern.edu

Abstract—We present Counterfactual Reflection Credit Assignment (CRCA), an operator-level protocol comparing stop, generic review, and arithmetic verification from identical drafts. Paired correctness and token-cost differences train a prefix-only gate. With frozen Qwen3-1.7B on GSM8K, 384 held-out drafts from 128 problems yield 57.55% strict accuracy at 179.7 generated tokens per answer. Relative to always reviewing, CRCA saves 25.19% of tokens at 1.30 percentage points lower accuracy. Its utility gain over stopping is 0.0014 (paired 95% interval [-0.0066, 0.0115]). An equally sized outcome predictor selects identical actions on all test states. Action values depend on the fixed cost weight, review cap, and prefix features. The evidence supports an auditable credit protocol, with no demonstrated policy advantage from explicit differencing; the accuracy gain over stopping disappears under numeric fallback scoring.

Index Terms—Counterfactual credit assignment, self-reflection, reasoning models, efficient inference.

I. Introduction and Related Work

A correct final answer does not establish that a preceding reflection helped: stopping could return the same answer, while revision could introduce an error. Chain-of-thought prompting [1] exposes intermediate reasoning but leaves its incremental value unresolved. We ask whether matched outcomes teach a gate when one explicit review justifies its token cost.

Self-Refine [2] and Reflexion [3] expose models to feedback, while intrinsic self-correction can fail [4], [5]. SCoRe [6] trains multi-turn correction through reinforcement learning. Budget forcing in s1 [7], Adaptive Reflection and Length Coordinated Penalty [8], CyclicReflex [9], and the Echo Trap study [10] address reasoning budgets or reflection credit. Our narrower experiment freezes the model and replays two review operators against an explicit stopping reference.

CRCA predicts net review advantages from the question and draft. Its contribution is a reproducible operator-level credit protocol and a controlled evaluation of action selection, pairing, and token cost. The study concerns explicit review turns in a small hybrid reasoning model; it does not evaluate spontaneous reflection within native long reasoning traces.

II. Counterfactual Reflection Credit

A. Fixed Prefix and Executable Interventions

Let h contain a problem and its sampled initial draft. Skip (S) returns the draft answer. Keep (K) applies generic self-review to inspect and correct the solution. Replace (R) instead instructs arithmetic reconstruction and checks of units and constraints. "Keep" preserves a review operator, rather than a generated reflection sentence. Both review branches receive the identical draft and equal continuation caps.

Following counterfactual difference rewards [11], [12], we compare executable review operations with stopping. Condition verification motivates replacement [13], without reproducing its masking procedure. Figure 1 shows the information flow: references enter offline rewards only, never generation prompts or gate inputs.

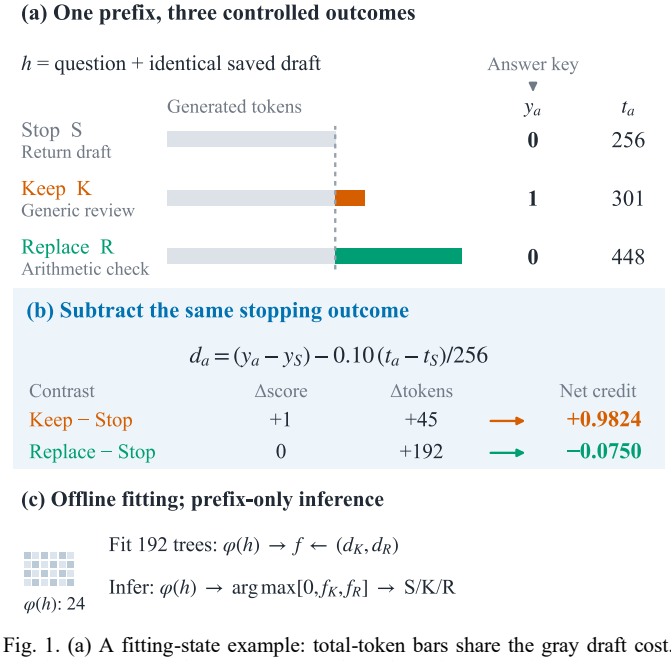


Fig. 1. (a) A fitting-state example: total-token bars share the gray draft cost. Missing answer markers score zero. (b) Subtracting stop gives incremental credit. (c) Inference uses only prefix features; the 24-feature grid is schematic and generator weights remain frozen.

