FROM RE-EXPRESSION TO RETENTION: A FRAMEWORK FOR UNDERSTANDING FORGETTING

Anonymous authors
Paper under double-blind review

ABSTRACT

Why can on-policy learning reduce forgetting while a model adapts to new tasks? Prior studies identify on-policy data as a key factor and propose mode-seeking or KL-based explanations, but the mechanism linking new-task updates to the preservation of old capabilities remains insufficiently understood. We propose *implicit rehearsal* as a mechanism for this retention benefit: the current policy carries previously learned behaviors into newly generated responses and actions, allowing them to participate in new-task training. When these behaviors help solve the new task, feedback can reinforce their use, and the updated policy can reuse them in subsequent trajectories. This recurring process provides a route to preserving skills shared by old and new tasks without replaying old-task examples. We introduce the *Selection, Function, and Persistence* (SFP) framework to formalize this mechanism. Selection analyzes which historical contributions new-task updates reinforce; Function connects their preservation to old-task capability; Persistence tracks how their effects evolve as the policy and its training data change together. We derive conditions under which reinforcing historical behavior improves retention after further training. Complementary experiments in language models and control reinforcement learning provide evidence for the framework's components.

1 INTRODUCTION

Reinforcement learning (RL) has become a standard component of large language model (LLM) post-training, adapting pretrained models to human preferences and task objectives (Ouyang et al., 2022; OpenAI, 2025b). It strengthens reasoning capabilities (Shao et al., 2024; OpenAI, 2024) and trains agents to plan, use tools, and complete multi-step tasks (OpenAI, 2025a; Xiaomi MiMo Team, 2026; Qwen Team, 2026). As models are continually adapted to new tasks, a central question is how to acquire new capabilities while retaining those learned earlier.

Sequential training can impair earlier capabilities across neural architectures, including feedforward (McCloskey & Cohen, 1989; Goodfellow et al., 2015), convolutional (Mirzadeh et al., 2022), recurrent (Ehret et al., 2021), and Transformer networks (Wang et al., 2023). Common responses include mixing old and new data, replaying stored or generated examples (Shin et al., 2017), and restricting changes to parameters important for previous tasks (Kirkpatrick et al., 2017). InstructGPT mitigated benchmark regressions by mixing pretraining gradients into RL updates (Ouyang et al., 2022). Curriculum learning offers a complementary perspective: example selection and ordering can change learning outcomes (Bengio et al., 2009).

Recent findings offer another route into this question. RL often forgets less than supervised fine-tuning (SFT) (Lai et al., 2026; Shenfeld et al., 2025). Systematic ablations identify *on-policy data*, generated by the current model, as a key factor: low forgetting largely persists without explicit KL regularization or GRPO's advantage estimator, and refreshing self-generated SFT data can also help (Chen et al., 2026a). Yet on-policy learning can still forget substantially (Wang et al., 2026; Luo et al., 2026). A mechanistic account must therefore explain both the retention benefit and its limits: how does learning from current behavior preserve earlier capabilities, and when does that connection fail?

Our central idea is that previously learned behavior can remain part of the training signal on a new task. Current-policy generation brings that behavior into responses or actions; when it serves the

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.

new objective, feedback can support it again. We call this route *implicit rehearsal*: prior learning participates in fresh training trajectories without explicitly replaying old-task examples (Figure 1).

We propose the **Selection, Function, and Persistence (SFP)** framework to analyze this mechanism. Selection describes how new-task credit weights historical contributions to an update. Function connects these contributions to independently evaluated old-task capability. Persistence describes how their effects propagate as the updated policy generates further experience. We use SFP to test implicit rehearsal through measurements of historical expression, interventions on new-task credit, and independent evaluations of old-task capability.

This perspective also connects replay and training order. Replay directly restores old-task examples; current-policy generation can elicit earlier behavior on new examples. Curricula and starting checkpoints may change which behaviors a later task elicits and rewards, consistent with checkpoint-dependent recovery during RL (Jin et al., 2026). Shared usefulness permits reinforcement; conflicting feedback can promote forgetting.

We derive conditions connecting historical reinforcement to retention, and examine the mechanism with complementary experiments. Math-to-Code trajectories and credit interventions reveal how historical expression persists and responds to new-task feedback; control-policy lesions establish shared functional dependence; paired credit operations reveal opposing effects on old-task performance. Further related work is discussed in Appendix C.

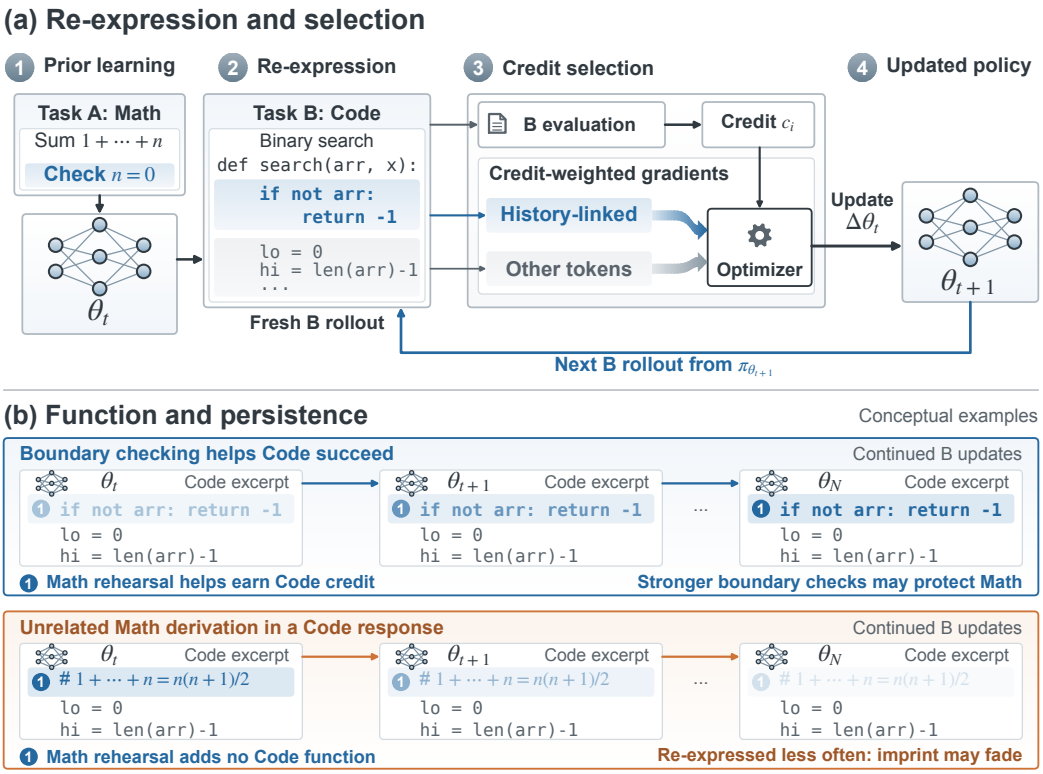

Figure 1: **SFP illustrated through on-policy learning.** (a) *Selection*: the A-initialized model $\theta_t \equiv \theta_{AB,t}$ generates fresh Task-B behavior. New-task actor credit c_i weights token gradients; the realized update $\Delta\theta_t = \theta_{t+1} - \theta_t$ changes subsequent behavior, allowing historical effects to re-enter learning. (b) *Function and persistence*: rehearsed boundary checking helps Code succeed, contributing to favorable new-task credit. Darker blue depicts increasingly reliable boundary checking, which may protect shared Math skills. The unrelated derivation adds no Code functionality; fading blue depicts declining re-expression without reinforcing feedback. N denotes a later update. Both paths are conceptual, not measured trajectories.

2 PRELIMINARIES: OBSERVABLE HISTORICAL EFFECTS

To study implicit rehearsal, we follow prior learning from current behavior into new-task updates and then into old-task performance. Output differences locate its influence, actor credit measures how the new objective weights its contributions, and independent task evaluation measures retained capability.

Policies and observations. Tasks A and B are the earlier and subsequent tasks; the model has D parameters $\theta \in \mathbb{R}^D$. Let θ_0, θ_A, $\theta_{AB,t}$, and $\theta_{B,t}$ denote the parameters of the base model, the A-trained model, and the models after t B-updates initialized from A and from the base model, respectively. For an LLM, s is a prompt plus its generated prefix, y is the next token, $z_\theta(s)$ is the vocabulary-logit vector, and $\pi_\theta(y \mid s)$ is its softmax probability. We abbreviate z_{θ_A} as z_A, and likewise for other checkpoints and policies. LLM output comparisons score the same saved prefixes. An AB–B difference measures dependence on training history, including its interaction with later training; it need not recover an unchanged A-specific component.

Locating and tracking historical expression. To locate the influence of A-training, compare the two histories on the same token and prefix. Freeze a checkpoint index t^* before intervention, and record both the sampled-token preference and the full logit difference:

$$\ell(s, y) = \log \pi_{AB,t^*}(y \mid s) - \log \pi_{B,t^*}(y \mid s), \qquad d_*(s) = z_{AB,t^*}(s) - z_{B,t^*}(s). \tag{1}$$

Positive ℓ identifies tokens favored by the A-initialized history. To measure whether an update preserves this preference, project the updated outputs onto the frozen historical direction. Let $\mathcal{D}_E$ contain saved prefixes, $p_*(s) = \pi_{AB,t^*}(\cdot \mid s)$, and $W_*(s) = \mathrm{diag}(p_*) - p_* p_*^\top$. This output-space Fisher weights logit changes by their local probability effects, so changes that leave probabilities unchanged do not count. With $\langle x, y\rangle_W = x^\top W y$ and $\|x\|_W^2 = x^\top W x$, define

$$E(\theta) = \mathrm{Avg}_{s \in \mathcal{D}_E} \frac{\langle d_*(s), z_\theta(s) - z_{B,t^*}(s)\rangle_{W_*(s)}}{\|d_*(s)\|^2_{W_*(s)}}. \tag{2}$$

Exclude degenerate denominators and average positions within prompts, then prompts equally. The anchors are $E(\theta_{AB,t^*}) = 1$ and $E(\theta_{B,t^*}) = 0$: a lower score means less expression along the measured historical direction. References, prefixes, and weights stay fixed, so only the measured model changes. Section 4.1 separately tracks the evolving training-path difference.

Credit and capability. To connect historical preference to learning, measure its contribution to the actual actor objective. On a batch of n valid tokens, let $L_i(\theta)$ be token i's contribution to the actor objective being maximized, including the implemented advantage normalization and clipping. Its credit is $c_i = \partial L_i / \partial \log \pi_\theta(y_i \mid s_i)$ and its gradient is $v_i = \nabla_\theta L_i$. Credit is positive when the objective favors raising the token's log probability, negative when it favors lowering it. To select history-favored tokens, use a separately calibrated positive threshold τ_{hist} and define $m_i = \min\{[\ell(s_i, y_i)]_+ / \tau_{\mathrm{hist}}, 1\}$, where $[x]_+ = \max(x, 0)$. Writing $\mathbf{1}\{\cdot\}$ for the indicator, attenuation selects tokens favored by both history and credit using the frozen score $q_i = m_i \mathbf{1}\{c_i > 0\}$ and loss weight $1 - \lambda q_i$, with dose $0 \le \lambda < 1$. In the local analysis, scores are computed at the shared pre-update parameters and held fixed across the compared updates.

To measure capability, let $x \sim \mathcal{D}_A$ be a prompt from a fixed old-task evaluation distribution, $Y \sim p_\theta^{\mathrm{eval}}(\cdot \mid x)$ a complete response under the declared decoder, and $R_A(x, Y)$ its verifier score (e.g., correctness). Define

$$V_A(\theta) = \mathbb{E}_{x \sim \mathcal{D}_A,\, Y \sim p_\theta^{\mathrm{eval}}(\cdot|x)} R_A(x, Y). \tag{3}$$

Here $\mathbb{E}$ denotes expectation. Sampled completions estimate V_A independently of E: one measures task performance, the other historical expression. We also write $V_A(\pi_\theta) = V_A(\theta)$. Training, historical probes, and capability evaluation can use different distributions.

3 THE SFP FRAMEWORK: SELECTION, FUNCTION, AND PERSISTENCE

A checking strategy learned in mathematics might help solve a code problem and receive further support from code rewards. This illustrates implicit rehearsal: prior learning shapes behavior, behavior earns new-task credit, and updates shape its future use. A simple example shows why task compatibility matters.

