BEYOND FIRST CROSSINGS: PRESERVATION IN FACTUAL REWRITING AT MATCHED MEAN OPTIMIZATION-PROMPT LOSS

Anonymous authors
Paper under double-blind review

ABSTRACT

We study which preservation differences in factual rewriting persist after matching the mean negative log-likelihood (NLL) achieved on optimization prompts. Starting from the same complete intermediate state, we apply suffix programs with different learning rates and compare discrete first crossings with directly evaluated parameter states within the final update. An initial exposure-matched load–cost reversal is absent under matched base-training steps, motivating these within-state comparisons. In five new controlled initialization–mapping pairs, subsequent mean-loss matching retains high-load old net-cost reductions of approximately 1.15 and 1.07 nats for slow relative to fast suffixes across two prefixes. Here, old net cost is the change in mean old-target NLL from the base model. On Qwen3-8B-Base/CounterFact, mean coarse-step contrast magnitudes shrink from roughly 0.03–0.06 nats to approximately 0.0063 nats or less, whereas fine-step comparisons retain much smaller differences of approximately 0.002–0.004 nats after extending eligible trajectories and narrowing the matching band. In the fine-step narrow-band comparison, slow suffixes have lower mean old net cost but higher mean held-out test NLL on complete-expression batches, despite equal test-accuracy counts in every paired batch. Thus, the large controlled differences survive mean-loss matching, whereas the coarse natural-fact contrasts largely do not. Per-edit losses and independent-expression effectiveness can still differ at a matched mean optimization loss.

1 INTRODUCTION

Factual rewriting aims to implement a requested association while preserving facts that should remain unchanged. Model-editing and continual-learning methods address this tension through localized updates, learned editors, stopping rules, replay, and protection objectives (Meng et al., 2022; Mitchell et al., 2022; Kirkpatrick et al., 2017; Gupta et al., 2024; 2025). Yet a preservation comparison also depends on which model states are evaluated. Two update programs can satisfy the same stopping criterion while attaining different editing losses. We ask which preservation differences remain when the programs begin from the same complete intermediate state and are evaluated at closely matched mean optimization-prompt NLL.

A shared stopping threshold can leave two edit runs at different levels of fit. One update may move only just below the threshold, while another goes much farther. A preservation difference between those endpoints can therefore reflect a difference in how far the requested edits have been fitted. We compare first-crossing states with directly evaluated states whose mean optimization-prompt NLL lies in a common narrow band. Both programs start from the same complete intermediate state, holding their pre-fork history fixed. The matching aligns the mean, not each edit separately. Independent expressions then test what transfers beyond the prompts used for fitting and state selection.

Our controlled study initially reveals a load–cost reversal: under matched per-fact exposure, the high-load condition has lower old net cost at a moderate editing requirement but higher cost at a stricter one. This reversal is absent when base-training steps are matched, showing that the ordering depends on the base-training comparison. We then branch fully saved intermediate states into slow and fast suffix programs to compare subsequent updates without changing their starting state.

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.

On Qwen3-8B-Base and CounterFact, mean-loss matching sharply attenuates the coarse-step contrasts. Fine-step programs retain much smaller differences, whose direction and approximate magnitude persist after extending eligible trajectories and narrowing the matching band tenfold. In the fine-step comparison under the narrower band, complete-expression batches show lower mean old net cost but higher mean test NLL for the slow suffix, while the two programs have equal test-accuracy counts in every paired batch. The residual preservation difference is accompanied by a probability-level trade-off on independent expressions.

We make three empirical contributions:

- We confirm same-state first-crossing effects on five new initialization–mapping pairs. A subsequent analysis of the same pairs shows that the large high-load preservation differences persist after matching mean optimization-prompt NLL.
- On Qwen3-8B-Base/CounterFact, mean-loss matching sharply reduces coarse-step preservation differences, while fine-step comparisons retain much smaller residuals after extending eligible trajectories and narrowing the matching band.
- In the fine-step narrow-band comparison on complete-expression batches, lower mean old net cost accompanies higher mean held-out test NLL, despite closely matched mean optimization-prompt NLL and equal test-accuracy counts in every paired batch.

2 Related Work

Optimization schedules and preservation. Repeated editing can accumulate forgetting, while stopping and norm control can limit overoptimization (Gupta et al., 2024; 2025). FINCH adapts the learning rate to minibatch loss without changing the fine-tuning objective (Prashant et al., 2026); sharpness-aware pretraining shows that base-training optimization can alter retention under later updates (Watts et al., 2026). We evaluate existing suffix schedules from a shared intermediate state at both discrete first crossings and directly evaluated mean-loss-matched terminal states.

Evaluation and independent expressions. Editor rankings depend on evaluation choices and edit-batch size (Pohl et al., 2025), and factual changes can have broader ripple effects than direct edit success captures (Cohen et al., 2024). WILD further shows gaps between editing scores and practical question answering (Yang et al., 2025b). We examine how preservation contrasts change under realized mean-loss matching, then evaluate the selected states on independent expressions using both target NLL and top-1 accuracy.

Output diagnostics and probability measures. Probability quality and discrete correctness measure different aspects of model behavior (Guo et al., 2017). Localization and effective editing sites can also diverge (Hase et al., 2023). We treat target likelihood, top-1 retention, and exploratory output diagnostics as distinct measures; Appendix E gives the decompositions and common-scale analyses at their corresponding first-crossing states.

Other editing procedures. Locate-and-edit and learned-editor methods directly construct factual updates (Meng et al., 2022; 2023; Mitchell et al., 2022). Projection and distribution-preserving approaches constrain interference through activation, gradient, decoupled-knowledge, output-distribution, or curvature structure (Fang et al., 2025; Saha et al., 2021; Xu et al., 2025; Jeong et al., 2026; Ikram et al., 2026).

3 Method

3.1 Effectiveness and preservation measures

Let $S_1 = \{(x_j^1, y_j^1)\}_{j=1}^M$ denote the requested associations and let R_0 be a fixed old-fact report set. For an evaluation set A, define

$$L_A(\theta) = -\frac{1}{|A|} \sum_{(x,y)\in A} \log p_\theta(y \mid x), \qquad D_{\text{net}}(t) = L_{R_0}(\theta_t) - L_{R_0}(\theta_0). \tag{1}$$

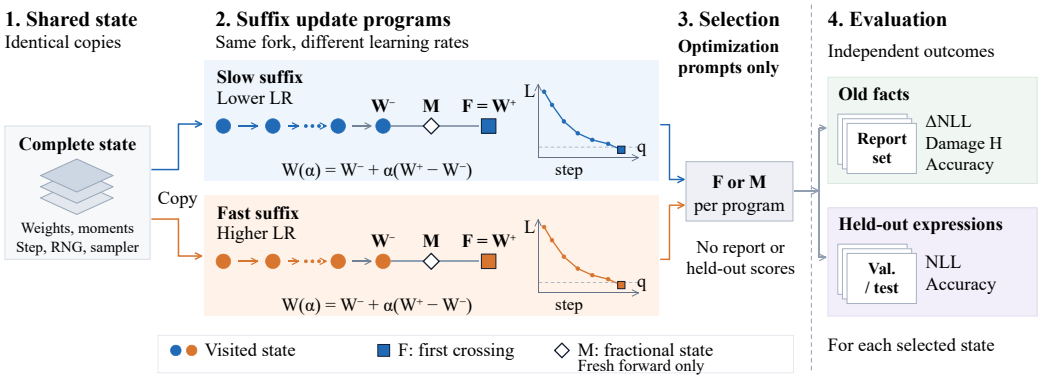

Figure 1: **Same-state suffix intervention and terminal-state selection.** Slow and fast suffix programs inherit the same complete state. F denotes the observed first-crossing state, and M denotes an evaluated fractional-step state within the final update. Optimization prompts determine each terminal state; the old report set and independent expressions are evaluated afterward.

We use *old net cost*, D_{net}, to denote the change in mean old-target NLL from the base state. Positive values indicate deterioration and negative values indicate improvement in this mean. To distinguish the signed average from damage to individual facts, we also report per-fact positive damage,

$$H_{R_0}(t) = \frac{1}{|R_0|} \sum_{i \in R_0} \max\{0, \ell_i(t) - \ell_i(0)\}, \qquad \ell_i(t) = -\log p_{\theta_t}(y_i \mid x_i). \tag{2}$$

Top-1 retention is evaluated on the fixed subset of report facts that are correct at the base state and is distinguished from accuracy over the entire report pool. Early trajectory analyses contain only aggregate old losses, whereas the same-state and matching studies record per-fact outputs.

Mean optimization-prompt NLL defines the editing criterion used for state selection. Terminal states are selected from optimization prompts alone; independent validation and test expressions then assess transfer beyond those prompts.