B. Paired Net Credit

For action a, let y_a be one if its final numeric answer exactly matches the reference and zero otherwise. Let t_a be the number of generated tokens in the initial draft plus the chosen review, if any. The utility is

$$u_a = y_a - \lambda\, t_a / 256 \quad (1)$$

where λ is the cost weight and 256 is a fixed token-normalization constant. The initial draft cost is charged to every action. The incremental credit for each review is

$$d_a = u_a - u_S,\ a \in \{K, R\} \quad (2)$$

The common draft cost cancels in (2). The primary λ = 0.10 is a fixed design preference, specified before main collection, not a test-selected optimum. It charges 0.10 utility per 256 extra tokens. A review is preferred to stop when its predicted correctness gain exceeds λ times its expected extra tokens divided by 256. A full 192-token review therefore requires a gain above 0.075 (7.5 percentage points). Unchanged correctness earns nonpositive credit; higher λ favors stopping.

For ideal conditional predictors, $E[d_a \mid \varphi(h)] = E[u_a \mid \varphi(h)] - E[u_S \mid \varphi(h)]$, so subtracting the common stopping value preserves the utility argmax. Paired labels add no outcome information to the fully observed branches. With fitted trees, differencing can change partitions and prediction error, but offers no guaranteed policy improvement. The direct outcome comparator tests this distinction.

Every action is evaluated offline at each prefix, so no propensity correction is needed for these observed branch returns. Each contrast includes wording, computation, and revision together. It identifies the realized effect of a complete prompted operation under the fixed cap, not the effect of an individual reflection phrase or of verification independently of extra reasoning.

C. Learning a Prefix-Only Gate

A two-output regressor f fits the paired credits. Its fixed 24 hand-engineered scalars describe lengths (question words, draft words, draft tokens); eight surface counts (question/draft numbers, equations, sentence punctuation, fraction marks, percent marks, question/draft dollar signs); two numeric ratios (unique numbers and numbers per draft word); two flags (formatted answer and draft truncation); three log-magnitude summaries (question maximum/mean and draft maximum); and six lexical counts (draft uncertainty, verification, negation; question comparison, rate, time). This compact design exposes arithmetic and completion cues without learned text embeddings. Dimension 24 was not selected by feature search. References, correctness, reviews, realized review costs, identifiers, and seeds are excluded. No feature ablation establishes minimality or sufficiency; truncation-aligned decisions below are descriptive, not feature attribution.

Set $g_S(h) = 0$, $g_K(h) = f_K(\varphi(h))$, and $g_R(h) = f_R(\varphi(h))$. The selected action is

$$\pi(h) = \underset{a \in \{S,K,R\}}{\operatorname{argmax}}\ g_a(h) \quad (3)$$

Ties prefer S, then K. The gate estimates conditional incremental utility, rather than knowing a particular answer's correctness. Runtime uses one draft, one gate evaluation, and at most one review. Exhaustive branching supplies offline labels and evaluation only; language-model weights remain frozen.

III. Experimental Protocol

A. Data and Generation

We use GSM8K [14], snapshot 3101c7d50724 (7,473 training and 1,319 test problems). SHA-256 ordering of split seed 20261105 and question text selects 192 fitting and 64 validation questions from training, and independently 128 test questions. Selection ignores outcomes; exact normalized-question checks find no duplicates across selected splits or missing questions/references. No generated record is excluded.

The frozen Qwen3-1.7B Q8_0 checkpoint [15], revision 90862c4b9d27, runs in llama.cpp b12121 using its non-thinking template and explicit calculation/review prompts. Initial drafts use a 256-token cap, temperature 0.7, top-p 0.8, top-k 20, minimum-p zero, and neutral repetition penalty. Greedy reviews share a fixed 192-token cap: a bounded additional budget equal to 75% of the draft cap and identical for K/R. This pre-main engineering setting was not optimized across caps. Equality controls allowed tokens, not completion rates: longer verification may truncate more often. Changing the cap changes answers, costs, and credit labels, and requires new branches and refitting; the present replay cannot establish its optimum.

Draft seeds 17, 29, and 43 yield 576 fitting, 192 validation, and 384 test states (1,152 prefixes). Request seeds add 1009 times the selected-question index. Each prefix has S/K/R outcomes. Prompts, hashes, outputs, counters, and ordering are retained. A four-question fitting-only engineering pilot is excluded from reported metrics.

B. Controllers and Fair Comparisons

The gate uses 192 extremely randomized trees [16], scikit-learn 1.7.2, all 24 features, and estimator seed 101. Validation utility selects minimum leaf size from {4, 12, 24}; ties favor lower cost, then larger leaves. Outcome-only matches features, trees, seed, leaf grid, and validation rule, but predicts three absolute utilities instead of two paired credits. Both primary models select leaf size 24.