Shared usefulness creates a route to reinforcement. Consider a scalar logit ϑ and two actions: the old-useful one has probability $p = \sigma(\vartheta)$, where σ is the logistic function; the other has probability $1 - p$. Let old-task success be $V_A = p$ in this illustration. With fixed new-task rewards r_H and r_O for these actions, their expected reward J_B satisfies

$$J_B(\vartheta) = pr_H + (1-p)r_O, \qquad \frac{dJ_B}{d\vartheta} = p(1-p)(r_H - r_O). \tag{4}$$

When the old-useful action earns more new-task reward, gradient ascent increases both its probability and old-task success. Conflicting rewards reverse this effect. On-policy generation brings the modified behavior into subsequent updates. We extend this example to shared parameters, multiple functions, and evolving training states below; full statements and proofs appear in Appendix B.

3.1 Selection: which historical contributions does learning reinforce?

We first ask whether credit on history-favored tokens supports their later expression. Compare updates from the same state and batch, with these contributions retained or attenuated. Write $\langle x_i \rangle_n = n^{-1} \sum_{i=1}^n x_i$ for a batch average and reserve $\mathbb{E}$ for sampling expectations. With token gradients v_i and frozen scores q_i from Section 2, the full gradient g_0 and attenuated gradient g_q satisfy, for a fixed denominator,

$$g_0 = \langle v_i \rangle_n, \quad g_q = \langle (1 - \lambda q_i) v_i \rangle_n, \quad g_0 - g_q = \lambda \langle q_i v_i \rangle_n. \tag{5}$$

Removing credit changes both its placement and the update size. Partition the batch into strata C of comparable tokens; a uniform shuffle preserves attenuation-score mass $\sum_i q_i$ within each stratum. To predict how either update changes E, let $h = \nabla_\theta E$. Write the local parameter displacement as $U = u_0 + \eta M g$, with $\theta^+ = \theta + U$, common offset u_0, step size η, and fixed linear map $M \in \mathbb{R}^{D \times D}$ converting gradients into parameter changes. Each token's first-order influence on the readout per unit step is $\psi_i = h^\top M v_i$. For a uniform within-stratum permutation σ, let q_i^σ be the shuffled scores and U_q, U_{q^σ} the respective updates. Theorem 1 gives

$$\mathbb{E}_\sigma h^\top (U_{q^\sigma} - U_q) = \eta\lambda \sum_C \frac{n_C}{n} \operatorname{Cov}_{i \in C}(q_i, \psi_i), \tag{6}$$

where n_C is stratum size and covariance uses its empirical distribution. The covariance is positive when larger attenuations fall on contributions that most increase the historical readout. Targeting these contributions then lowers expression more than shuffling the same attenuation mass; zero covariance predicts no mean contrast in this local model. Shared parameters enter through ψ_i, whose sign can differ from the token's credit.

This local identity motivates targeted-versus-shuffled comparisons. Loss normalization and optimizer dynamics determine the realized response; Adam and multi-epoch PPO require the actual update map or independently copied sibling updates (Schulman et al., 2017b).

3.2 Function: when does preserving expression preserve capability?

To predict retention from historical expression, we need to know which functions the measured behavior supports. Expression can change differently across contexts. Applying Equation 2 separately to fixed Code and Math prefix sets, for example, gives two scalar readouts. Let K count these prespecified measurements and write $\mathbf{E} = (E_1, \ldots, E_K)^\top$; a single measurement has $K = 1$. We ask how well their joint changes predict independently evaluated old-task capability V_A.

At common parameters where these quantities are differentiable, let $H = [\nabla E_1, \ldots, \nabla E_K] \in \mathbb{R}^{D \times K}$ collect the readout gradients and $b_A = \nabla V_A$ the capability gradient. For a small displacement u, $H^\top u$ measures expression changes, whereas $b_A^\top u$ measures first-order capability change. A readout misses a functional effect if an update changes capability without changing the readout.

The parameter-space Fisher $F = \mathbb{E}_{s \sim \mathcal{D}_F, y \sim \pi_\theta}[\xi \xi^\top]$ provides a local policy-change metric, with fixed reference-prefix distribution $\mathcal{D}_F$ and score gradient $\xi = \nabla_\theta \log \pi_\theta(y \mid s)$ (Kakade, 2001). Restrict to a common subspace where F is positive definite; define $\|u\|_F^2 = u^\top F u$. To predict capability change from expression, choose weights $\beta \in \mathbb{R}^K$ so the combined readout gradient best approximates b_A, minimizing the residual $\varepsilon_A = \|b_A - H\beta\|_{F^{-1}}$. Theorem 2 implies

$$\left| b_A^\top u - \beta^\top H^\top u \right| \le \varepsilon_A \|u\|_F. \tag{7}$$

Here β converts expression changes into predicted capability changes; ε_A measures sensitivity to changes the readouts miss. If $b_A \in \mathrm{span}(H)$, every capability-relevant direction is covered and first-order prediction is exact. Otherwise, the residual bounds the error at a given update size. The coefficients use frozen-probe gradients and verifier-weighted scores of independent A-task completions (Appendix B.1).

Why direction matters beyond KL. Equal KL can accompany unequal functional damage because performance and policy-change curvatures differ (Appendix B.6). A damaging lesion establishes dependence, not that increasing the component helps: either direction from an optimum can hurt. This motivates matched lesions with independent task evaluations.

Measuring function at finite intervention sizes. Two observable comparisons connect these conditions to experiments. Matched lesions test functional direction at equal perturbation size; projection-preserving controls test whether the readout captures all functional change. Proposition 1 quantifies the latter's prediction error. Paired positive and negative perturbations further distinguish directional responses from symmetric finite effects (Appendix A.3.1), probing changes beyond a linear approximation.

3.3 Persistence: does local support change the final outcome?

On-policy learning feeds an updated policy back into the generation of training data. An update can therefore affect old capability both immediately and through the experience and updates that follow. Because later learning depends on optimizer memory as well as model weights, let S contain parameters, optimizer and critic states, and all memory needed to resume learning. Compare reference rule 1 (e.g., retain credit) with alternative rule 0 (attenuate it) from the same initial state distribution, and write $v(S) = V_A(\theta(S))$, where $\theta(S)$ extracts model parameters. At reference-path state S_t, applying either rule gives sibling next states $S'_{1,t}$ and $S'_{0,t}$.

To measure an update's lasting effect, continue both successor states under the same alternative rule. For a horizon of N updates, define $J^0_{t+1}(S)$ as expected A-task performance after continuing from S under rule 0 for the remaining $N - t - 1$ updates, with $J^0_N = v$. Let $\mathbb{E}_r$ average training paths under rule $r \in \{0, 1\}$; within $\mathbb{E}_1$, both sibling transitions start at the same reference-path state. The immediate sum and terminal relative effect are

$$L_N = \sum_{t=0}^{N-1} \mathbb{E}_1[v(S'_{1,t}) - v(S'_{0,t})], \tag{8}$$

$$\Delta_N = \sum_{t=0}^{N-1} \mathbb{E}_1[J^0_{t+1}(S'_{1,t}) - J^0_{t+1}(S'_{0,t})]. \tag{9}$$

Theorem 3 establishes $\Delta_N = \mathbb{E}_1 v(S_N) - \mathbb{E}_0 v(S_N)$ by the performance-difference identity on training states (Schulman et al., 2017a). Positive L_N means an average immediate benefit of rule 1; positive Δ_N means better old-task performance at the end than under rule 0. Later learning can change the sign, so immediate support and final retention require separate analysis. Either rule may still forget relative to initialization.

Putting the links together. We now combine selection, function, and persistence. Let $\Delta\mathbf{E}_t = \mathbf{E}(\theta(S'_{1,t})) - \mathbf{E}(\theta(S'_{0,t}))$ and calibrate β_t at S_t. Weighting each expression contrast by its functional coefficients gives the accumulated prediction Z_N. Two error budgets connect it to retention: A_N bounds the error in predicting immediate function from the readouts, using gradient coverage and Taylor curvature (Equation 38); B_N bounds the change from immediate to continuation effects (Equation 43). Specifically,

$$Z_N = \sum_{t=0}^{N-1} \mathbb{E}_1[\beta_t^\top \Delta\mathbf{E}_t], \qquad |L_N - Z_N| \le A_N, \quad |\Delta_N - L_N| \le B_N. \tag{10}$$

The triangle inequality gives $|\Delta_N - Z_N| \le A_N + B_N$. Thus $Z_N > A_N + B_N$ suffices for relative protection: predicted support exceeds both functional prediction error and the change introduced by later learning. Finite-sample use additionally requires statistical uncertainty bounds.

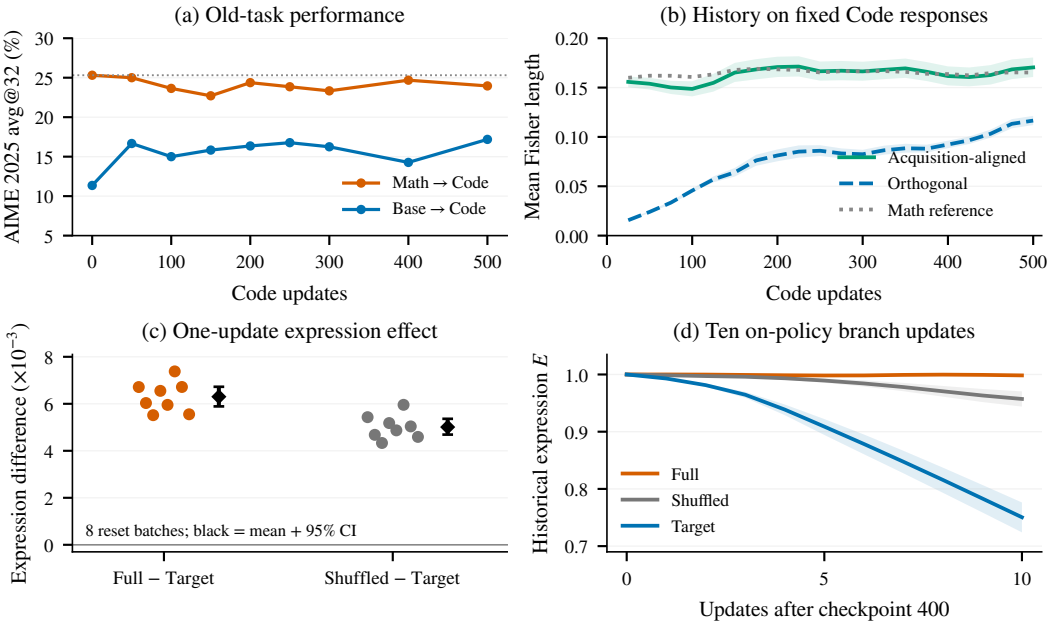

Figure 2: **Historical expression participates in Code learning.** (a) AIME accuracy. (b) Aligned and orthogonal components; dotted: Math–Base reference length. (c) Eight reset batches; black markers: means and 95% batch-bootstrap intervals. (d) Ten on-policy updates. FULL retains credit; TARGET attenuates history-weighted positive credit; SHUFFLED permutes masks. Bands in (b,d) resample 100 probe prompts, conditional on the observed runs.

4 EMPIRICAL EVIDENCE

LLM experiments locate historical expression, intervene on Code credit, and track subsequent expression. Control experiments test shared function and separate competing credit effects. Protocols appear in Appendix A; intervals below are 95% bootstrap intervals.

4.1 LANGUAGE MODELS: NEW-TASK CREDIT SUSTAINS HISTORICAL EXPRESSION

Learning Code while retaining Math. We compare one Math→Code and one Base→Code GRPO run using OLMo checkpoints (Olmo et al., 2025), each with 500 updates. On the Math-initialized path, AIME 2025 accuracy starts at 25.3% and ends at 24.0% (Figure 2a), while HumanEval+ and MBPP+ improve from 53.0% and 57.7% to 61.6% and 63.0%. Base initialization yields 60.4% and 60.6% on these Code benchmarks. The Math-initialized model thus learns Code while approximately maintaining Math accuracy. AIME scores include answer-format compliance; decoding and parsing diagnostics appear in Appendix A.7.