3.2 Same-state suffix intervention

The prefix is the portion of an editing run before the saved intermediate state; the suffix is its continuation. We run each prefix program to the first observed state with mean optimization-prompt NLL at most 0.05. We save the model parameters, optimizer moments and step counters, and random or sampling state. Two copies then continue with smaller and larger suffix learning rates, which we call the slow and fast suffixes. The labels denote the smaller and larger learning rates. For experimental unit b and prefix p, the preservation contrast is

$$d_{b,p} = D_{\text{net}}(\theta^{\text{slow}}_{b,p}) - D_{\text{net}}(\theta^{\text{fast}}_{b,p}). \tag{3}$$

Each contrast is computed within a fork: both suffixes inherit the same complete starting state, whereas different prefixes define separate forks. A negative $d_{b,p}$ indicates lower old net cost for the slow suffix. Subsequent parameter paths, optimizer states, update counts, and stopping times may differ, so the contrast measures the effect of the suffix program as a whole. Figure 1 summarizes the resulting intervention and evaluation sequence.

3.3 First crossings and matched terminal states

For a threshold q, a first crossing is the first recorded state whose mean optimization-prompt NLL is at most q. Discrete updates can overshoot this threshold by different amounts, leaving achieved mean losses unequal under a common first-crossing criterion.

For the matching analysis, let W^- and W^+ be the saved parameter values immediately before and after a bracketing final step. We evaluate

$$W(\alpha) = W^- + \alpha(W^+ - W^-), \qquad \alpha \in [0, 1]. \tag{4}$$

Table 1: Controlled base conditions for the suffix experiment.

Condition	Facts	Base steps	Role
Low-load short-training	200	200	Exposure-matched low-load reference
High-load long-training	3000	3000	High-load condition
Low-load long-training	200	3000	Training-step-matched reference

Each candidate $W(\alpha)$ is loaded into the model and evaluated with a fresh forward pass, and all reported metrics come from these evaluated states. Interpolation applies to parameter tensors within the bracket; optimizer moments and counters are not interpolated, and the accepted state is terminal. Search uses at most 20 candidate queries. A branch is recorded as a matching failure when the search yields no state inside the target band. Appendix D gives the full procedure.

The primary band, $M_0 = [0.0099, 0.0100]$, is used for the controlled matching analysis and the Qwen coarse- and fine-step comparisons. The narrower band, $M_1 = [0.009945, 0.009955]$, is a sensitivity analysis for the Qwen fine-step comparison only. Rescoring uses FP32 forward computation on the same saved parameter values, with autocast and TF32 disabled; optimization retains the training precision. Optimization prompts alone determine the selected state. Mean-loss matching aligns the batch mean while leaving the per-edit loss vector, computation, and independent-expression effectiveness unconstrained.

State	Definition
F	First-crossing state located along the realized path in FP32
M_0	Evaluated fractional state with mean NLL in $[0.0099, 0.0100]$
M_1	Qwen-only sensitivity in $[0.009945, 0.009955]$

3.4 Populations and inference

The independent unit is an initialization–mapping pair in the controlled study and a five-edit batch in the Qwen study. Discovery and confirmation cohorts are analyzed independently. Qwen prefixes are analyzed separately because they partly reuse the same batches; batches remain the independent units. Qwen uncertainty is summarized by marginal intervals from 10,000 paired batch-bootstrap resamples with seed 2026092201, conditional on the fixed old-report pool. They summarize uncertainty around the reported means; no equivalence or non-inferiority hypothesis is tested. Independent expressions are evaluated after terminal-state selection.

4 Experimental Design

4.1 Controlled rewriting and cohort structure

The controlled model is a four-layer Transformer (Vaswani et al., 2017) with hidden width 256, MLP width 1024, and four attention heads. Facts use a fixed vocabulary and prompt templates. Each edit batch jointly rewrites 20 trained subjects by permuting their learned answers, and only the final block's MLP expansion matrix is updated. The suffix experiment uses the three base conditions in Table 1. Both prefix learning rates are evaluated, 0.005 and 0.0025; the fast and slow suffix learning rates are 0.005 and 0.0025, respectively.

The discovery cohort contains five initialization–mapping pairs, and the confirmation cohort contains five new pairs. Discovery pairs use initialization seeds 301–305 and mapping seeds 401–405; confirmation pairs use initialization seeds 1001–1005 and mapping seeds 2001–2005, paired by index. Base training uses AdamW (Loshchilov & Hutter, 2019) with learning rate 0.001, weight decay 0.01, batch size 512, and gradient clipping at 1. Editing updates only the fourth block's MLP input matrix, with no rehearsal, weight decay zero, and an 800-step budget. The trajectory analysis uses a two-load by two-exposure design with 20 bases; the suffix experiment uses the three conditions in Table 1. At exposure 256, the low-load trajectories do not reach NLL 0.01 within the recorded budget.

4.2 QWEN NATURAL-FACT STUDY

We use Qwen3-8B-Base (Yang et al., 2025a) and CounterFact (Meng et al., 2022). The study has 20 batches of five edits and five independent calibration batches. Prefix learning rates are 5×10^{-4} for p_{H} and 2.5×10^{-4} for p_{L}. Coarse suffix rates are 5×10^{-4} and 2.5×10^{-4}, while fine suffix rates are 6.25×10^{-5} and 3.125×10^{-5}, respectively.

The fixed old-report pool contains 200 facts; 198 are correct under the FP32 base evaluation. Eleven batches have complete expression sets, each with 15 validation and 15 test expressions (three of each per edit). Their allocation and qualification are specified in Appendix A.6. Calibration fixed the coarse total-update budget at 80. Fine paths allow at most 64 suffix updates. Seven fine-step paths were continued under FP32 stopping within the same 64-step ceiling; Appendix D.4 describes these runs. Appendix A gives the full experimental setup. Qwen editing updates only the final-layer MLP up-projection matrix with AdamW, batch size 4, weight decay zero, and gradient clipping at 1.

4.3 COVERAGE, POPULATIONS, AND AGGREGATION

Table 2 distinguishes predefined batches, valid initial forks, forks eligible for FP32 matching, common matching success, and complete-expression comparisons. Matching fractions retain the number of valid forks within each prefix as the denominator. One initially valid p_{H} fork already satisfies the strict criterion under FP32 evaluation and is excluded from subsequent matching. Another p_{H} comparison is unavailable at M_1 because one suffix's final-step interval does not bracket the narrower band. Batch-level inclusion is reported in the supplementary data.

Aggregation follows Section 3.4: controlled cohorts are summarized over initialization–mapping pairs, while Qwen prefixes are summarized separately over edit batches. Comparisons between M_0 and M_1 use the same common batches.

Independent expressions are evaluated after the terminal states have been selected. Construction, screening, and complete-batch inclusion for the held-out expressions are specified in Appendix A.6.

Table 2: Qwen batch counts at each stage of matching and held-out evaluation. The M_1 rows apply only to the fine-step sensitivity analysis.

Stage	p_{H}	p_{L}
Predefined batches	20	20
Valid forks	14	20
Eligible for FP32 matching	13	20
Both branches reach M_0	13/14	20/20
Both branches reach M_1	12/14	20/20
M_1 batches with complete expressions	8	11

5 RESULTS

5.1 LOAD–COST ORDERING DEPENDS ON TRAINING HISTORY

The exposure-matched load–cost reversal disappears under the training-step-matched comparison. In the exposure-512 trajectories, all five high-minus-low old net-cost differences are negative at the NLL requirement of 0.05 and positive at 0.01. At exposure 256, the low-load runs remain above 0.01 throughout the recorded budget, so the stricter requirement has no common coverage.

Halving the editing learning rate preserves the exposure-matched reversal but reduces its magnitude and its accuracy consequences. At NLL 0.01, the mean high-minus-low old net-cost difference falls from 1.559 to 0.271 nats, while the corresponding old-accuracy difference changes from -0.212 to -0.004. In contrast, after training the low-load reference for the same 3000 base steps, high-minus-low old net cost is positive at both 0.05 and 0.01 under both learning rates.

The ordering varies with base-training history, motivating the same-state suffix interventions below. Appendix B.1–B.2 provides the complete threshold comparisons, interval summaries, and trajectory analyses (Figures 4–7; Tables 4–6).

5.2 CONTROLLED SUFFIX EFFECTS SURVIVE MEAN-NLL MATCHING

The high-load first-crossing contrasts, defined in Eq. (3), are reproduced in the confirmation cohort. For prefix learning rates 0.005 and 0.0025, respectively, the discovery cohort has mean slow-minus-fast differences of -1.3220 ± 0.0985 and -1.1667 ± 0.1655 nats; the new confirmation cohort has corresponding differences of -1.1529 ± 0.1819 and -1.0751 ± 0.2552 nats. Each estimate is a mean $\pm$ sample SD over five initialization–mapping pairs. The matching analysis uses the same confirmation pairs, giving five independent units.