Controls always stop/keep/replace, select the validation-best fixed action, or apply a length gate. The latter tunes operator, threshold direction, and 21 fitting-quantile thresholds plus endpoints on validation. Ablations remove replacement, shuffle stopping labels within each fitting seed, or remove the fitting cost penalty (still evaluated at the primary cost). These compare outer gates, rather than full reinforcement-learning methods.

After fitting, controllers select one saved terminal outcome per held-out prefix. All actions were executed or return the saved draft, so selected-path returns have no missing outcomes. This is offline replay under fixed drafts, not fresh online generation.

C. Metrics and Uncertainty

The primary metric is mean utility (1), alongside accuracy, generated tokens, review frequency, rescue, and harm. Accuracy averages all three drafts per problem without best-of-three selection. Strict decimal scoring parses the last "#### number";

missing/wrong answers score zero. An already emitted correct marker remains scorable after truncation. Last-number fallback checks formatting sensitivity with actions fixed.

Cost counts draft plus selected-review completion tokens; prompts and gate computation are excluded, so token savings do not establish latency or energy gains. Non-improving reviews have $y_a \leq y_S$. Paired 95% percentile intervals resample 128 problems 5,000 times, retaining their three drafts and conditioning on fitted gates. Prespecified cost weights {0, 0.025, 0.05, 0.10, 0.20, 0.40} each refit targets and reselect leaves on validation.

Secondary diagnostics report credit MSE, five descriptive equal-count calibration bins, marker-related transitions, draft truncation, and tree-seed refits (202/303). They do not select test policies or establish calibrated individual effects.

IV. Results and Discussion

A. Selected-Path Utility and Matched Outcomes

Table I evaluates 384 held-out states. CRCA chooses 336 stops, 16 generic reviews, and 32 replacements. Its 57.55% accuracy exceeds stop by 1.04 points but trails keep by 1.30 points; it saves 25.19% of keep's tokens while using 14.69% more than stop. The label-using oracle is a nondeployable upper bound.

TABLE I. Held-out comparison at λ = 0.10

Policy	Acc. (%)	Tokens	Utility
Stop	56.51	**156.7**	0.5039
Keep	**58.85**	240.3	0.4947
Replace	42.97	318.0	0.3055
Length gate	57.29	180.3	0.5025
Outcome-only	57.55	179.7	**0.5053**
CRCA	57.55	179.7	**0.5053**
Oracle bound	60.16	161.9	0.5383

Bold: best point estimate among executable policies.

Validation-best fixed selects stop. CRCA's accuracy difference from stop has a paired 95% interval [0.26, 2.08] points; its utility difference is 0.0014 [-0.0066, 0.0115], so a utility improvement is not established. Crucially, outcome-only matches all 384 actions, not merely aggregate metrics: both achieve 57.55% accuracy, 179.7 tokens, and 0.5053 utility. Hence no incremental action-selection or accuracy–token benefit can be attributed to explicit counterfactual differencing in this experiment.

Figure 2 separates strict-score rescue and harm. Generic review repairs 9 failures without losing a passing answer; replacement repairs 5 but loses 57. All 57 losses hit the review cap without the required marker. These are budget-limited completion failures, rather than 57 observed semantic answer reversals.

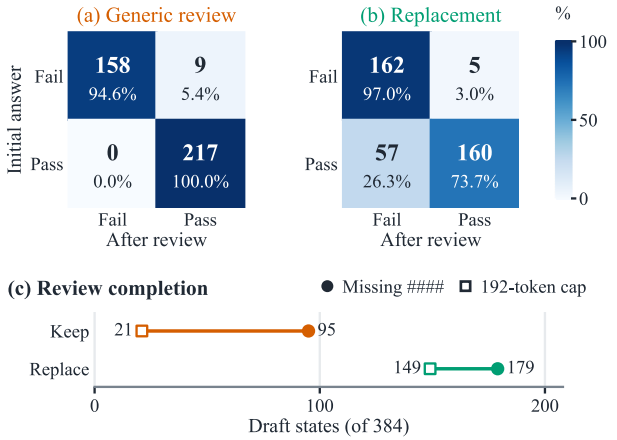


Fig. 2. (a,b) Strict-score transitions as counts and row percentages; fail includes missing markers. (c) Marker absence and cap hits can overlap. All panels cover 384 drafts from 128 problems.