What remains from the earlier training? Does the initial Math preference fade, or do other differences grow around it? On fixed responses to 100 held-out Code prompts, we separate the current history difference $d_t(s) = z_{AB,t}(s) - z_{B,t}(s)$ into components along and Fisher-orthogonal to the initial Math–Base difference $a(s) = z_A(s) - z_0(s)$ (Appendix A.1). On Code-model responses, the mean aligned length remains 0.925–1.031 times the mean initial length across updates 25–500, while the mean orthogonal length grows from 0.016 to 0.117 (Figure 2b). Thus the original Math-aligned output component remains present as other differences grow. Official solutions show the same endpoint pattern (Figure 4).

Does Code credit help maintain that expression? If Code credit helps sustain historical expression, attenuating it should reduce that expression. At checkpoint 400, the mask m_i from Section 2 selects history-favored tokens for this intervention. FULL keeps unit loss weights. TARGET attenuates according to each token's mask; SHUFFLED uses masks $\widetilde{m}_i$ permuted within strata of positive-

credit tokens:

$$w_i^{\text{Target}} = 1 - \tfrac{1}{2} m_i \mathbf{1}\{c_i > 0\},$$
$$w_i^{\text{Shuffled}} = 1 - \tfrac{1}{2} \widetilde{m}_i \mathbf{1}\{c_i > 0\}. \tag{11}$$

For arm r after k additional updates, write $E_r(k) = E(\theta_{r,400+k})$, using Equation 2 with the direction, Fisher weights, B-400 baseline, and probe responses all frozen at update 400. Thus every arm starts at $E_r(0) = 1$. A separate rollout illustration locates these historical preferences within generated Code responses (Figure 6).

For each of eight rollout batches, we apply the three updates separately from the same training state. TARGET leaves less historical expression than both FULL and SHUFFLED in every batch. The mean FULL−TARGET and SHUFFLED−TARGET expression gaps are 0.00630 [0.00589, 0.00672] and 0.00501 [0.00469, 0.00536], respectively (Figure 2c). Thus Code credit helps maintain this historical expression: attenuating the selected contributions weakens it. Removed positive-credit mass differs by at most 0.21% between the attenuated arms. These contrasts include the induced policy drift; Appendix A.1 reports update sizes.

The expression effect accumulates during on-policy continuation. To test whether the expression effect persists as training data adapt to the policy, each branch generates its own subsequent responses. After ten updates, expression is 0.998 under FULL, 0.957 under SHUFFLED, and 0.750 under TARGET (Figure 2d). Paired prompt-bootstrap differences are 0.248 [0.223, 0.273] and 0.207 [0.192, 0.222]. TARGET has lower expression than both controls on all 100 probe prompts (Appendix A.1). The decrease thus extends across prompts and accumulates during continued Code training.

Does sustaining expression also preserve Math performance? This tests the functional consequence of implicit rehearsal: if the supported behavior serves Math, retaining its Code credit should help preserve Math performance as Code learning continues. Table 1 evaluates AIME 2025 after ten and thirty further Code updates. With 128 responses per problem, pass@128 measures the fraction of problems solved at least once, testing whether a successful solution remains accessible within this sampling budget.

At step 410, FULL and TARGET have equal coverage, while SHUFFLED solves more problems. Their subsequent paths diverge: FULL improves, SHUFFLED remains stable, and TARGET declines. By step 430, both controls solve more problems within 128 attempts than TARGET. In these observed paths, lower historical expression is accompanied by lower later coverage of successful Math solutions. This pattern is consistent with the SFP prediction that preserving useful historical contributions to Code updates can support old-task retention. The control experiments next test shared usefulness by directly perturbing historical behavior. Evaluation details and Math-prefix probes appear in Appendix A.7.

Table 1: AIME 2025 pass@128 (%), 30 problems, 128 responses each; contrast in percentage points.

Arm	Step 410	Step 430
Full	70.0	73.3
Shuffled	73.3	73.3
Target	70.0	66.7
Shuffled−Target	3.3	6.7

4.2 CONTROL RL: FUNCTIONAL REUSE AND COMPETING CREDIT EFFECTS

Matched lesions test shared function. Gaussian PPO policies on Meta-World-based manipulation (Yu et al., 2021) let us perturb action means and measure episode success V_A, V_B. Short-to-long pushing tests shared function; spatially split pick-and-place tests credit effects.

From shared base weights, A learns short pushing while the control continues multidirectional pushing for equally many updates. Their mean actions at state s, $\mu_A(s)$ and $\mu_{\text{ctrl}}(s)$, define the historical direction

$$d_A(s) = \mu_A(s) - \mu_{\text{ctrl}}(s). \tag{12}$$

Their Adam initializations also differ (Appendix A.2). Let $\mu_t(s)$ and diagonal $\Sigma_t(s)$ be the tested policy's action mean and covariance. We remove only the component c_A of its displacement from μ_{ctrl} along the positive historical direction. The positive-part operator $[x]_+ = \max(x, 0)$ keeps a positive projection coefficient and sets a nonpositive one to zero. Thus a displacement opposite

to the historical direction is left unchanged. Within the early region of forward object progress $0 \le z < 0.12$, define, with state arguments suppressed,

$$c_A = \left[\frac{d_A^\top \Sigma_t^{-1}(\mu_t - \mu_{\text{ctrl}})}{d_A^\top \Sigma_t^{-1} d_A + \epsilon} \right]_+ d_A, \qquad \mu_t^{\text{hist}}(\lambda) = \mu_t - \lambda c_A, \qquad \mu_{t,j}^{\perp}(\lambda) = \mu_t + \lambda u_j, \quad 0 \le \lambda \le 1. \tag{13}$$

Here $\epsilon = 10^{-8}$ stabilizes the projection. The dose λ is the fraction of c_A removed at evaluation: zero leaves the mean unchanged, 0.5 removes half, and one removes the entire measured component. Model weights and covariance stay fixed; outside the region, means are unchanged.

Choose J control vectors $u_1, \ldots, u_J$ in opposite pairs. After scaling each action coordinate by its standard deviation, these vectors are perpendicular to the historical direction and have the same length as c_A. Thus the same λ sets the displacement size in both arms, while its direction differs. The resulting Gaussian policies $\pi_\lambda^{\text{hist}}$ and $\pi_{\lambda,j}^{\perp}$ have equal pre-clipping KL from the tested policy at each common nondegenerate state; Appendix A.2 gives the construction and experimental choice of J. At each dose, define

$$\text{FD}_B(\lambda) = \frac{1}{J} \sum_{j=1}^{J} V_B(\pi_{\lambda,j}^{\perp}) - V_B(\pi_\lambda^{\text{hist}}). \tag{14}$$

A positive value means that removing this fraction of the historical component hurts B more than an equally sized orthogonal perturbation. Varying λ tests how this excess damage changes with removal strength. Replacing V_B with V_A defines FD_A for the old task.

Across 16 seeds, dose-averaged A-dependence is positive in both source and final policies (Figure 3a). Final-policy B-dependence rises from a near-zero half-dose contrast to 0.0839 [0.0481, 0.1232] at full dose (panel b). Both task contrasts are positive in 13/16 seeds at full dose (Table 6): the historical direction remains useful for A after B training and also contributes to B success.

Old-task dependence is stronger in the early than terminal region ($0.12 \le z < 0.18$).

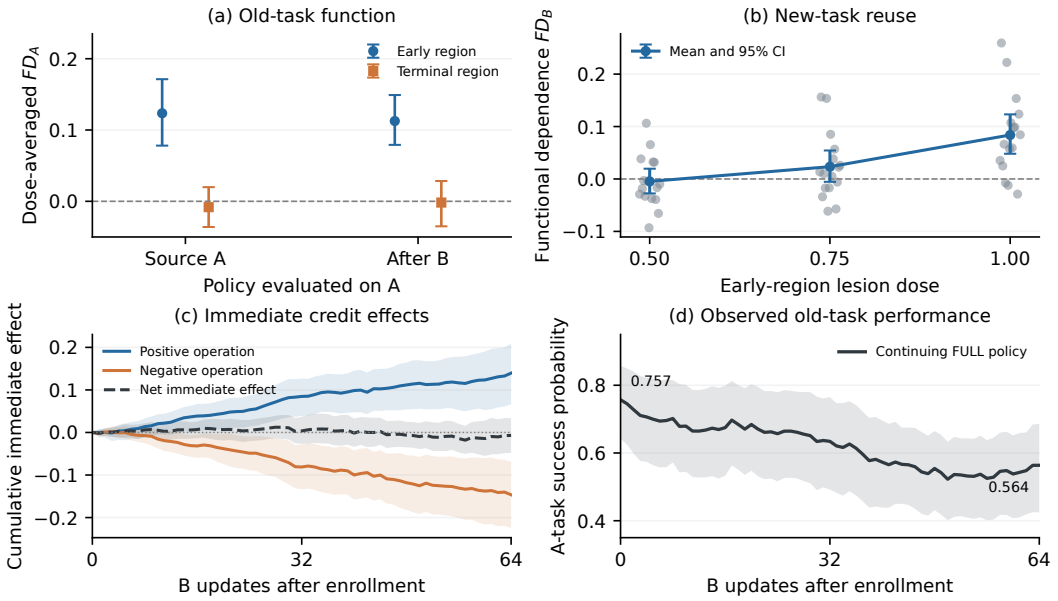

Figure 3: Functional dependence, local support, and observed forgetting. (a,b) Pushing lesions: old-task dependence by region and new-task dependence by dose, across 16 seeds. For the same 17 eligible pick-and-place runs, (c) decomposes cumulative immediate FULL–BOTH_OFF differences; (d) tracks A-success before and after updates along the continuing FULL path. Shading gives pointwise 95% seed-bootstrap intervals.

Testing the link from expression to function. Does preserving a historical readout suffice to preserve performance? The orthogonal controls test functional changes outside this readout (Section 3.2): they preserve the covariance-weighted historical projection at each common state while changing other action components. At full dose in the early region, final policies lose an average 5.73 percentage points [1.77, 9.76] of A-success relative to the unmodified policy, despite this invariant projection. Thus a useful historical component can coexist with functional contributions outside the readout. This explains why SFP separately measures expression and function: predicting retention requires accounting for capability changes the readout misses. Appendix A.3 gives the dose responses and the corresponding prediction-error bound.

New-task credit has competing effects on old function. To understand how old-task support can coexist with forgetting, we separate the effects of positive and negative new-task credit. The pick-and-place study uses 17 runs selected by fixed source-success and B-progress criteria (Appendix A.5).

History masks compare the current policy with a frozen B-only reference (Equation 24). Within each sign group, we shuffle masks across comparable samples and attenuate actor-loss contributions at dose $\lambda = 0.5$. Sibling updates share the shuffle.

Four sibling updates share each saved reference state; only FULL advances the reference path. Let $f_{pn,t}$ be estimated next-update A-success, with $p, n \in \{0, 1\}$ controlling the positive and negative groups, respectively: one retains credit; zero applies shuffled attenuation. Thus $f_{11,t}$ is FULL, and $f_{00,t}$ is BOTH_OFF. The four combinations capture how the two operations interact. Splitting that interaction equally (Shapley, 1952) gives

$$\begin{aligned} \Phi_{+,t} &= \tfrac{1}{2}\big[(f_{10,t} - f_{00,t}) + (f_{11,t} - f_{01,t})\big], \\ \Phi_{-,t} &= \tfrac{1}{2}\big[(f_{01,t} - f_{00,t}) + (f_{11,t} - f_{10,t})\big], \\ \widehat{L}_N &= \sum_{t=0}^{N-1} (\Phi_{+,t} + \Phi_{-,t}) = \sum_{t=0}^{N-1} (f_{11,t} - f_{00,t}). \end{aligned} \tag{15}$$

These allocations include update-magnitude effects. Run-mean sums estimate L_N, with FULL and BOTH_OFF as rules 1 and 0. Intervals resample eligible runs (Appendix A.5).

Over 64 updates, the allocations have opposite signs in 15/17 runs (Figure 3c). Their mean net effect, -0.0070 [-0.0479, 0.0337], is small compared with either component, while mean A-success declines along the same path (panel d).

In aggregate, retaining positive credit supports A success, while negative credit opposes it. The paired updates with both signs attenuated account for most of the decline along the FULL path (Appendix A.5). Thus old-task support remains present during forgetting, alongside other damaging effects.