The large differences remain after mean-loss matching. Under the same FP32 evaluator, the high-load contrast for prefix 0.005 changes from -1.1463 nats at F to -1.1457 ± 0.1781 at M_0. For prefix 0.0025, it changes from -1.0752 to -1.0721 ± 0.2513 nats. Table 3 gives the absolute matched endpoints, and Figure 2(a) shows the paired F-to-M_0 comparisons.

Table 3: High-load confirmation-cohort absolute M_0 endpoints on the 50-fact report pool, averaged over five pairs. Updates are total/suffix counts. Here every old-fact loss change is nonnegative, so mean H_{R_0} equals mean D_{net}.

Prefix	Branch	Achieved NLL	D_{net}	H_{R_0}	Old acc.	Updates	n
0.005	Slow	0.009942	0.7537	0.7537	0.928	41.0/32.8	5
0.005	Fast	0.009949	1.8993	1.8993	0.784	31.2/23.0	5
0.0025	Slow	0.009958	0.3536	0.3536	0.984	43.0/29.8	5
0.0025	Fast	0.009933	1.4257	1.4257	0.872	35.4/22.2	5

The difference is also visible in old-fact accuracy. For prefix 0.005, accuracy is 0.928 for the slow suffix and 0.784 for the fast suffix; for prefix 0.0025, the corresponding values are 0.984 and 0.872 (Table 3). The slow suffix uses more post-fork updates in both cases: 32.8 versus 23.0 and 29.8 versus 22.2, respectively. Thus, at closely matched mean editing loss, the preservation difference extends beyond target likelihood, but it is not obtained at equal update count.

The near-identical F and M_0 contrasts show that achieved mean-loss overshoot is insufficient to explain the large controlled differences. Per-edit loss allocation, update count, optimizer trajectory, and stopping time can still differ. Appendix B.4 reports all three base conditions and both report pools using the same pairs and FP32 evaluator.

5.3 NATURAL-FACT CONTRASTS ATTENUATE UNDER MEAN-NLL MATCHING

Coarse-step Qwen contrasts shrink sharply when the same batches are compared at F and M_0. For p_{H} ($n = 13$), the mean difference in old net cost changes from -0.05672 to -0.00632 nats; for p_{L} ($n = 20$), it changes from -0.03279 to approximately -3.27×10^{-5} nats. Many comparisons involve single-step crossings. The matched p_{H} mean is sensitive to one batch, and all eligible batches are included in the analysis. These estimates show substantial attenuation (Table 8).

Fine-step comparisons retain smaller residual differences. At M_0, after the additional continuation runs, the mean differences are -0.003460 nats for p_{H} ($n = 13$) and -0.002480 nats for p_{L} ($n = 20$). Table 9 reports the marginal uncertainty for the coverage-completed M_0 estimates and the M_1 common-success estimates.

We evaluate tolerance sensitivity on the batches where both suffixes also attain M_1. Within this same population, the p_{H} mean ($n = 12$) changes from -0.003410 at M_0 to -0.003502 at M_1, and the p_{L} mean ($n = 20$) changes from -0.002480 to -0.002467. The direction and approximate magnitude of the residual are stable across the two tolerances. Every leave-one-batch-out mean is also negative. Figure 2(c) and Table 10 compare M_0 and M_1 on the same batches; Appendix C.5 reports the leave-one-out ranges.

The fine-step residual should be distinguished from the overall change observed after editing. For example, at M_0 under p_{H}, the slow suffix has a negative mean old net cost (Eq. (1)) of -0.14778 nats, yet its mean positive damage (Eq. (2)) is $H_{R_0} = 0.20647$ and its old-pool accuracy is 0.8569, below the base value of 0.99 (Table 12). Improved mean old-target NLL can therefore coexist with damage to individual facts and lower accuracy. The slow–fast contrast measures a difference between two edited states; a negative contrast does not mean that either state preserves all old facts.

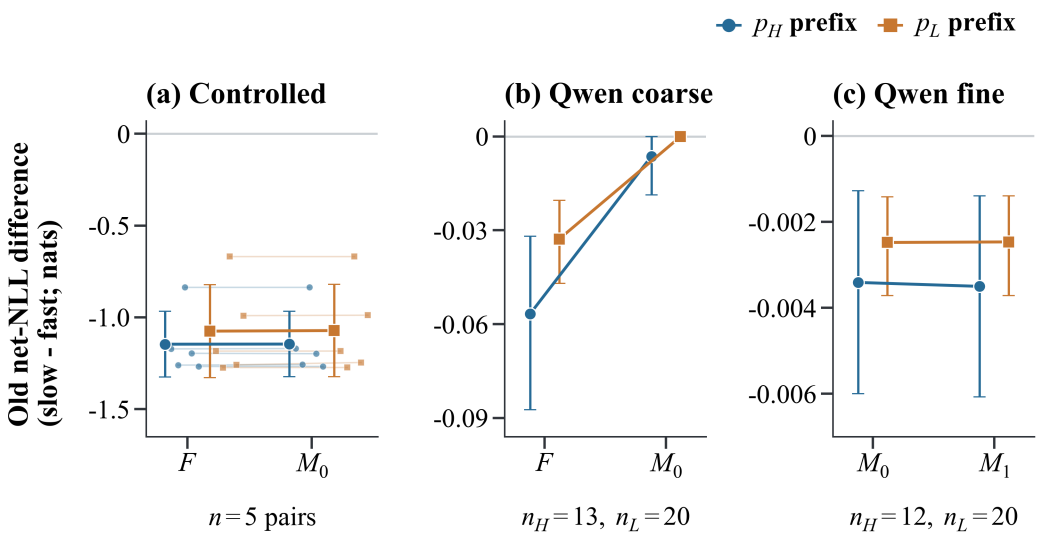

Figure 2: **Preservation contrasts under mean-loss matching and tolerance narrowing.** The vertical metric is *Difference in old net cost (slow − fast; nats)*; the horizontal gray line marks zero, and panels use separate vertical scales. (a) Controlled confirmation results from F to M_0; light lines show the five independent pairs for each prefix, and large markers with error bars show the across-pair mean and sample SD. Here p_{H} and p_{L} denote controlled prefix learning rates 0.005 and 0.0025. (b) Coarse Qwen results from F to M_0 on the same 13 p_{H} and 20 p_{L} batches. (c) Fine Qwen M_0-to-M_1 sensitivity on the common-success populations ($n_H = 12$, $n_L = 20$). Qwen error bars are 95% marginal intervals from 10,000 paired batch-bootstrap resamples on the displayed populations.

5.4 Matched mean loss coexists with different held-out likelihood

Closely matched five-edit means coexist with larger discrepancies for individual edits. Across the M_1 comparisons, the maximum absolute slow–fast difference in mean optimization-prompt NLL is approximately 6.7×10^{-6}, whereas the largest observed individual-edit difference is approximately 0.00530 nats. Matching constrains the batch mean while allowing different allocations of loss across its five edits.

Independent expressions reveal a further difference. In the fine-step M_1 comparison, the eight complete-expression p_{H} batches have a mean old net-cost difference of -0.00375 nats and a mean test-NLL difference of $+0.01123$ nats. The corresponding values for the eleven p_{L} batches are -0.00237 and $+0.02072$ nats. Each joint comparison uses the same batches for preservation and test evaluation, and the test-accuracy counts are equal for the two suffixes in every included batch. Figure 3 shows the batch-level joint differences. Table 11 and its accompanying text report the population-specific estimates and intervals, while the M_1 panel of Table 12 gives the corresponding absolute branch means.

The slow suffix has lower mean old net cost but higher mean held-out test NLL in these comparisons. This is a probability-level preservation–effectiveness trade-off: equal observed correct counts coexist with different held-out likelihoods. Appendix E reports the exploratory output decompositions and common-scale analyses at the first-crossing states.