B. Pairing, Replacement, and Cost Ablations

TABLE II. Ablations under the primary utility

Variant	Leaf	Acc. (%)	Tokens	Utility
CRCA	24	57.55	179.7	0.5053
No replacement	4	**57.81**	**168.7**	**0.5122**
Shuffled pairing	4	56.77	195.2	0.4914
No cost penalty	24	57.55	185.4	0.5031

Bold: best point estimate; no significance claim.

Removing replacement gives the highest fitted-gate utility point estimate (0.5122; 168.7 tokens). Its advantage over CRCA is 0.0069 [-0.0061, 0.0200], which does not establish a reliable improvement. Leaf size is reselected on validation and differs across variants (Table II).

Shuffling stopping labels increases review frequency from 12.50% to 60.94%; CRCA's utility advantage is 0.0139 [0.0039, 0.0247]. This supports preserving state alignment against deliberately corrupted labels, not superiority over direct prediction using the same aligned outcomes. Both primary predictors have access to all three branch labels.

Without the cost penalty, accuracy remains 57.55%, while tokens increase from 179.7 to 185.4 and reviews from 12.50% to 17.45%. Of CRCA's 48 reviews, 44 do not improve the strict score. Lower review frequency therefore does not establish accurate identification of useful reviews.

C. Sensitivity and Credit Prediction

Figure 3 shows the cost trade-off. At λ = 0, 0.10, 0.20, and 0.40, review counts are 67, 48, 10, and 0. From zero to 0.10, tokens fall from 185.4 to 179.7 with unchanged 57.55% accuracy. At 0.20 only generic reviews remain; at 0.40 all drafts stop (56.51%). Thus larger penalties can sacrifice accuracy to save tokens. Small-weight token use is nonmonotone because targets and leaf selection change. This sensitivity describes the chosen preference scale, without establishing 0.10 as universally optimal.

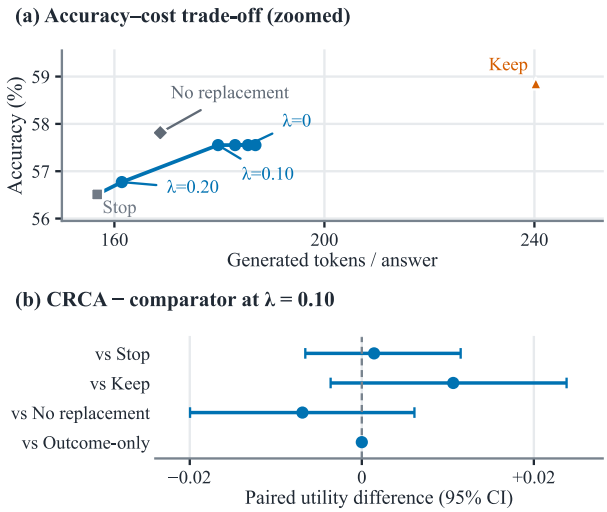


Fig. 3. (a) CRCA at six cost weights; some markers overlap and λ = 0.40 coincides with stop. Replacement remains in Table I. (b) Paired 95% utility intervals condition on fitted gates; all include zero.

Tree seeds 101/202/303 reproduce CRCA's accuracy, cost, and action counts. CRCA/outcome-only credit MSEs are 0.0667/0.0669; this descriptive difference changes no action and establishes no predictive advantage.

In CRCA's highest predicted-credit replacement bin, predicted mean credit is 0.0160 but observed credit is -0.0244; the generic-review bin observes 0.0464. These calibration diagnostics reveal remaining errors, despite identical policy decisions between predictors.

D. Formatting, Truncation, and Interpretation

All 48 initially truncated drafts receive CRCA review; all 336 untruncated drafts stop. This association suggests completion cues dominate the observed policy, but does not isolate any feature's causal contribution. Four strict-score successes are recovered. Initial drafts lack markers in 106 states, versus 95 for keep and 179 for replace. Replacement hits the 192-token review cap in 149/384 states (38.80%), versus 21/384 (5.47%) for keep. The cap therefore affects measured action value through completion as well as cost; more tokens could enable completion but also raise cost or introduce errors, so the net effect is unmeasured.

With actions fixed and last-number fallback, stop and CRCA both score 70.05%, keep 73.70%, and the no-replacement gate 72.66%. The primary gain over stop disappears under this parser, preventing interpretation as stronger arithmetic reasoning.