5 CONCLUSION

We propose implicit rehearsal as a mechanism by which on-policy learning can preserve earlier capabilities: prior behavior enters new-task trajectories, receives support when useful to the new objective, and reappears through subsequent generation. Prior work identifies benefits of on-policy data and offers mode-seeking or KL-based explanations (Appendix C). We introduce a new mechanistic perspective and analytical framework for explaining retention through the reuse and reinforcement of prior learning during new-task training. SFP links credit selection, shared function, and persistence through a selection identity, functional prediction bounds, and a continuation identity connecting local interventions to terminal outcomes.

Complementary LLM and control experiments connect these questions to measurements and interventions. Math-to-Code experiments link credit attenuation to historical expression and later Math performance; control lesions identify shared functional dependence, while paired updates reveal supporting and opposing effects during forgetting. Together, the framework and evidence provide a basis for studying retention through the continued use of prior learning. They direct continual-learning research toward identifying which earlier behaviors new tasks elicit and reward, which capabilities those behaviors support, and how that support evolves during training.

AI USE STATEMENT

Generative AI assisted us in optimizing writing and expression, verifying the correctness of mathematical derivations, and checking submission compliance.

ETHICS STATEMENT

This work uses open-weight OLMo models, public mathematical and code datasets, and simulated Meta-World control environments. Our experiments do not involve human participants or the collection of personal data, and control policies are evaluated entirely in simulation. Applications of retention mechanisms should assess retained behaviors for bias and safety. The underlying models, datasets, and software should be used in accordance with their licenses and usage restrictions.

REPRODUCIBILITY STATEMENT

The supplementary material includes the core code for our language-model and control-RL experiments, covering training, interventions, and evaluation, together with configurations, dependency information, and execution instructions. Appendix A details training and evaluation protocols, statistical units, per-seed measurements, and supplementary results. Appendix B gives the full theoretical statements, assumptions, and derivations.

REFERENCES

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. On-policy distillation of language models: Learning from self-generated mistakes, 2024. URL https://arxiv.org/abs/2306.13649.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 41–48, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

Howard Chen, Noam Razin, Karthik Narasimhan, and Danqi Chen. Retaining by doing: The role of on-policy data in mitigating forgetting, 2026a. URL https://arxiv.org/abs/2510.18874.

Yan Chen, Taojie Zhu, Meng Zhang, Xin Chen, Jiaqi Huang, Dongyang Xu, and Yizhi Wang. On-policy replay for continual supervised fine-tuning, 2026b. URL https://arxiv.org/abs/2605.29495.

Benjamin Ehret, Christian Henning, Maria R. Cervera, Alexander Meulemans, Johannes von Oswald, and Benjamin F. Grewe. Continual learning in recurrent neural networks, 2021. URL https://arxiv.org/abs/2006.12109.

Ian J. Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks, 2015. URL https://arxiv.org/abs/1312.6211.

Hangzhan Jin, Sitao Luan, Tianwei Ni, Sicheng Lyu, Guillaume Rabusseau, Reihaneh Rabbany, Doina Precup, and Mohammad Hamdaqa. Rl fine-tuning heals ood forgetting in sft, 2026. URL https://arxiv.org/abs/2509.12235.

Sham M Kakade. A natural policy gradient. In T. Dietterich, S. Becker, and Z. Ghahramani (eds.), *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL https://proceedings.neurips.cc/paper_files/paper/2001/file/4b86abe48d358ecf194c56c69108433e-Paper.pdf.

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, March 2017. ISSN 1091-6490. doi: 10.1073/pnas.1611835114. URL http://dx.doi.org/10.1073/pnas.1611835114.

Song Lai, Haohan Zhao, Rong Feng, Changyi Ma, Wenzhuo Liu, Hongbo Zhao, Xi Lin, Dong Yi, Qingfu Zhang, Hongbin Liu, Gaofeng Meng, and Fei Zhu. Reinforcement fine-tuning naturally mitigates forgetting in continual post-training, 2026. URL https://arxiv.org/abs/2507.05386.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation, 2023. URL https://arxiv.org/abs/2305.01210.

Mao-Lin Luo, Zhe-Xu Wang, Zi-Hao Zhou, Bo Ye, Jian Zhao, Min-Ling Zhang, and Tong Wei. Rl forgets! towards continual policy optimization, 2026. URL https://arxiv.org/abs/2607.04364.

Michael McCloskey and {Neal J.} Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. *Psychology of Learning and Motivation - Advances in Research and Theory*, 24(C):109–165, January 1989. ISSN 0079-7421. doi: 10.1016/S0079-7421(08)60536-8. The research reported in this chapter was supported by NIH grant NS21047 to Michael McCloskey, and by a grant from the Sloan Foundation to Neal Cohen. We thank Sean Purcell and Andrew Olson for assistance in generating the figures, and Alfonso Caramazza, Walter Harley, Paul Macaruso, Jay McClelland, Andrew Olson, Brenda Rapp, Roger Rat-cliff, David Rumelhart, and Terry Sejnowski for helpful discussions.

Seyed Iman Mirzadeh, Arslan Chaudhry, Dong Yin, Huiyi Hu, Razvan Pascanu, Dilan Gorur, and Mehrdad Farajtabar. Wide neural networks forget less catastrophically, 2022. URL https://arxiv.org/abs/2110.11526.

Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

OpenAI. Learning to reason with LLMs. OpenAI blog, 2024. URL https://openai.com/index/learning-to-reason-with-llms/.

OpenAI. Introducing deep research. OpenAI blog, 2025a. URL https://openai.com/index/introducing-deep-research/.

OpenAI. Expanding on what we missed with sycophancy. OpenAI blog, 2025b. URL https://openai.com/index/expanding-on-sycophancy/.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

Qwen Team. Qwen3.8-max: A new bar for coding and cowork, August 2026. URL https://qwen.ai/blog?id=qwen3.8.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization, 2017a. URL https://arxiv.org/abs/1502.05477.

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

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, 2024. URL https://arxiv.org/abs/2402.03300.

Lloyd S. Shapley. *A Value for N-Person Games*. RAND Corporation, Santa Monica, CA, 1952. doi: 10.7249/P0295.

Idan Shenfeld, Jyothish Pari, and Pulkit Agrawal. Rl's razor: Why online reinforcement learning forgets less, 2025. URL https://arxiv.org/abs/2509.04259.

Idan Shenfeld, Mehul Damani, Jonas Hübotter, and Pulkit Agrawal. Self-distillation enables continual learning, 2026. URL https://arxiv.org/abs/2601.19897.

Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay, 2017. URL https://arxiv.org/abs/1705.08690.

Meng Wang, Haohan Zhao, Wenzhuo Liu, Lu Yang, Geng Liu, Haiyang Guo, Guo-Sen Xie, Gaofeng Meng, Hongbin Liu, and Fei Zhu. Denser $\neq$ better: Limits of on-policy self-distillation for continual post-training, 2026. URL https://arxiv.org/abs/2607.01763.

Xiao Wang, Yuansen Zhang, Tianze Chen, Songyang Gao, Senjie Jin, Xianjun Yang, Zhiheng Xi, Rui Zheng, Yicheng Zou, Tao Gui, Qi Zhang, and Xuanjing Huang. Trace: A comprehensive benchmark for continual learning in large language models, 2023. URL https://arxiv.org/abs/2310.06762.

Xiaomi MiMo Team. Mimo-v2.6-pro-rl. https://huggingface.co/XiaomiMiMo/MiMo-V2.6-Pro-RL, 2026.

Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Avnish Narayan, Hayden Shively, Adithya Bellathur, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning, 2021. URL https://arxiv.org/abs/1910.10897.

A Additional Experiments and Reproducibility Details

This appendix asks how earlier learning remains expressed, whether that expression serves both tasks, and how supporting and opposing updates combine during forgetting. The LLM comparisons track historical outputs and their response to Code credit; pushing lesions test shared function and readout sufficiency; pick-and-place comparisons relate signed credit effects to the observed training path. Training and evaluation details accompany the evidence for each question.

A.1 Language-model protocol and statistical units

The trajectory comparison asks whether Code training erases the Math-associated output component or changes the policy around a component that remains present. Fixed response prefixes allow us to compare the policies on identical inputs, separating changes in their output preferences from changes in the text presented to them. The intervention then asks whether new-task credit helps sustain the measured component.

The base checkpoint is `allenai/Olmo-3-1025-7B`; the Math initialization is `allenai/Olmo-3.1-7B-RL-Zero-Math`; the frozen response generator is `allenai/Olmo-3.1-7B-RL-Zero-Code`. The Math model card explicitly names the chosen base in its `base_model` metadata (Olmo et al., 2025); our Math stage uses that released checkpoint. Both Code paths use GRPO with matched prompts, schedule, hyperparameters, and 500 updates. Each update contains 32 prompts and eight responses per prompt. Training responses have an 8192-token cap; checkpoints and resumable states are saved every 25 updates. The two paths use 16 and 24 GPUs, respectively.

Table 2: Language-model training and evaluation protocol.

Item	Setting
Main training replication	One Math→Code and one Base→Code run
Mechanism prompts	100 held-out prompts, disjoint from both self-training paths
Main Math evaluation	AIME 2025, 30 problems, 32 responses/problem; $T = 0.6$, top-$p = 0.95$
Code evaluation	HumanEval (164 prompts) and MBPP (378 prompts), base and strengthened tests; greedy
Evaluation length	16384 tokens; Math template `olmo_thinker_rlzero`, Code template `olmo_thinker_code_rlzero`
Readout aggregation	Position first, then within prompt, then equal prompt weights

Official solutions and Code-model responses share the same prompts and supply two fixed prefix sets. The first 13 response-template tokens are excluded from Code-model responses; official solutions use their full token range. Code-model responses include 97 normally terminated and three length-truncated outputs; valid-position coverage is at least 99.99%. Historical trajectory measurements use bf16-exported models with an fp32 output head and float64 aggregation.

Historical geometry and its decomposition. A falling cosine can mean that an earlier component is shrinking, that other components are growing, or both. We separate these possibilities before interpreting alignment as evidence about forgetting. For each saved prefix s, define the acquisition difference a, current training-path difference d_t, and current probability weight W_t:

$$\begin{aligned} a(s) &= z_A(s) - z_0(s), \qquad d_t(s) = z_{AB,t}(s) - z_{B,t}(s), \\ p_t(s) &= \pi_{AB,t}(\cdot \mid s), \qquad W_t(s) = \mathrm{diag}(p_t(s)) - p_t(s)p_t(s)^\top. \end{aligned} \tag{16}$$

$M_t(s)$ measures the size of the history difference; $\kappa_t(s)$ is its cosine similarity to the initial Math difference in the same metric:

$$M_t(s) = \|d_t(s)\|_{W_t(s)}, \qquad \kappa_t(s) = \frac{\langle a(s), d_t(s)\rangle_{W_t(s)}}{\|a(s)\|_{W_t(s)}\,\|d_t(s)\|_{W_t(s)}}. \tag{17}$$

To distinguish loss of the initial component from growth in other directions, let $P_t(s)$ be the signed length along acquisition, and $Q_t(s)$ the orthogonal length:

$$P_t(s) = \frac{\langle a(s), d_t(s)\rangle_{W_t(s)}}{\|a(s)\|_{W_t(s)}}, \qquad Q_t(s) = \sqrt{\max\{M_t(s)^2 - P_t(s)^2, 0\}}. \tag{18}$$

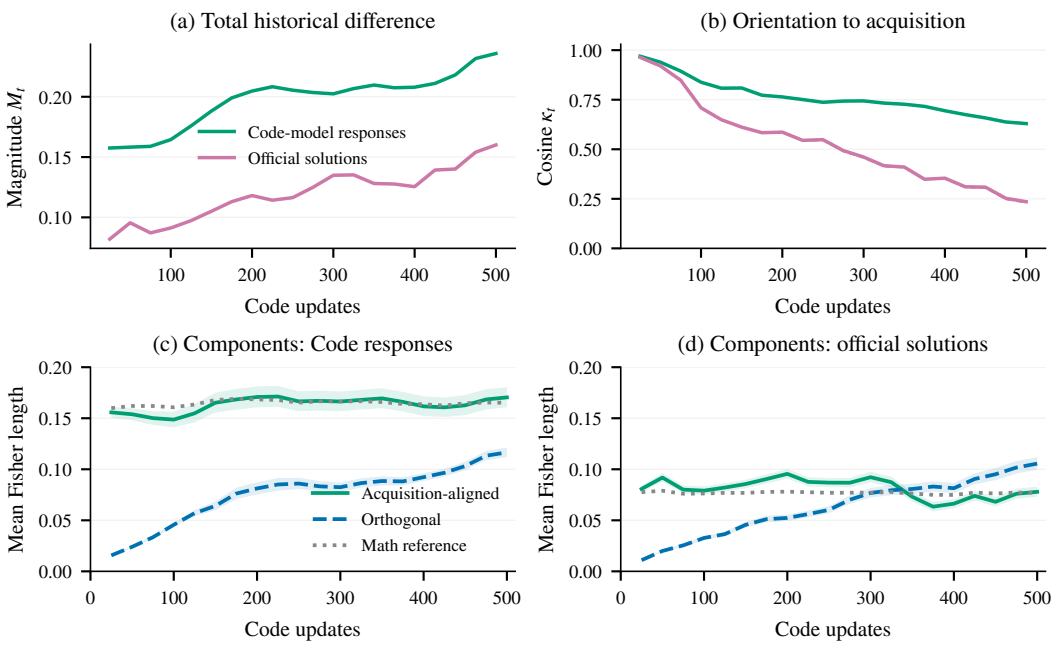

Figure 4: **Decomposing historical expression on both fixed response sets.** (a) Total difference magnitude. (b) Alignment with acquisition; dashed curves use frozen Math-model Fisher weights. (c,d) Signed acquisition-aligned and orthogonal lengths, computed at each position before averaging. Dotted lines give the acquisition reference length under current weights. Bands resample whole probe prompts; they quantify probe sampling, conditional on the single pair of training runs.