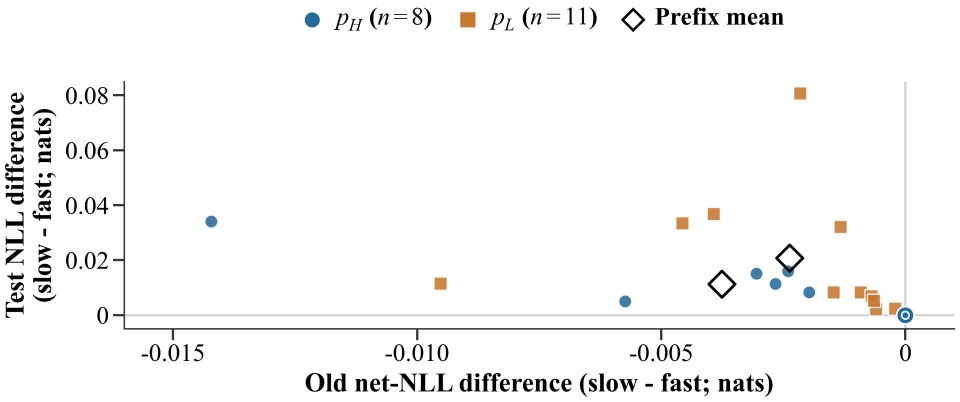

Figure 3: **Old net-cost and held-out test-NLL differences at M_1.** Each point is a batch-level slow-minus-fast comparison: the horizontal coordinate is the difference in old net cost and the vertical coordinate is the test-NLL difference. Hollow diamonds mark prefix means, and gray lines mark zero on both axes. The plot includes all complete-expression common batches (p_{H}: $n = 8$; p_{L}: $n = 11$). Test top-1 accuracy difference is zero in every included comparison. The prefixes partly reuse the same batches; some batches appear under both prefixes, and inference is prefix-specific.

6 Discussion and Limitations

First-crossing and matched-state evaluations answer different questions. The first measures preservation when a program first meets a stopping requirement; the second compares terminal states after bringing their achieved mean editing losses into close agreement. A first-crossing difference remains relevant to that stopping rule even when it shrinks after matching. In the controlled study, the large difference persists under both comparisons. In coarse-step Qwen editing, it becomes much smaller after matching. The two evaluations therefore support different interpretations of the preservation gap, rather than making one set of measurements valid and the other invalid.

The controlled and natural-fact experiments also serve different purposes. The controlled setting varies factual load and base-training history, while Qwen tests the comparison on natural-language facts in a fixed pretrained model. These experiments do not isolate which difference between the settings explains the contrast in magnitude. They show that the same evaluation procedure can retain a large preservation gap in one setting and substantially reduce it in another. The remaining Qwen fine-step difference must also be read alongside held-out performance: ranking programs only by old net cost would favor the slow suffix on average, but would omit its higher mean test NLL.

The controlled evidence comes from small discovery and confirmation cohorts, a templated task, one editable matrix, and specified optimization programs. New initialization–mapping pairs strengthen reproducibility within that setting; cross-architecture generality remains open. The natural-fact study uses one model, one dataset, single-token filtered edits, a fixed 200-fact report pool, and eleven batches with complete independent expressions. Prefixes partly reuse the same batches, and their uncertainty estimates remain prefix-specific and conditional on the report pool.

The confirmation cohort independently tests the first-crossing effect, whereas mean-loss matching and the Qwen tolerance analyses reuse the same experimental units. The output decompositions and common-scale checks are exploratory. The results support comparisons among the specified unregularized local-update programs, but do not establish a causal mechanism or generalize directly to rehearsal and subspace-constrained editors.

7 Conclusion

Preservation comparisons at a common stopping threshold can change when achieved mean optimization-prompt NLL is matched. Large controlled same-state contrasts persist under this alignment, whereas coarse-step Qwen contrasts attenuate substantially. Fine-step Qwen comparisons re-

tain much smaller differences after trajectory extension and band narrowing, but their lower mean old net cost accompanies higher held-out test NLL despite equal test-accuracy counts. The matched states can still differ in per-edit loss and held-out likelihood.

AI USE STATEMENT

Generative AI tools were used to assist with literature search and related-work exploration, research ideation and experimental design, implementation and execution of experiments, code debugging, analysis and interpretation of results, preparation of figures and tables, and drafting and revising sections of the manuscript. AI assistance also included developing and checking mathematical derivations, such as the loss-decomposition identities used in the analysis. The authors reviewed the AI-assisted material incorporated into the work and checked the reported results against the experimental records. The authors take responsibility for the final manuscript and supplementary materials, including the methodology, code, references, mathematical statements, and scientific claims.

REPRODUCIBILITY STATEMENT

Experimental details are in Appendix A; controlled results in Appendix B; natural-fact results in Appendix C; the numerical matching procedure in Appendix D; exploratory diagnostics in Appendix E; and supporting data and analysis scripts in Appendix F. Table 13 links the main-text displays to their evaluation populations and source records.

REFERENCES

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 12:283–298, 2024. doi: 10.1162/tacl_a_00644. URL https://aclanthology.org/2024.tacl-1.16/.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. AlphaEdit: Null-space constrained knowledge editing for language models. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=HvSytvg3Jh.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1321–1330, 2017. URL https://proceedings.mlr.press/v70/guo17a.html.

Akshat Gupta, Anurag Rao, and Gopala Krishna Anumanchipalli. Model editing at scale leads to gradual and catastrophic forgetting. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024. doi: 10.18653/v1/2024.findings-acl.902. URL https://aclanthology.org/2024.findings-acl.902/.

Akshat Gupta, Phudish Prateepamornkul, Maochuan Lu, Ahmed Alaa, Thomas Hartvigsen, and Gopala Anumanchipalli. Lifelong knowledge editing requires better regularization. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 22653–22675. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.findings-emnlp.1234. URL https://aclanthology.org/2025.findings-emnlp.1234/.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2301.04213.

Zarif Ikram, Arad Firouzkouhi, Stephen Tu, Mahdi Soltanolkotabi, and Paria Rashidinejad. CrispEdit: Low-curvature projections for scalable non-destructive LLM editing. *arXiv preprint arXiv:2602.15823*, 2026. URL https://arxiv.org/abs/2602.15823.

Hojun Jeong, Gyunyeop Kim, and Sangwoo Kang. KLOD: Locality-preserving knowledge editing via non-target distribution preservation. *arXiv preprint arXiv:2608.27839*, 2026. URL https://arxiv.org/abs/2608.27839.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2202.05262.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=MkbcAHIYgyS.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=0DcZxeWfOPt.

Sebastian Pohl, Max Ploner, and Alan Akbik. Towards a principled evaluation of knowledge editors. In *Proceedings of the First Workshop on Large Language Model Memorization (L2M2)*, pp. 47–60, 2025. doi: 10.18653/v1/2025.l2m2-1.4. URL https://aclanthology.org/2025.l2m2-1.4/.

Parjanya Prajakta Prashant, Jiongli Zhu, Aldan Creo, and Babak Salimi. Fine-tuning without forgetting via loss-adaptive learning rates. *arXiv preprint arXiv:2605.20005*, 2026. URL https://arxiv.org/abs/2605.20005.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations*, 2021. URL https://arxiv.org/abs/2103.09762.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. URL https://arxiv.org/abs/1706.03762.

Ishaan Watts, Catherine Li, Sachin Goyal, Jacob Mitchell Springer, and Aditi Raghunathan. Sharpness-aware pretraining mitigates catastrophic forgetting. *arXiv preprint arXiv:2605.02105*, 2026. URL https://arxiv.org/abs/2605.02105.

Haoyu Xu, Pengxiang Lan, Enneng Yang, Guibing Guo, Jianzhe Zhao, Linying Jiang, and Xingwei Wang. Knowledge decoupling via orthogonal projection for lifelong editing of large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13194–13213, 2025. doi: 10.18653/v1/2025.acl-long.646. URL https://aclanthology.org/2025.acl-long.646/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a. URL https://arxiv.org/abs/2505.09388.

Wanli Yang, Fei Sun, Jiajun Tan, Xinyu Ma, Qi Cao, Dawei Yin, Huawei Shen, and Xueqi Cheng. The mirage of model editing: Revisiting evaluation in the WILD. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15336–15354, 2025b. doi: 10.18653/v1/2025.acl-long.745. URL https://aclanthology.org/2025.acl-long.745/.

A EXPERIMENTAL DETAILS

Appendix guide. Appendix A gives experimental details and analysis populations. Appendices B and C provide additional controlled and natural-fact results, Appendix D describes numerical matching, Appendix E reports exploratory diagnostics, and Appendix F summarizes supporting data and analyses.

A.1 DISCOVERY AND CONFIRMATION EXPERIMENTS

The discovery cohort comprises the five controlled pairs used to formulate the same-state suffix comparison. The confirmation cohort comprises five new initialization–mapping pairs. Mean-loss matching reuses the confirmation cohort. The Qwen FP32 continuation, M_0-to-M_1 comparison, and leave-one-out checks reuse the same edit batches.