Exhaustive collection uses 3,456 completion calls and 450,171 generated tokens, including 295,977 for fitting/validation. These offline costs are separate from selected-path tokens in Table I; inference-path savings do not establish a net deployment benefit after collection and input processing.

Evidence is limited to one quantized model, an arithmetic subset, explicit prompts, and fixed caps/features. Pretraining exposure is unknown. Each branch has one realized continuation; greedy decoding does not guarantee identical repeated execution. Feature-group ablations, alternative review caps, repeated continuations, matched answer formats, and larger models are needed to test robustness and distinguish completion recovery from reasoning gains.

V. Conclusion

CRCA provides executable, stop-referenced credit auditing for one review turn. Its actions match direct outcome prediction exactly; explicit differencing has no demonstrated additional policy benefit. The accuracy gain over stop disappears under fallback scoring, and all 57 replacement losses involve truncation and missing markers. Cost, cap, and feature choices condition the measured trade-off; their global optimality and transfer to native long-trace reasoning remain unestablished.

References

[1] J. Wei et al., "Chain-of-thought prompting elicits reasoning in large language models," in Adv. Neural Inf. Process. Syst., vol. 35, 2022, pp. 24824–24837, doi: 10.52202/068431-1800.

[2] A. Madaan et al., "Self-Refine: Iterative refinement with self-feedback," in Adv. Neural Inf. Process. Syst., vol. 36, 2023, pp. 46534–46594, doi: 10.52202/075280-2019.

[3] N. Shinn, F. Cassano, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language agents with verbal reinforcement learning," in Adv. Neural Inf. Process. Syst., vol. 36, 2023, pp. 8634–8652, doi: 10.52202/075280-0377.

[4] J. Huang et al., "Large language models cannot self-correct reasoning yet," in Proc. ICLR, 2024, pp. 32808–32824.

[5] R. Kamoi, Y. Zhang, N. Zhang, J. Han, and R. Zhang, "When can LLMs actually correct their own mistakes? A critical survey of self-correction of LLMs," Trans. Assoc. Comput. Linguistics, vol. 12, pp. 1417–1440, 2024, doi: 10.1162/tacl_a_00713.

[6] A. Kumar et al., "Training language models to self-correct via reinforcement learning," in Proc. ICLR, 2025, pp. 54523–54549.

[7] N. Muennighoff et al., "s1: Simple test-time scaling," in Proc. EMNLP, 2025, pp. 20275–20321, doi: 10.18653/v1/2025.emnlp-main.1025.

[8] Z. Yu et al., "Stop unnecessary reflection: Training LRMs for efficient reasoning with adaptive reflection and length coordinated penalty," in Proc. ICLR, 2026, pp. 4836–4854.

[9] C. Fan, Y. Zhang, J. Jia, A. Hero, and S. Liu, "CyclicReflex: Improving reasoning models via cyclical reflection token scheduling," in Proc. ICLR, 2026, pp. 109066–109088.

[10] L. Du et al., "Escaping the Echo Trap: On credit assignment failure in multi-turn LLM self-reflection," in Proc. ACL, 2026, pp. 35393–35405, doi: 10.18653/v1/2026.acl-long.1636.

[11] J. Foerster, G. Farquhar, T. Afouras, N. Nardelli, and S. Whiteson, "Counterfactual multi-agent policy gradients," in Proc. AAAI, vol. 32, no. 1, 2018, doi: 10.1609/aaai.v32i1.11794.

[12] A. Meulemans, S. Schug, S. Kobayashi, N. Daw, and G. Wayne, "Would I have gotten that reward? Long-term credit assignment by counterfactual contribution analysis," in Adv. Neural Inf. Process. Syst., vol. 36, 2023, pp. 68685–68735, doi: 10.52202/075280-3006.

[13] Z. Wu et al., "Large language models can self-correct with key condition verification," in Proc. EMNLP, 2024, pp. 12846–12867, doi: 10.18653/v1/2024.emnlp-main.714.

[14] OpenAI, "GSM8K dataset," GitHub repository, rev. 3101c7d50724. Accessed: Nov. 5, 2026. [Online]. Available: https://github.com/openai/grade-school-math

[15] Qwen Team, "Qwen3-1.7B-GGUF," Q8_0 checkpoint and model card, rev. 90862c4b9d27. Accessed: Nov. 5, 2026. [Online]. Available: https://huggingface.co/Qwen/Qwen3-1.7B-GGUF

[16] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," Mach. Learn., vol. 63, no. 1, pp. 3–42, 2006, doi: 10.1007/s10994-006-6226-1.