All operations precede averaging within prompts and then equally across prompts. The exported decomposition sets $P_t(s) = 0$ when the acquisition norm is at most 10^{-4}; the cosine excludes positions with either norm below this threshold, using numerical stabilizer 10^{-12}. The ratio reported in Section 4.1 divides the mean of P_t by the mean acquisition norm; it is not an average of position-wise ratios. Because lengths are averaged after decomposition, the squared mean lengths need not satisfy a Pythagorean identity.

Figure 4 shows a largely maintained acquisition-aligned component alongside growing orthogonal differences. On Code-model responses, the aligned length stays near the acquisition reference while the orthogonal length grows substantially. Official solutions show the same qualitative decomposition at a different absolute scale. Thus new-task learning develops additional output differences without a corresponding disappearance of the initial component.

The fixed-metric comparison checks whether changing probability weights alone accounts for the lower alignment. Current and frozen Math-model weights give similar cosine trajectories, so the decrease remains visible under a fixed metric. Together, the panels show why a decreasing cosine need not mean that the historical component has been erased: its share of the total difference can shrink as other components grow. This identifies a persistent historical component for the subsequent credit intervention to track.

Credit interventions and fixed-reference measurement. The TARGET intervention has maximum attenuation 0.5, using the clipped positive historical score from Section 2. The calibration threshold is the 95th percentile of positive historical scores on an independent calibration batch. Its 476 usable responses contain 1,838,467 valid tokens: 55.68% have positive ℓ, and $\tau_{\text{hist}} = 0.7785$ nats. Thus many tokens sampled for the new task retain a historical preference; positive actor credit determines which are attenuated. Positive-credit tokens are stratified by prompt/response group, position, log probability, and credit magnitude before masks are permuted for SHUFFLED; zero masks participate in permutation. During continuation, token scores compare each arm's current pre-update policy with the frozen B-400 reference, using the same calibration threshold. Masks and credits are recomputed before each update and held fixed within that update. The single-update experiment has

eight reset batches, indexed 401–408 by their prompt-schedule positions, and three sibling updates per batch. Those labels do not denote a sequential training trajectory.

All 24 single-update and 38 continuation readouts in the supplied summaries use the same AB-400 direction, AB-400 Fisher weights, B-400 reference, and 100 branch-probe prompts. They instantiate Equation 2 with $t^* = 400$, rather than subtracting a changing B checkpoint. The valid-position fraction is 0.999976; intervention readouts use fp32 models. For paired arm contrasts the common reference cancels algebraically. The branch states themselves, and therefore the generated training trajectories, differ across arms.

Table 3: Single-update comparisons. The credit gap and KL ratio compare TARGET with SHUFFLED; policy KL is measured on the separate fixed Code-model response set.

Batch	Full E	Shuffled E	Target E	Credit gap (%)	KL ratio
401	1.00056	0.99928	0.99385	0.152	5.26
402	1.00004	0.99869	0.99401	0.077	4.02
403	1.00104	0.99986	0.99553	0.043	3.55
404	1.00062	0.99925	0.99407	0.008	6.35
405	1.00030	0.99921	0.99434	0.065	4.15
406	0.99967	0.99825	0.99229	0.207	5.79
407	1.00056	0.99888	0.99384	0.016	5.76
408	1.00053	0.99956	0.99497	0.079	4.05

What the repeated comparisons establish. Attenuating history-weighted positive Code credit lowers the expression retained by the next update. Each batch compares updates from the same starting state, and both controls leave more expression than TARGET in all eight batches. The comparison with FULL tests removing the selected contributions; the comparison with SHUFFLED changes where similar removed credit mass is allocated. Their common ordering supports a role for the implemented credit weighting in sustaining historical expression across rollout batches. Because the direction, probe prefixes, and reference weights are fixed, the difference reflects the updated policy's response along the same measured component.

Table 3 also separates credit-mass matching from policy-change matching. The small removed-credit gaps show that the attenuated arms suppress similar total mass, while the larger TARGET KL shows that the resulting policy changes differ in magnitude. The expression ordering therefore describes the implemented updates, including their induced drift. This supports the conclusion that the credit-weighting rule changes historical expression; it does not isolate a contrast at equal policy KL. Intervals use 10000 paired bootstrap draws with seed 20260925: whole rollout batches for single updates and whole probe prompts for continuation contrasts. Probe-bootstrap bands are pointwise percentile intervals.

Consistency across probe prompts. Does the mean expression gap describe the probe set broadly, or only a few large responses? We reconstruct each prompt's expression by averaging its valid positional readouts, retaining all 100 Code probe prompts. At step 410, both FULL−TARGET and SHUFFLED−TARGET lie above zero for every prompt (Figure 5a). Attenuation therefore reduces expression across the measured set, while the distribution's spread quantifies how strongly each prompt responds. The SHUFFLED−TARGET ordering also holds throughout the probe set at step 430, following the optimizer restart at 410 (panel b). These distributions establish the breadth of the response within the observed branches; the reset-batch experiment separately establishes its recurrence across update batches.

Historical preferences within Code responses. To see where training history appears in generated text, we compare the two policies' probabilities of the same token given the same prefix. At update 400, the historical score ℓ in Equation 1 is positive when Math→Code assigns higher probability and negative when Base→Code does. Figures 6–10 show 20 longer excerpts from distinct Code prompts, spanning algorithmic reasoning, implementation details, and checks of task requirements.

Examples 01 and 05 compare candidate solutions to parity and matching problems; Examples 12 and 17 revisit field mappings and regular-expression constraints. Within these passages, positive

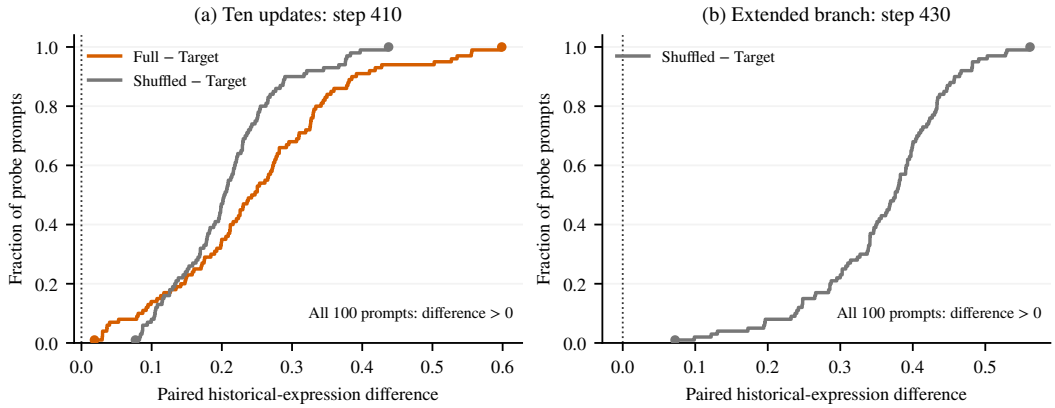

Figure 5: **Expression differences across the complete Code probe set.** Empirical cumulative distributions of paired prompt-level contrasts after (a) ten branch updates and (b) the extension to step 430. Every curve contains all 100 prompts. All observed differences lie to the right of zero; their spread shows prompt heterogeneity. The two extended branches restart their optimizers at step 410.

and negative preferences occur in adjacent parts of the same argument. Training history therefore changes which local outputs are favored, rather than assigning a uniform preference to the entire response. This variation motivates token-level historical masks. The heatmaps locate history-associated expression; the sibling-update experiment separately tests how new-task credit affects its maintenance.

The excerpts come from the same independent score-only rollout batch at checkpoint 400, without a parameter update. The export selects one response per distinct prompt in seeded random order (seed 20260926), then chooses a window using text-based eligibility and sentence-boundary heuristics, without filtering by historical scores or advantage. We display all 20 exported windows, containing 114–141 tokens each, including their original code fragments. Words and scores are preserved; whitespace is simplified for display and lines wrap at word boundaries. Every page uses the same symmetric color scale, set to the 98th percentile of absolute scores over the complete source dump. Example headers give zero-based token positions in the original response.

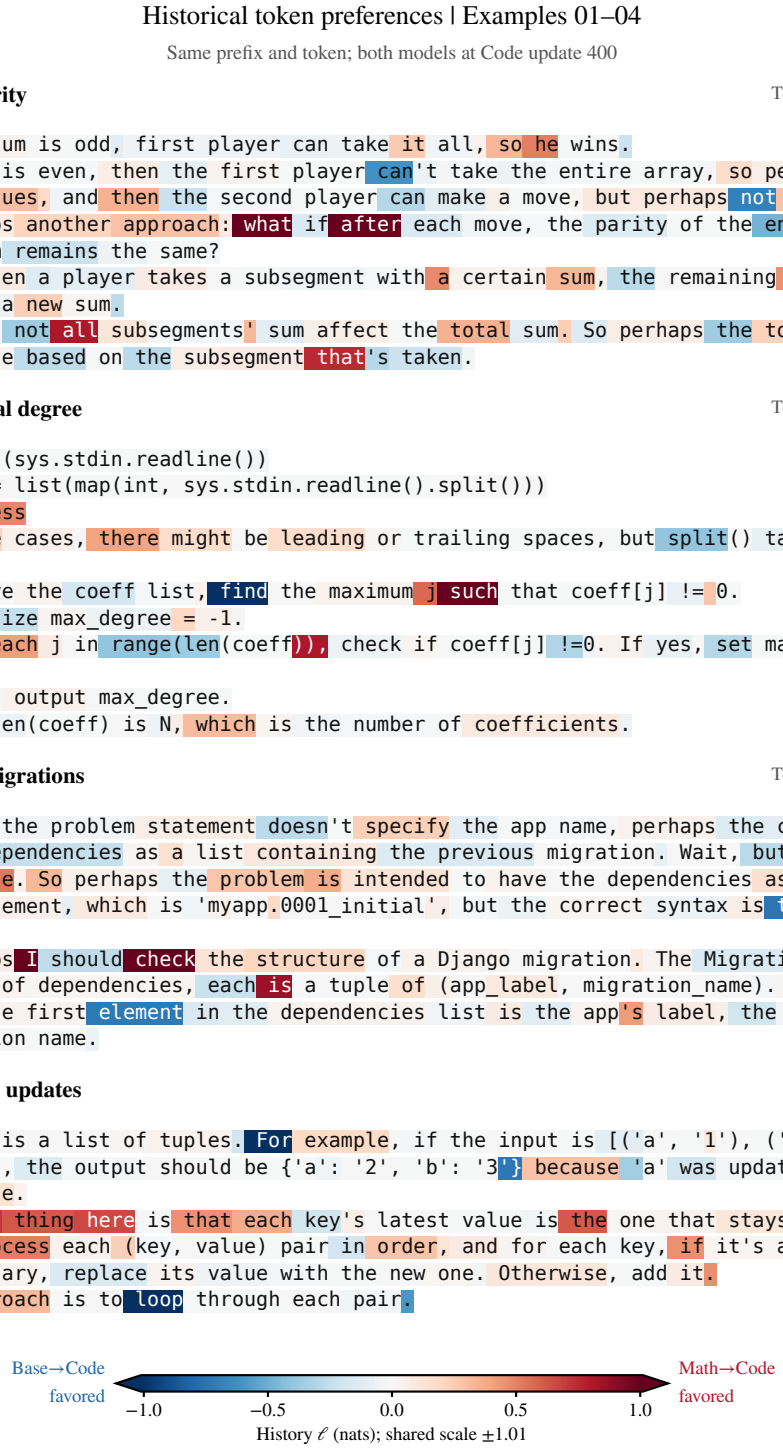

Figure 6: **Historical token preferences, examples 01–04.** Red favors Math→Code and blue favors Base→Code at the same prefix; near-white indicates similar probabilities. All gallery pages use the same History ℓ scale at update 400. Headers identify the task and the excerpt's token positions. Text and token scores are preserved.