Discovery uses initialization seeds 301–305 paired by index with mapping seeds 401–405. Confirmation uses initialization seeds 1001–1005 paired with mapping seeds 2001–2005. The controlled bases are the low-load short-training condition (200 facts, 200 steps), high-load long-training condition (3000 facts, 3000 steps), and low-load long-training condition (200 facts, 3000 steps). Base training uses AdamW with learning rate 0.001, weight decay 0.01, batch size 512, and gradient clipping at 1. Editing updates only the fourth block's MLP input matrix and uses AdamW with $(\beta_1, \beta_2) = (0.9, 0.999)$, $\epsilon = 10^{-8}$, batch size 512, weight decay zero, gradient clipping at 1, no rehearsal, and an 800-step budget. Both prefix learning rates are evaluated, 0.005 and 0.0025; the fast and slow suffix learning rates are 0.005 and 0.0025, respectively. Edit facts are sampled uniformly with replacement. Within a fork, the two copies restore the same saved sampling state and make the same batch-construction calls, so the same suffix-step index uses the same sampled edit batch in both branches. Parameters are stored in FP32; controlled training uses BF16 autocast for the forward computation with FP32 cross-entropy and TF32 disabled. FP32 rescoring evaluates saved parameter values without retraining.

A.2 CONTROLLED TASK CONSTRUCTION

The controlled decoder-only Transformer has four layers, hidden width 256, four attention heads, MLP width 1024, and zero dropout. Prompts contain six word-level tokens of the form `<bos> The attribute of entity_00001 is`; the next-token target is a single attribute token. The vocabulary contains 3,220 entity tokens, 9,660 attribute tokens, and six template tokens. Within each initialization–mapping pair, all load conditions share this vocabulary, the entity–attribute mapping, initial parameters, and nested training sets.

Facts 0–49 form the protection pool, facts 50–99 form the 50-fact report pool, and subjects 100–119 form the 20-edit batch. Each edit batch replaces the learned answers by a single-cycle permutation of those subjects' old labels, so every requested answer differs from its subject's old answer while remaining a target observed during base training. Facts 120–199 form the external 80-fact report pool. They are trained facts in all three controlled bases, disjoint from both the 50-fact report pool and the edited subjects, and held out from state selection. Only the final block's MLP expansion matrix is updated. These task definitions are shared by the trajectory and same-state controlled analyses; the same-state study retains the factual task while changing the cohort structure and suffix intervention.

Exposure denotes the expected number of presentations per fact under uniform sampling with replacement. For factual load N, batch size 512, and requested exposure E, the base-training budget is $\lceil EN/512 \rceil$ updates. The exposure-matched conditions in Table 1 use $E = 512$, while the 200-fact, 3000-step reference has exposure 7680.

A.3 QWEN MODEL AND EDITING SETUP

The Qwen study uses `Qwen/Qwen3-8B-Base@49e3418fbbbca6ecbdf9608b4d22e5a407081db4` and updates only the final-layer MLP up-projection matrix. It uses AdamW, batch size 4, weight decay zero, and gradient clipping at 1. No separate version identifier is available for the CounterFact release used in these experiments.

CounterFact cases are screened for contextual single-token targets before batch construction. Fixed, disjoint case-ID lists define the 20 five-edit batches, five calibration batches, and 200-fact old-report pool; the supplementary data provide their fact IDs, prompts, targets, endpoints, and population summaries.

A.4 State Selection and Held-Out Evaluation

Optimization prompts alone are used to locate fork states, first crossings, brackets, and fractional-step terminal states. Old-report facts are never used to select terminal states. Validation and test expressions are also excluded from state search; they are evaluated only after terminal states are locked.

A.5 Evaluation Sets and Included Batches

Controlled summaries use initialization–mapping pairs as independent units. The Qwen population contains 20 five-edit batches, with five separate calibration batches. The fixed old-report pool contains 200 facts, of which 198 are correct under the FP32 base evaluation. Eleven batches have complete sets of 15 validation and 15 test expressions. Corrected narrow-band populations are 12/20 for $p_{\mathrm{H}}/p_{\mathrm{L}}$ in the all-common population and 8/11 in the complete-expression population. Batch membership is provided in the supplementary data.

A.6 Construction and Screening of Held-Out Expressions

For the natural-fact study, each of the five edits in a batch is assigned three validation and three test expressions, yielding 15 expressions per split when a batch is complete. Assignments are fixed by edit and split before model endpoints are scored. The supplementary data include expression IDs, text, edit association, and split, but do not reliably distinguish expression provenance. We apply the same qualification criteria to every expression rather than stratifying by source. Qualification requires each expression to preserve the intended relation, prompt the edited target as the answer, avoid duplicate assignments and literal target leakage, remain unambiguous in context, and satisfy the single-token target and contextual-tokenization checks used by the study. A batch enters the complete-expression population only when all 30 assigned expressions pass.

Eleven of the twenty batches meet this complete-batch requirement. Each of the other nine lacks at least one eligible validation or test expression. The available metadata do not support a reliable decomposition of these nine batches by failure category. Semantic qualification and split assignment precede terminal model evaluation; later checks verify technical properties such as contextual tokenization. Expression sets are evaluation-only: optimization prompts determine fork, bracket, and terminal-state selection. This is the 15+15 held-out-expression design used in the comparisons.

B Additional Controlled Results

B.1 Load reversal and training-history controls

Figure 4 and Table 4 report the six predefined NLL criteria from the trajectory experiment. First crossings are evaluated at observed states; the interpolation column provides an adjacent-state scalar sensitivity. Exposure 256 at 0.01 is not reached within the recorded budget.

The learning-rate-halving comparison preserves the exposure-matched sign reversal but reduces its strict-threshold magnitude. At NLL 0.01, lowering the edit LR from .005 to .0025 changes mean high-minus-low old cost from $1.5592 \pm .2082$ to $.2712 \pm .0808$, and old-accuracy difference from $-.212$ to $-.004$. With the step-matched low reference, high-minus-low old cost is positive at both 0.05 and 0.01: .0786 and 1.9266 at LR .005, and .0226 and .3740 at LR .0025. Pair-level values are provided in the supplementary data.

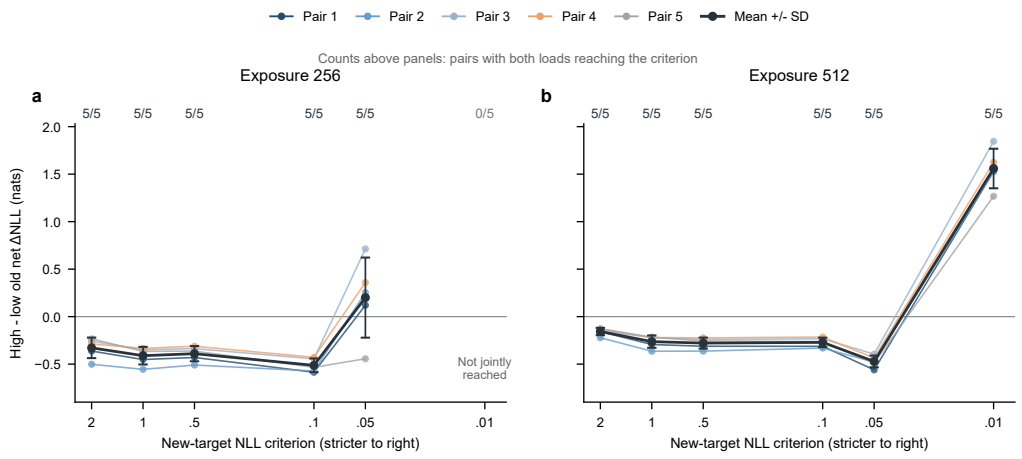

Figure 4: **Load–cost reversal across predefined NLL criteria.** Curves show five paired high-minus-low differences; dark summaries are means and sample SDs.

Table 4: Predefined NLL thresholds: high-minus-low old net cost (nats), mean ± sample SD across five pairs. All cells have 5/5 common coverage except exposure 256 at 0.01. Interpolation is an adjacent-state scalar sensitivity, not an evaluated model state.

	Exposure 256		Exposure 512	
NLL	Observed	Interpolated	Observed	Interpolated
2	-0.3287 ± 0.1080	-0.3028 ± 0.0841	-0.1567 ± 0.0385	-0.1421 ± 0.0431
1	-0.4111 ± 0.0925	-0.4081 ± 0.0944	-0.2632 ± 0.0650	-0.2512 ± 0.0639
0.5	-0.3900 ± 0.0802	-0.3974 ± 0.0795	-0.2787 ± 0.0578	-0.2787 ± 0.0593
0.1	-0.5129 ± 0.0725	-0.5253 ± 0.0807	-0.2711 ± 0.0497	-0.2728 ± 0.0490
0.05	0.2002 ± 0.4217	0.1164 ± 0.4045	-0.4717 ± 0.0620	-0.4769 ± 0.0601
0.01	NA (0/5)	NA (0/5)	1.5592 ± 0.2082	1.5538 ± 0.2160

B.2 Full trajectories

For the original controlled trajectories at exposure 512, Figure 5 and Table 6 decompose the change in old net cost between the observed NLL crossings at 0.1, 0.05, and 0.01. Table 5 reports the associated additional update counts, base-relative update norm at the strict crossing, and old-fact accuracies. Figure 6 displays all recorded states against new-target NLL, while Figure 7 shows the first-crossing lookup over each pair's common attained range. The jumps in Figure 7 mark changes in the selected recorded state. These analyses describe the observed trajectories without identifying the internal mechanism.

Table 5: Observed $0.1 \to 0.01$ interval for unconstrained controlled rewriting without rehearsal, exposure 512. Entries are mean ± sample SD across five initialization–mapping pairs. The strict-crossing norm is $\|W_{t_{0.01}} - W_0\|_F$, not the norm of displacement within the interval. Both endpoints are attained in all five pairs.

Quantity	$N = 200$	$N = 3000$
Additional updates	251.4000 ± 29.0224	26.6000 ± 1.5166
Additional old net NLL cost	0.1825 ± 0.0791	2.0128 ± 0.1705
Base-relative norm at 0.01	31.8627 ± 0.6810	21.8973 ± 0.2659
Old accuracy at 0.1	1.0000 ± 0.0000	1.0000 ± 0.0000
Old accuracy at 0.01	0.9920 ± 0.0179	0.7800 ± 0.0678

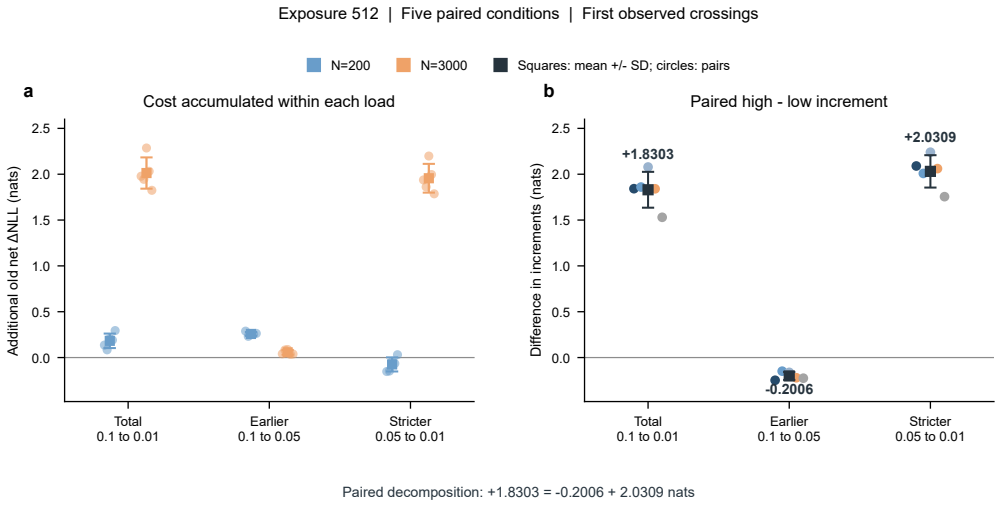

Figure 5: Old cost added within the exposure-512 strict fitting interval. SDs summarize five paired differences.

Table 6: Old net cost added between observed crossings at exposure 512 (nats), mean ± sample SD across five pairs. The last column is paired before aggregation; subinterval SDs cannot be added.

NLL interval	$N = 200$	$N = 3000$	High minus low
$0.1 \to 0.05$	0.2575 ± 0.0208	0.0569 ± 0.0260	-0.2006 ± 0.0431
$0.05 \to 0.01$	-0.0750 ± 0.0768	1.9559 ± 0.1569	2.0309 ± 0.1766
$0.1 \to 0.01$	0.1825 ± 0.0791	2.0128 ± 0.1705	1.8303 ± 0.1956

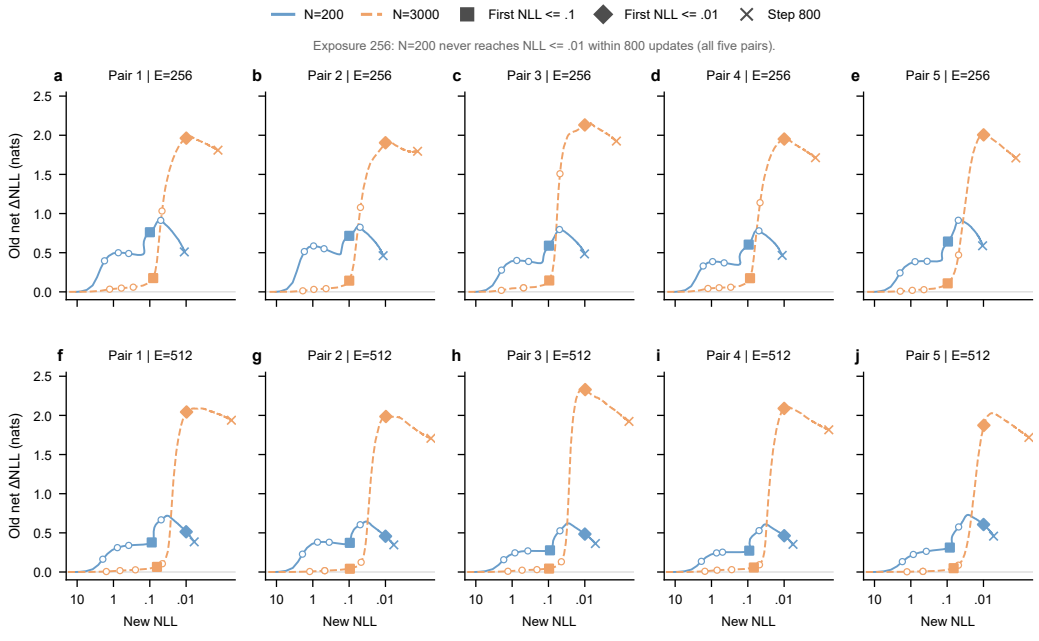

Figure 6: All 801 recorded states for each trajectory run, shown against new-target NLL.

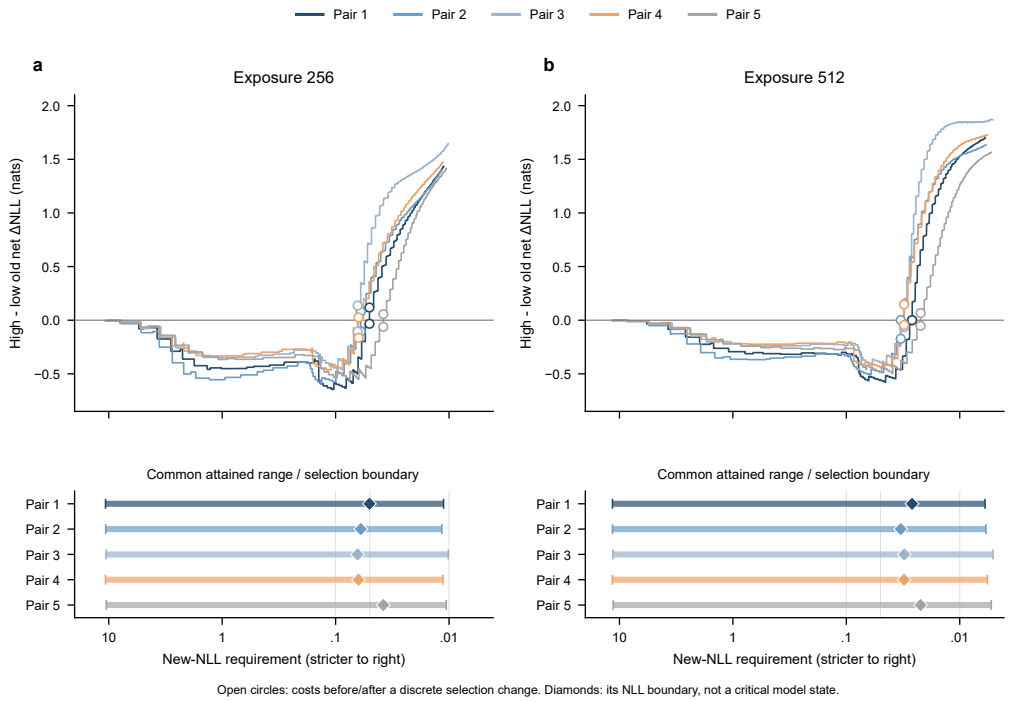

Figure 7: First-crossing lookup over each pair's common attained range. Selection jumps mark discrete lookup boundaries.