Historical token preferences | Examples 05–08

Same prefix and token; both models at Code update 400

05 Grid matching Tokens 2113–2241

But implementing Hopcroft-Karp is a bit involved. Alternatively, for each W, try to find a P that is not yet matched, starting from the first possible.
But perhaps a better approach is to model the problem as a bipartite graph and then use a backtracking approach (DFS) to find the maximum matching.
Alternatively, since the problem is small, maybe a dynamic programming approach is feasible, but that's probably not.
Alternatively, perhaps we can represent the bipartition as left nodes (W positions) and right nodes (P positions), and then for each W, try to match it to a P in the set.

06 Alarm-clock selection Tokens 2546–2666

Initialize a selected array, and a pointer to the next a_i to process.
For each a_i in sorted order:
  if selecting this a_i would cause a window of m consecutive a_i's to have more than k-1 in the selected, then this a_i cannot be selected.
Thus, perhaps in this way, we can greedily select the a_i's in such a way that the condition is satisfied.
But perhaps a better approach is to precompute for each position which a_i can be in the selected set such that no window of m consecutive a_i's has more than k.

07 Road fares Tokens 393–509

So the fee becomes normal_fare - some value, but the final is maximum 0 (unlikely, since it's rounded up perhaps?) Wait, no, the halved amount might not be a multiple of 50, so we have to round up to the next multiple of 50, or maybe just round to 50 yen bank if not.
Wait, the example shows that the fare is normally 450, which is 9 sections of 50 (9 *50=450). Wait, wait. Wait, that's an important point. What is the normal fare?

08 Payment aggregation Tokens 2292–2427

Approach
1. **Initialize a Dictionary**: Start with an empty dictionary to keep track of the payment amounts for each customer. This dictionary will store the final amounts after all operations are processed.
2. **Process Each Operation**: Iterate through each operation in the list.
  - For 'add' and 'update' operations, extract the customer ID and the amount. Update the payment amount for the corresponding customer in the dictionary.
  - For 'get' operations, simply check the current payment amount for the customer, which is already stored in the dictionary.
3. **Return the Result**: After processing all operations, return the dictionary containing the final payment amounts for each customer.

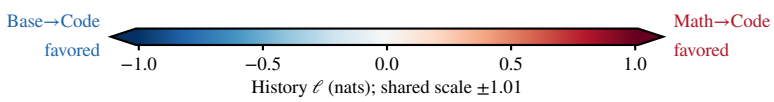

Figure 7: **Historical token preferences, examples 05–08.** Matching, alarm-clock selection, road fares, and payment aggregation. Colors and token positions follow Figure 6.

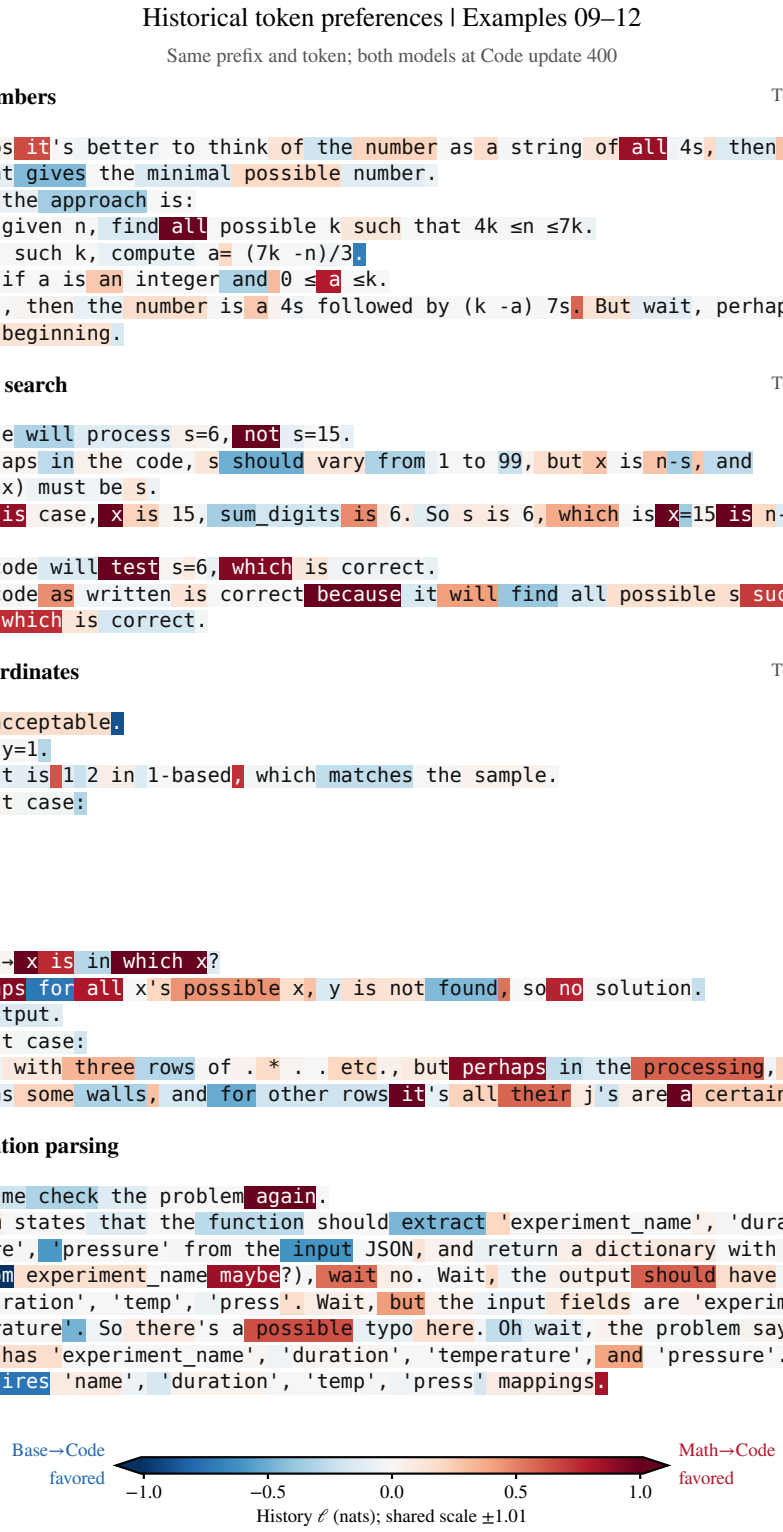

Figure 8: **Historical token preferences, examples 09–12.** Lucky numbers, digit-sum search, depot coordinates, and configuration parsing. Colors and token positions follow Figure 6.

Historical token preferences | Examples 13–16

Same prefix and token; both models at Code update 400

13 Moves on a tree Tokens 1600–1713

```
 Wait, no, that's not correct because each move is moving two pieces, but perhaps I
need to make all other S_i zero except for x.
Wait, perhaps it's better to model the problem as: the sum of all the S_i must be
equal to the target's count. Because all pieces are moved to x, so the sum of all S_i
is K, which is the target. So, perhaps x must have a count equal to K. So the sum of
all S_i is K, which must be exactly the count on x.
```

14 Tour groups Tokens 1175–1312

```
So perhaps, for each group, the process is to compute the maximum possible time when a
new 'A' is added.
Let's think of the steps:
Each minute, the current state is a string. Then, for each position i, if s[i] is 'P'
and s[i+1] is 'A', then in the next state, s[i] becomes 'A'. Wait, perhaps not. Let's
perhaps look at the example.
In the first example, the initial string is 'PPAP' which is P P A P. So the students
are positions 0: P, 1: P, 2: A, 3: P.
```

15 Masha and bears Tokens 233–356

```
 Wait, no, let's see the problem statement again.
Wait, the problem says:
Masha came to test. She could climb into all cars, but liked only the smallest car. So
the smallest car must be at least Vm, else she couldn't climb. But she likes it only
if the smallest car is >= 2*Vm. Wait, no, maybe not exactly. Let's see:
The problem states:
Masha can climb into all cars (since there's no size restriction for her—she's size
Vm, which is the fourth input). But she liked only the smallest car.
```

16 Hangman Tokens 5165–5298

```
### Approach
1. **Input Reading**: Read the word and the permutation. The word is a string of
uppercase letters, and the permutation is a string of all 26 letters of the alphabet.
2. **Set Conversion**: Convert both the word and the permutation into sets. The set of
the word contains all unique letters, and the set of the permutation contains all 26
letters.
3. **Subset Check**: Check if all elements in the permutation set are a subset of the
word set. This is done using the `issubset` method.
4. **Result Determination**: If all letters in the permutation are present in the
word, Ned wins.
```

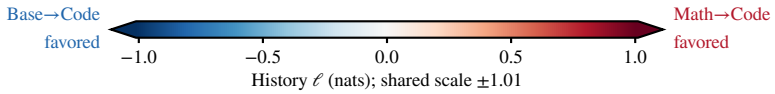

Figure 9: **Historical token preferences, examples 13–16.** Tree moves, tour groups, the Masha-and-bears problem, and Hangman. Colors and token positions follow Figure 6.

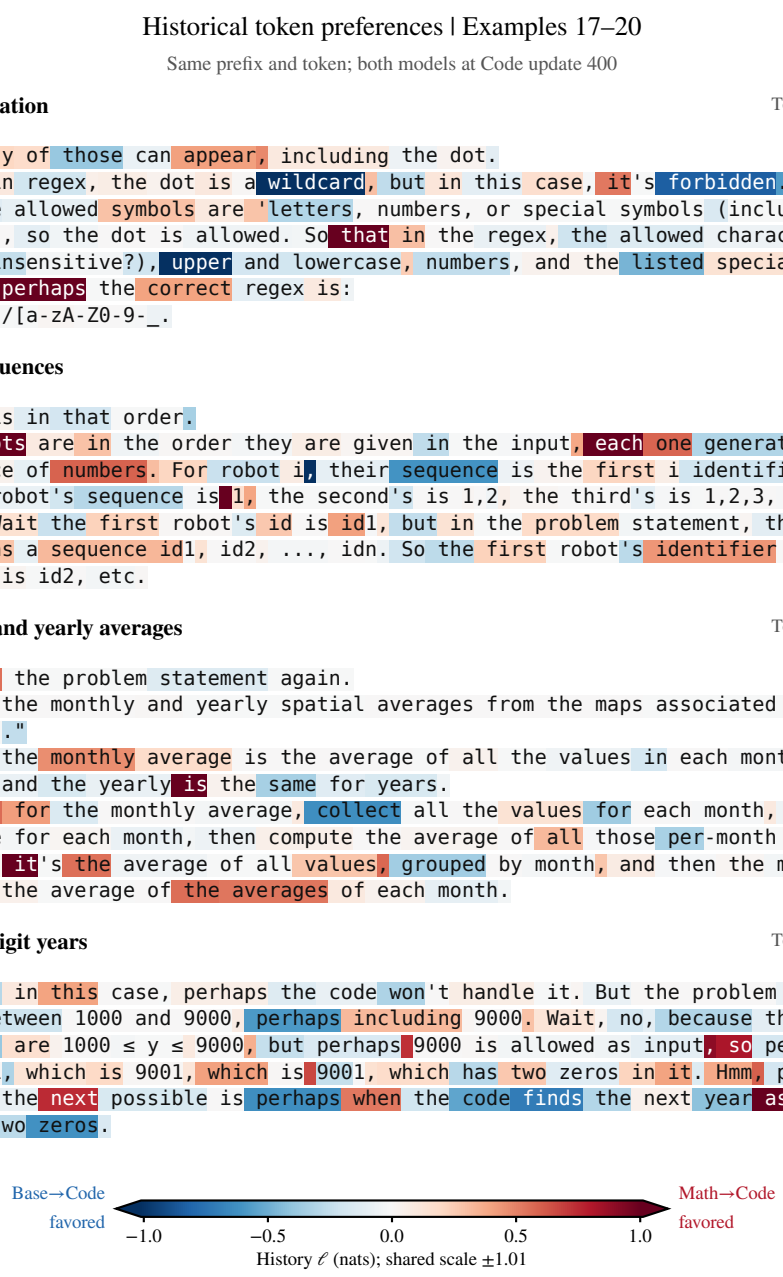

Figure 10: **Historical token preferences, examples 17–20.** DOI validation, robot sequences, temporal aggregation, and distinct-digit years. Colors and token positions follow Figure 6.