Per-run, per-pair, norm, step, and overshoot values are provided in the supplementary data. The reported norm scalars do not reconstruct the parameter vectors.

B.3 Same-state discovery and confirmation

For prefixes .005 and .0025, high-load discovery first-crossing contrasts are $-1.3220 \pm .0985$ and $-1.1667 \pm .1655$, while confirmation contrasts are $-1.1529 \pm .1819$ and $-1.0751 \pm .2552$. These are two five-pair cohorts, analyzed independently. Four-cell and pair-level values are provided in the supplementary data.

B.4 Mean-loss matching in the confirmation cohort

Table 7 reports the before/after matching comparison for all three base conditions and both report pools. The high-load 50-fact differences change only from -1.1463 to -1.1457 nats for prefix .005 and from -1.0752 to -1.0721 for prefix .0025; the external pool shows the same pattern. F-to-M_0 uses the same five pairs and the same FP32 evaluator, isolating the effect of mean-loss matching.

B.5 Computation and terminal-state scope

Per-state update counts, interpolation coefficients, brackets, and update norms are provided in the supplementary data.

C Additional Natural-Fact Results

C.1 Coarse-step raw results and M0

Table 8 reports the matched-common coarse-step comparison on the same populations at both evaluation states.

Table 7: Controlled confirmation-cohort F-to-M_0 comparison. Every row uses the same five initialization–mapping pairs at both evaluation states. Differences are slow minus fast. Uncertainty is sample SD across pairs; ΔNLL is the slow–fast difference in achieved mean optimization-prompt NLL at M_0. Base conditions are abbreviated as LL-S (low-load short-training), HL-L (high-load long-training), and LL-L (low-load long-training).

Base	Prefix	n	F ΔD	M_0 ΔD	M_0 Δacc.	mean ΔL	max $\lvert\Delta L\rvert$
50-fact report pool							
LL-S	.0050	5	$-.0674 \pm .0117$	$-.0674 \pm .0117$	$+.0040$	$-.000000$	.000009
LL-S	.0025	5	$-.0186 \pm .0034$	$-.0186 \pm .0034$	$+.0000$	$+.000010$	.000032
HL-L	.0050	5	$-1.1463 \pm .1785$	$-1.1457 \pm .1781$	$+.1440$	$-.000006$	.000038
HL-L	.0025	5	$-1.0752 \pm .2535$	$-1.0721 \pm .2513$	$+.1120$	$+.000025$	.000055
LL-L	.0050	5	$-.0661 \pm .0198$	$-.0631 \pm .0207$	$+.0000$	$-.000034$	.000077
LL-L	.0025	5	$-.0081 \pm .0037$	$-.0074 \pm .0038$	$+.0000$	$+.000008$	.000032
External 80-fact report pool							
LL-S	.0050	5	$-.0694 \pm .0167$	$-.0694 \pm .0167$	$+.0075$	$-.000000$	.000009
LL-S	.0025	5	$-.0197 \pm .0047$	$-.0197 \pm .0047$	$+.0000$	$+.000010$	.000032
HL-L	.0050	5	$-1.1753 \pm .1251$	$-1.1744 \pm .1245$	$+.1725$	$-.000006$	.000038
HL-L	.0025	5	$-1.1055 \pm .1842$	$-1.1024 \pm .1838$	$+.1125$	$+.000025$	.000055
LL-L	.0050	5	$-.0600 \pm .0228$	$-.0573 \pm .0229$	$+.0025$	$-.000034$	.000077
LL-L	.0025	5	$-.0083 \pm .0046$	$-.0077 \pm .0048$	$+.0000$	$+.000008$	.000032

Table 8: Same-population coarse F-to-M_0 attenuation in slow-minus-fast old net cost (nats).

Prefix	n	F difference	M_0 difference
p_{H}	13	-0.05672	-0.00632
p_{L}	20	-0.03279	-0.00003268

The matched-common estimates include every eligible batch, including the influential high-prefix observation.

C.2 Fine-step F and M0

After the additional continuation runs, M_0 common success is 13/14 valid forks for p_{H} and 20/20 for p_{L}. Mean old net-cost differences are -0.003460162 and -0.002480331, respectively.

Table 9 summarizes the slow-minus-fast old net-cost contrasts and their marginal uncertainty at each matching endpoint. Its p_{H} rows use different batch populations, so their difference is not a paired estimate of tolerance sensitivity.

Table 9: Marginal uncertainty for the primary fine-step endpoints. Values are slow-minus-fast differences in old net cost. The M_0 rows use the populations after the additional continuation runs; the M_1 rows use batches with common narrow-band success. In particular, the two p_{H} rows have different populations and do not define a paired band-change estimate.

Evaluation state	Prefix	n	Mean $\pm$ SD (nats)	Marginal 95% CI
M_0, after continuation	p_{H}	13	-0.003460 ± 0.004225	$[-0.005879, -0.001487]$
M_0, after continuation	p_{L}	20	-0.002480 ± 0.002729	$[-0.003713, -0.001421]$
M_1, common success	p_{H}	12	-0.003502 ± 0.004368	$[-0.006072, -0.001398]$
M_1, common success	p_{L}	20	-0.002467 ± 0.002768	$[-0.003717, -0.001393]$

C.3 M0 to M1

To separate tolerance sensitivity from changes in batch inclusion, Table 10 evaluates both matching bands on the same common-success batches within each prefix.

Table 10: **Same-batch sensitivity of Qwen fine-step old net-cost contrasts to the matching band.** Entries are batch means of $d = D_{\text{net}}^{\text{slow}} - D_{\text{net}}^{\text{fast}}$, evaluated on the fixed 200-fact old-report pool and measured in nats. Old net cost is the change in mean old-target NLL from the base model. Each row uses the same n edit batches at both matching endpoints ($n_H = 12$, $n_L = 20$). Under FP32 evaluation, M_0 and M_1 constrain mean optimization-prompt NLL to $[0.0099, 0.0100]$ and $[0.009945, 0.009955]$, respectively. Negative entries indicate lower old net cost for the slow suffix. Paired changes between columns and their uncertainty are reported below.

Prefix	Batches n	M_0: mean d (nats)	M_1: mean d (nats)
p_{H}	12	-0.003410373	-0.003501725
p_{L}	20	-0.002480331	-0.002466976

The all-common populations contain 12/20 batches, whereas the complete-expression populations contain 8/11. On the all-common population, the paired change in the suffix-program contrast, $d_{M_1} - d_{M_0}$, is -9.14×10^{-5}, 95% interval $[-2.29 \times 10^{-4}, -8.37 \times 10^{-6}]$, for p_{H}, and $+1.34 \times 10^{-5}$, interval $[-3.43 \times 10^{-5}, 7.28 \times 10^{-5}]$, for p_{L}.

C.4 Joint independent-expression evaluation

Table 11 reports paired preservation and held-out test differences at M_1, using the same complete-expression batches for both outcomes within each prefix.

Table 11: M_1 complete-expression common population. Confidence intervals use the prefix-specific sets of eight and eleven batches.

Prefix	Batches	Old net-NLL diff.	Test NLL diff.	Test acc. diff.
p_{H}	8	-0.003754953	$+0.011229775$	0
p_{L}	11	-0.002367826	$+0.020715535$	0

At M_1, the 95% old-cost intervals are $[-0.007066, -0.001332]$ for p_{H} and $[-0.004110, -0.001056]$ for p_{L}; test-NLL intervals are $[0.004761, 0.018965]$ and $[0.009094, 0.035296]$. Table 12 reports absolute branch means: its upper panel uses the full coverage-completed M_0 populations, whereas its lower panel uses the M_1 complete-expression populations and includes validation and test performance. Expression-level results and correct counts are provided in the supplementary material. Zero accuracy difference denotes equal correct counts; the NLL columns retain probability-level differences.

Table 12: Absolute fine-step branch means on the populations used for the corresponding comparisons. The M_0 panel uses the 13/20 batches after the additional continuation runs. The M_1 panel uses the complete-expression populations (8/11) and includes absolute validation and test results. Slash-separated columns report the two named quantities.

M_0, population after continuation

Prefix	Branch	n	Opt. NLL	D_{net}/H_{R_0}	Old acc./ret.	Suffix updates
p_{H}	Slow	13	.009943	$-.14778/.20647$	.8569/.8613	4.85
p_{H}	Fast	13	.009953	$-.14432/.20944$	.8562/.8605	2.77
p_{L}	Slow	20	.009947	$-.24215/.14584$	.8828/.8866	6.20
p_{L}	Fast	20	.009954	$-.23967/.14777$	.8820/.8859	3.30

M_1, complete-expression population

Prefix	Branch	n	Opt. NLL	D_{net}/H_{R_0}	Old acc./ret.	Val. NLL/acc.	Test NLL/acc.	Suffix updates
p_{H}	Slow	8	.009950	$-.17996/.18106$	.8606/.8655	1.3776/.7167	1.5505/.6583	5.63
p_{H}	Fast	8	.009950	$-.17620/.18420$	.8606/.8655	1.3696/.7250	1.5393/.6583	3.13
p_{L}	Slow	11	.009951	$-.28484/.10887$	.8950/.8994	1.8542/.5697	2.0806/.4606	6.73
p_{L}	Fast	11	.009951	$-.28247/.11062$	.8945/.8990	1.8354/.5697	2.0598/.4606	3.73

C.5 Per-edit vectors and influence diagnostics

Across the M_1 comparisons, the maximum absolute slow–fast difference between the five-edit mean optimization-prompt NLLs is 6.712×10^{-6}, while the maximum observed per-edit difference is 0.005295 nats. The latter occurs in the p_{L} batch with the largest coordinate difference. Its five slow-minus-fast differences are $[.000235, .000866, .001677, .002523, -.005295]$, with mean 1.32×10^{-6}. Leave-one-batch-out M_1 old-net means range from $-.003820$ to $-.002528$ for p_{H} and from $-.002589$ to $-.002096$ for p_{L}; all remain negative.

C.6 Additional retention-population summaries

We report all-fact accuracy on the fixed 200-fact pool separately from retention on the 198 facts correct under the FP32 base evaluation. A negative old net cost can coexist with positive H_{R_0} and individual forgetting or recovery.

D Numerical Matching

This section describes FP32 rescoring, fractional-step matching, and matching failures.

D.1 Training-time scoring versus FP32 rescoring

Training-time evaluation and FP32 rescoring use different numerical modes. FP32 rescoring reloads the same saved parameter values, evaluates forward passes in FP32, and disables autocast and TF32 while leaving the parameters unchanged. The observed BF16/FP32 differences reflect numerical evaluation of the same parameters.

D.2 Evaluated fractional-step matching

The executed search is:

1. Rescore the realized path in FP32 and identify adjacent final-step states W^- and W^+.
2. Evaluate both endpoints and accept an endpoint already inside the band. Otherwise, require an above-band and a below-band endpoint in the prescribed order; record a no-bracket outcome if this condition is not met.
3. Freeze the endpoint tensors and set $\alpha_{\mathrm{lo}} = 0$, $\alpha_{\mathrm{hi}} = 1$, with loss above the upper band edge at the former and below the lower edge at the latter.
4. Query the midpoint and evaluate the parameter state in Eq. (4) on the optimization prompts.
5. If its loss is above the band, replace α_{lo}; if below, replace α_{hi}; if inside, accept it.
6. Repeat until a candidate is accepted or the 20-query limit is reached. If no candidate is accepted, record a matching failure.
7. Save, reload, and freshly re-evaluate the accepted parameter tensors.
8. Optimizer state is not interpolated, and training stops at selection. The M_1 search uses the same terminal interval; if no point is accepted, matching fails.

Reported endpoint metrics come from fresh evaluations of the candidate model states.

D.3 Narrow-band matching failures

Failures arise from finite-precision scoring, insufficient path coverage, or the absence of a final-step bracket.

D.4 Additional FP32 continuation runs

Seven fine-step paths required one additional optimizer update under the 64-suffix-step ceiling. Replaying stored trajectories required 33 optimizer steps to identify the FP32 stopping states and 252 steps to recover terminal intervals for the tolerance analysis; these replays did not extend training.

D.5 MATCHING FAILURE MODES

We use the mutually distinguishable labels *already strict*, *path does not cover band*, *no bracketing interval*, *matching failure*, and *invalid fork or missing state*. Reload and precision checks are not counted as matching outcomes.

E EXPLORATORY OUTPUT DIAGNOSTICS

These exploratory analyses characterize output behavior without assigning a causal mechanism.

E.1 DECOMPOSITION

Let E be the edit-answer vocabulary set, $a_\theta(x) = \sum_{v \in E} p_\theta(v \mid x)$, and, for an eligible old target $y \notin E$, $q_\theta(y \mid x) = p_\theta(y \mid x)/(1 - a_\theta(x))$. Then

$$-\log p_\theta(y \mid x) = \underbrace{-\log(1 - a_\theta(x))}_{\ell_E(x)} + \underbrace{-\log q_\theta(y \mid x)}_{\ell_{\bar{E}}(x,y)},$$

so $\Delta\ell = \Delta\ell_E + \Delta\ell_{\bar{E}}$. For H_{R_0}, the positive part is applied per fact after the signed components are added, so the same additive component summary is unavailable.

E.2 CONTROLLED DISCOVERY DIAGNOSTICS

At the first observed NLL ≤ 0.01 state in the five-pair high-load discovery cohort, let $\Delta D_E(b)$ and $\Delta D_{\bar{E}}(b)$ be the edit-answer and complementary components of the same-fork suffix-program cost difference for pair b. The reported shares are means of within-pair ratios,

$$\frac{1}{5}\sum_{b=1}^{5} \frac{\Delta D_E(b)}{\Delta D_E(b) + \Delta D_{\bar{E}}(b)}.$$

They are 3.45% for prefix .0025 and 3.80% for prefix .005 on the high-load 50-fact old-report pool, leaving 96.55% and 96.20% in the complementary term. These are component shares of the same-fork suffix-program contrast, whose denominator differs from either branch's absolute endpoint damage. The confirmation-cohort M_0 analysis uses matched states instead.

E.3 NATURAL-FACT DECOMPOSITION

At the coarse first observed NLL ≤ 0.01 state, the Qwen shares use a ratio of aggregated mean suffix-program differences,

$$\frac{\mathrm{mean}_b[\Delta D_E(b)]}{\mathrm{mean}_b[\Delta D_E(b) + \Delta D_{\bar{E}}(b)]}.$$

This gives 77.38% under p_{H} and 94.22% under p_{L}. These are component shares of the coarse same-fork program contrast. Absolute endpoint damage and the later matched M_0/M_1 endpoints use different quantities.

E.4 COMMON-SCALE CHECKS

A common scalar is fit on a protection pool P and evaluated on the disjoint report pool. Let c_i^f and c_i^t denote, respectively, the centered non-edit-vocabulary logits for fact i at the fork and at the strict-threshold endpoint. The through-origin least-squares coefficient is

$$a = \frac{\sum_{i \in P} \langle c_i^f, c_i^t \rangle}{\sum_{i \in P} \| c_i^f \|_2^2}.$$

The fit uses only P; prediction errors and probability summaries are then computed on the report facts. The reference is $a = 1$, and this diagnostic is distinct from calibration-error minimization. Cell-mean fitted coefficients range from .399 to 1.060. For the high-load old pool at prefix/postfix

LR .0025, the fitted scale reduces the centered-logit residual to $.5590 \pm .0281$ of the $a = 1$ residual, but the mean absolute error for the complementary-term (B) conditional-NLL increment increases to 3.1329 ± 1.3922 times its $a = 1$ value. Thus a better centered-logit fit still predicts the preservation-defining increments poorly and substantially overestimates them for this program. At LR .005, inverse scaling reduces old-pool full NLL from $2.0645 \pm .1688$ to $.8802 \pm .2361$, while top-1 accuracy remains $.7800 \pm .0678$. A shared output-scale change does not fully account for the observed damage.

E.5 Other diagnostics

Margins, one-step response, nonlinear residuals, and Adam-history term proportions are exploratory probes; momentum, curvature, and other internal mechanisms remain unresolved.

F Reproducibility

F.1 Support for main-text displays

Table 13 summarizes the evaluation populations and source records for the main-text displays.

Table 13: Data and analyses supporting the main-text figures and tables.

Display	Analysis / evaluation	Population / estimator	Supporting material
Figure 1	Experimental design	Conceptual schematic	Figure source
Figure 2	Controlled confirmation and Qwen matching	Five pairs; prefix-specific Qwen batches	Paired summary tables
Figure 3	Qwen M_1, FP32 evaluation	Complete-expression populations, 8/11	Held-out summary tables
Table 1	Controlled design	Three base conditions; five confirmation pairs	Configurations and seeds
Table 2	Qwen batch inclusion	Prefix-specific batch counts	Batch inclusion lists
Table 3	Controlled confirmation at M_0	Five pairs per prefix	Endpoint summary tables

F.2 Analysis scripts

Supplementary scripts reconstruct the reported analyses, tables, and figures from saved model outputs.