A.2 FUNCTIONAL LESIONS IN PUSHING

Task geometry and training configuration. The environment is based on `push-v3-goal-observable`, retaining its reward and success signal. A and B place the goal 0.15 and 0.25 units along positive y from the initial object position, respectively, at height 0.02. Base-task goals are proposed at a uniform angle with radius uniform on $[0.13, 0.17]$. Proposals outside $[-0.20, 0.20] \times [0.43, 0.92]$ are rejected; proposals with larger absolute y than x displacement are additionally rejected with probability 0.33. Sampling makes at most 64 attempts, then uses the last proposal. Thus the implemented base sampler includes directional reweighting.

Table 4: Pushing training configuration shared by the base, A, control, and B stages.

Item	Setting
Environment / policy	Horizon 300; 39 observations, four actions; separate actor and critic, each with two width-256 hidden layers
Rollout / optimization	Eight environments, 2048 steps per environment; minibatch 4096; four PPO epochs
Optimizer / objective	Adam, learning rate 3×10^{-4}; discount 0.99; GAE 0.95; PPO clipping 0.2
Loss / clipping	Rollout-wide advantage normalization; value coefficient 0.5; entropy coefficient zero; joint actor–critic gradient norm capped at 0.5
Actions / success	Clip sampled Gaussian actions to $[-1, 1]$; score their pre-clipping values; episode success records whether success occurred at any step

Training lineage and lesion definition. In pushing, the archived lineage first trains a base policy for 70 updates on a multidirectional pushing task. From its weights, the A branch learns short pushing in the positive y direction for 50 updates; the control branch continues the base task for 50 updates. Both use the frozen base observation normalizer. A training starts fresh Adam moments; the control carries the base optimizer state. Thus μ_{ctrl} is a base-task continuation reference matched in update count, and $d_A = \mu_A - \mu_{\mathrm{ctrl}}$ includes the combined effect of these training histories. The final B policy continues from A for 24 updates on longer pushing, carrying A's optimizer state; lesions use that final checkpoint. At environment state s, $\mu_A(s)$ and $\mu_{\mathrm{ctrl}}(s)$ denote the two reference policy means. For each evaluated policy, denote its mean and covariance by $\mu_t(s), \Sigma_t(s)$, suppressing state arguments below. Set $\epsilon = 10^{-8}$, $x_A = \Sigma_t^{-1/2} d_A$, $x_t = \Sigma_t^{-1/2}(\mu_t - \mu_{\mathrm{ctrl}})$, and

$$\rho_t = \frac{x_t^\top x_A}{\|x_A\|^2 + \epsilon}, \quad c_A = [\rho_t]_+ d_A, \quad \mu_t^{\mathrm{hist}}(\lambda) = \mu_t - \lambda c_A \tag{19}$$

inside the frozen prefix region $0 \le z < 0.12$, with z denoting forward object progress. This is the covariance-weighted projection in Equation 13: whitening divides each action coordinate by its policy standard deviation. The dose scales the removed component, so $\lambda = 0.5$ subtracts half of c_A and $\lambda = 1$ subtracts all of it. These are evaluation-time mean changes; the model weights, covariance, critic, normalizer, environment, and action clipping remain unchanged. Outside the region, every policy uses the original mean.

Constructing the orthogonal policies. The reported experiments use four-dimensional actions, whose orthogonal complement to the whitened historical vector x_A has three dimensions. We use its full basis and both signs, giving $J = 6$ controls; this is the experimental configuration, rather than a requirement of the functional-dependence definition. For $\|x_A\| > 10^{-6}$, QR-orthogonalize its normalized vector followed by the four coordinate unit vectors. The final three columns give unit, mutually orthogonal vectors q_k, with $q_k^\top x_A = 0$. Match their lengths to the whitened historical component and transform back to action coordinates:

$$u_{2k-1} = \Sigma_t^{1/2} \|\Sigma_t^{-1/2} c_A\| q_k, \qquad u_{2k} = -u_{2k-1}, \qquad k = 1, \ldots, J/2. \tag{20}$$

The six controls sample from $\mathcal{N}(\mu_t + \lambda u_j, \Sigma_t)$ before the usual action clipping. Thus each pair perturbs in opposite directions, using the same dose as the historical lesion. At the common state, each direction satisfies

$$d_A^\top \Sigma_t^{-1} u_j = 0, \qquad u_j^\top \Sigma_t^{-1} u_j = c_A^\top \Sigma_t^{-1} c_A. \tag{21}$$

The first equality preserves the historical projection; the second gives both interventions the same pre-clipping Gaussian KL from the unmodified policy, $\lambda^2 c_A^\top \Sigma_t^{-1} c_A/2$. Increasing λ therefore scales both displacement lengths together, with KL growing quadratically. At $\|x_A\| \leq 10^{-6}$, the implementation sets orthogonal shifts to zero and retains the stabilized historical projection above. Each policy induces its own evaluated trajectories; matching applies to pre-clipping action distributions at a common nondegenerate state, not whole trajectories.

The precision evaluation of pushing lesions uses the same 16 final policies, seeds 7000–7015, with 1000 episodes per seed, dose, and condition; the original evaluation uses 30. Its horizon is 300 and its vectorization uses six environments. Bootstrap resamples the 16 seeds 5000 times.

Table 5: B-functional dependence under the pushing lesion. Intervals describe the mean across 16 seeds. The mid-high aggregate averages the three listed doses, as specified by the reported protocol.

Dose	Mean FD_B	Median	95% interval of mean
0.50	−0.0046	−0.0132	[−0.0276, 0.0196]
0.75	0.0234	0.0116	[−0.0054, 0.0543]
1.00	0.0839	0.0753	[0.0481, 0.1232]
Mid-high aggregate	0.0342	0.0217	[0.0092, 0.0629]

Table 6 places functional dependence alongside absolute success for all 16 seeds and the three primary doses. This distinction matters because a positive orthogonal-minus-historical contrast can coexist with damage from both perturbations. The baseline shows the total loss; the orthogonal controls show how much of that loss is specific to the chosen direction at matched local magnitude. A evaluations come from the original pushing lesion protocol and B from its precision evaluation. Success-drop intervals use 10000 paired seed resamples with generator seed 20260925.

How the dose response identifies shared function. The functional-direction argument in Section 3.2 predicts that perturbation size alone need not determine damage. Figure 11 tests this by plotting the within-seed orthogonal-minus-historical success contrast, rather than comparing separate marginal success intervals. The historical and orthogonal shifts have equal pre-clipping Gaussian KL at a common nondegenerate state. Positive contrasts therefore identify greater damage from removing the historical component at this matched local magnitude.

In the early region, the contrast grows from half to full dose in both A evaluations, and the final policy's full-dose B contrast is also positive. Thus continued B training retains dependence on an A-derived component, and B itself uses that component. The terminal-region curves have a different profile. Functional importance depends on where the component acts, as well as on its historical origin. The original B evaluation and its precision extension are shown separately; the extension increases evaluation precision without adding training seeds.

Shared dependence is also visible within individual policies: at full dose, both task contrasts are positive in 13/16 seeds. Pairing by seed rules out an explanation in which the positive A and B averages arise solely from disjoint groups of runs. Jointly positive estimates become more common at larger doses, showing why intervention strength belongs in the interpretation. These are signs of seed-level point estimates, rather than per-seed significance tests.

Regional comparison of old-task function. Does a historical component have the same functional importance wherever it is expressed? The same pushing lesion protocol evaluates the adjacent terminal region $0.12 \leq z < 0.18$, declared in the intervention code alongside the early region. Figure 3a compares both regions under the original 30-episode protocol.

The paired early-minus-terminal comparison shows greater old-task dependence in the early region. We reconstruct the three-dose means within each seed before subtracting regions, so this compares where a component matters within the same trained policy. The result shows why provenance alone does not determine functional importance: the same historical construction has different effects depending on where in the behavior it is removed. SFP therefore evaluates function in specified contexts, alongside historical origin. The contrast includes changes in visited trajectories under the region-gated interventions. Its exploratory interval uses 10000 paired seed resamples with seed 20260925; the original marginal intervals use 2000 resamples with seed 0.

Table 6: Early-region pushing success probabilities across 16 seeds. Unmodified success, historical-lesion success, and the mean of six orthogonal controls are shown at the three doses in the main comparison. The drop is unmodified minus historical-lesion success, with a paired seed-bootstrap 95% interval. A evaluations use 30 episodes per condition; final-policy B evaluations use 1000.

Policy / task	Dose	Unmodified	Historical	Orthogonal	Drop [95% interval]
Source on A	0.50	0.5708	0.4625	0.5344	0.1083 [0.0542, 0.1625]
Source on A	0.75	0.5708	0.4000	0.5045	0.1708 [0.1083, 0.2417]
Source on A	1.00	0.5708	0.2646	0.4587	0.3063 [0.2396, 0.3771]
Final on A	0.50	0.5562	0.5104	0.5542	0.0458 [-0.0021, 0.0958]
Final on A	0.75	0.5562	0.4083	0.5181	0.1479 [0.0792, 0.2229]
Final on A	1.00	0.5562	0.3146	0.4990	0.2417 [0.1708, 0.3187]
Final on B	0.50	0.4502	0.3852	0.3806	0.0649 [0.0434, 0.0864]
Final on B	0.75	0.4502	0.3105	0.3339	0.1397 [0.1038, 0.1765]
Final on B	1.00	0.4502	0.2131	0.2969	0.2371 [0.1854, 0.2900]

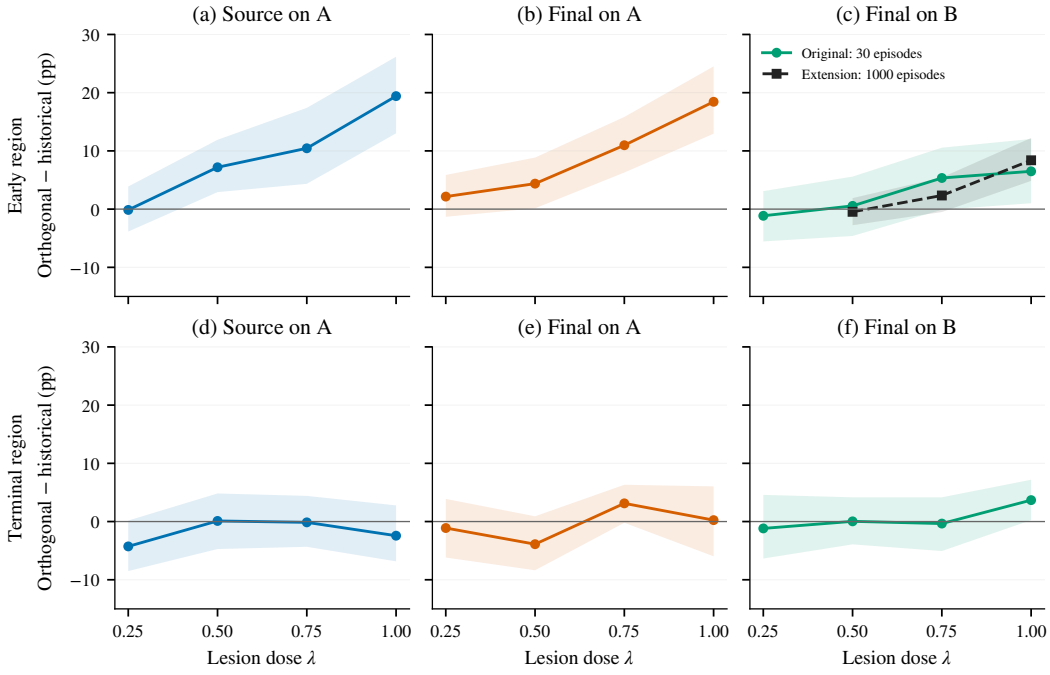

Figure 11: **Functional direction at matched local perturbation size.** Orthogonal-control success minus historical-lesion success, in percentage points, for both regions and all four original doses. Positive values indicate greater damage from the historical lesion. Colored curves use 30 episodes per condition; the dashed B curve separately shows the 1000-episode extension at its three available doses. Bands are pointwise 95% intervals from 10000 paired resamples of the same 16 training seeds (seed 20260925). Magnitude matching concerns pre-clipping Gaussian KL at common non-degenerate states. Absolute success values at the primary doses remain in Table 6.

A.3 An observable test of finite readout sufficiency

To test whether one projection captures all task-relevant changes, keep that projection fixed while perturbing other directions. For each tested pushing policy, freeze the historical direction d_A, its action covariance Σ, the reference mean μ_{ctrl}, and a probe-state distribution $\mathcal{D}_E$. Let ρ_π be Equation 19 with only the tested mean replaced by μ_π. Define

$$E_{\text{ctrl}}(\pi) = \text{Avg}_{s\in\mathcal{D}_E}\,\rho_\pi(s), \qquad D_A^{\perp}(\lambda) = V_A(\pi) - \frac{1}{J}\sum_{j=1}^{J} V_A(\pi_{\lambda,j}^{\perp}). \tag{22}$$

Here E_{ctrl} measures the historical projection, while $D_A^{\perp}$ measures success lost under orthogonal perturbations. The specified orthogonal shifts $\delta_j = \mu_{\pi^{\perp}_{\lambda,j}} - \mu_\pi$ satisfy $d_A^\top \Sigma^{-1} \delta_j = 0$ at every common state; outside the region and at degenerate states they are zero. Hence E_{ctrl} is invariant under the mathematical construction. This concerns pre-clipping means and fixed probes, not averages over each policy's visited-state distribution.

Proposition 1 turns this comparison into a prediction statement. At full dose in the early region, final policies lose A success under orthogonal perturbations even though the historical projection is unchanged (Table 7). A predictor using only that projection must assign the same prediction to policies with different success rates. Half the absolute population success contrast lower-bounds its worst-case error, averaged across seeds; the table gives the corresponding estimates and uncertainty. The conclusion concerns predictors receiving this projection alone, with each seed allowed its own predictor. It does not require the sharp minimax error to be estimated.

Figure 12 makes the sufficiency test visible: every plotted contrast compares policies with the same historical projection. The early-region success gap increases from half to full dose in both stages, whereas the terminal-region estimates follow a different profile. Thus preserving this projection does not fix task performance; the unmeasured directions have context-dependent functional consequences. Table 7 preserves both policy stages, both regions, and all four available doses. Intervals use 10000 paired seed resamples (seed 20260925); they are marginal, exploratory intervals. These are behavioral policy interventions. The finite-set error bound applies directly, without assuming that PPO realizes the same displacements.

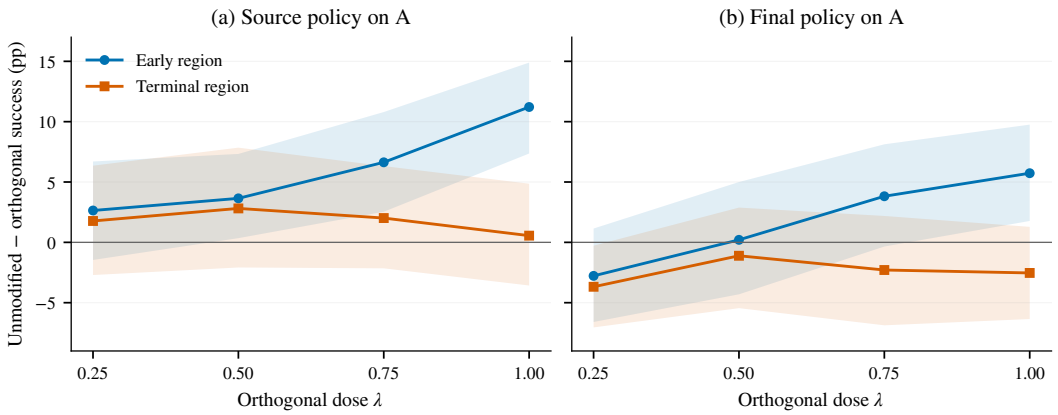

Figure 12: **Equal historical projection does not fix old-task success.** Unmodified A success minus the mean of six projection-preserving orthogonal controls, in percentage points. A nonzero population contrast of either sign indicates functional change outside this readout. Bands are pointwise 95% paired seed-bootstrap intervals. Under the specified invariant projection, half the absolute population mean contrast lower-bounds the mean within-seed worst-case error of any predictor using only that readout (Proposition 1).

The orthogonal controls serve two connected purposes. Relative to the historical lesion, they test whether removing the historical direction causes extra damage. Relative to the unmodified policy, they test whether task-relevant changes exist outside the measured projection. Both can hold: a historical component can be useful while other output directions also matter. This is the empirical motivation for separating historical expression from functional coverage in SFP.

A.3.1 Signed dose responses distinguish local sensitivity from finite effects

A success drop under a projection-preserving perturbation can reflect sensitivity along an unmeasured direction, curvature, or both. The archived positive and negative controls let us separate these responses. For one policy and evaluation task, let $V_j(a)$ denote success after adding the signed mean displacement $au_j(s)$, where u_j is the fixed region-gated, norm-matched orthogonal direction

Table 7: Exploratory test of projection sufficiency on A. Differences compare unmodified success with all six orthogonal controls over 16 seeds. Their historical mean projection is invariant under the specified construction. A positive population difference supplies half its value as a lower bound on mean worst-case projection-only prediction error. Intervals estimate the paired success difference; both regions and all four available doses are retained.

Policy / task	Region	Dose	Difference [95% interval]	Half-difference
Source on A	Early	0.25	0.0264 [-0.0146, 0.0670]	0.0132
Source on A	Early	0.50	0.0365 [0.0035, 0.0733]	0.0182
Source on A	Early	0.75	0.0663 [0.0253, 0.1080]	0.0332
Source on A	Early	1.00	0.1122 [0.0736, 0.1490]	0.0561
Source on A	Terminal	0.25	0.0177 [-0.0271, 0.0635]	0.0089
Source on A	Terminal	0.50	0.0281 [-0.0208, 0.0785]	0.0141
Source on A	Terminal	0.75	0.0201 [-0.0215, 0.0632]	0.0101
Source on A	Terminal	1.00	0.0056 [-0.0358, 0.0486]	0.0028
Final on A	Early	0.25	-0.0278 [-0.0660, 0.0115]	-0.0139
Final on A	Early	0.50	0.0021 [-0.0431, 0.0500]	0.0010
Final on A	Early	0.75	0.0382 [-0.0035, 0.0813]	0.0191
Final on A	Early	1.00	0.0573 [0.0177, 0.0976]	0.0286
Final on A	Terminal	0.25	-0.0368 [-0.0705, -0.0028]	-0.0184
Final on A	Terminal	0.50	-0.0111 [-0.0545, 0.0288]	-0.0056
Final on A	Terminal	0.75	-0.0229 [-0.0688, 0.0219]	-0.0115
Final on A	Terminal	1.00	-0.0253 [-0.0635, 0.0128]	-0.0127

and $j \in \{1, 2, 3\}$. All directions share the unmodified value $V_j(0) = V_0$. At positive dose λ, define

$$O_j(\lambda) = \frac{V_j(\lambda) - V_j(-\lambda)}{2}, \qquad C_j(\lambda) = \frac{V_j(\lambda) + V_j(-\lambda)}{2} - V_0. \tag{23}$$

The odd response O_j measures which sign performs better. The even response C_j measures their average change from the unmodified policy: a negative value means that the two signs together reduce success. Under local smoothness, their leading terms are $\lambda V_j'(0)$ and $\lambda^2 V_j''(0)/2$, respectively. These derivatives are with respect to behavioral dose. In particular, the A-task contrast above satisfies $D_A^{\perp} = -\frac{1}{3}\sum_{j=1}^{3} C_j$; it retains the even response while canceling the odd part. This explains why finite readout ambiguity alone does not identify an uncovered first-order gradient.

What the signed responses reveal. Figure 13 separates preference for one perturbation sign from the change that remains after averaging both signs. In the precision B evaluation, every direction has a negative mean even response at every available dose, and the across-direction loss grows with dose. The full-dose A averages are also negative in both policy stages. Thus functional change remains after the directional preference between signs has been averaged away. This gives a concrete reason to retain finite-update effects when translating historical expression into capability. The odd curves describe preferences within the archived direction convention; signed averages can cancel across seeds and do not measure total sensitivity.

Figure 14 shows why the same local response should not be assumed across contexts. In the original B evaluation, the mean even-response loss grows with dose in both regions. The terminal-region A point estimates do not reproduce this increasing negative pattern. Thus the common orthogonal construction can have different functional consequences across tasks and regions. Together with the early-region results, this motivates measuring the readout-to-function relation in the context where retention is being assessed.

A.4 Pick-and-place credit interventions

This study asks whether positive and negative new-task credit exert opposing effects on old-task success within the same training run. The four sibling updates compare retaining or attenuating

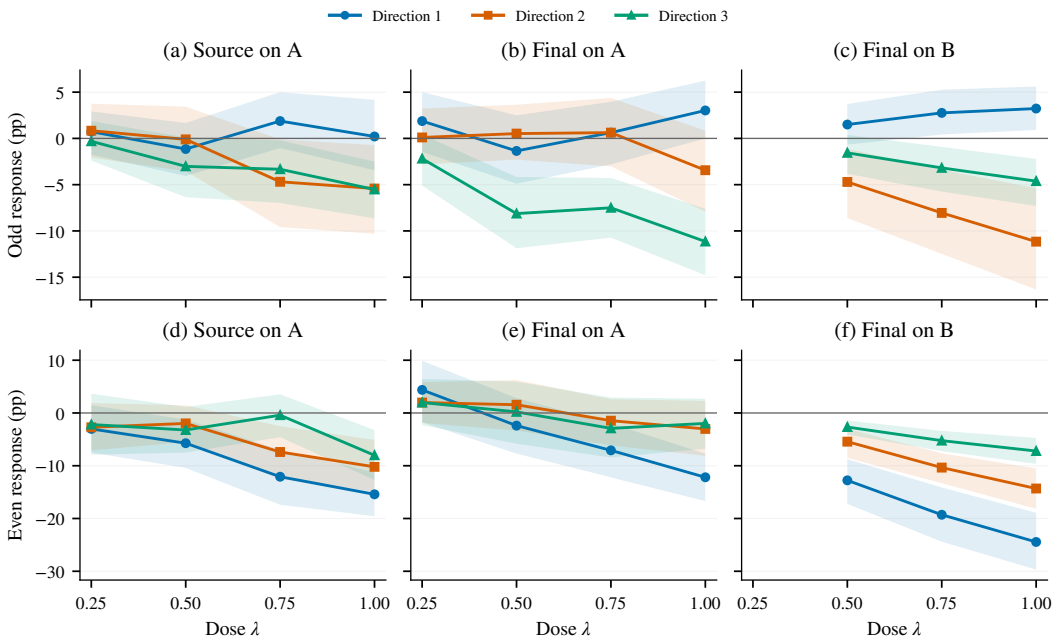

Figure 13: **Direction preference and symmetric functional change.** Odd responses (top) and even responses (bottom) for early-region orthogonal perturbations. Curves show seed means with pointwise 95% paired seed-bootstrap bands; each color retains one archived orthogonal direction and its two signs. A uses 30 episodes per condition and B uses 1000.

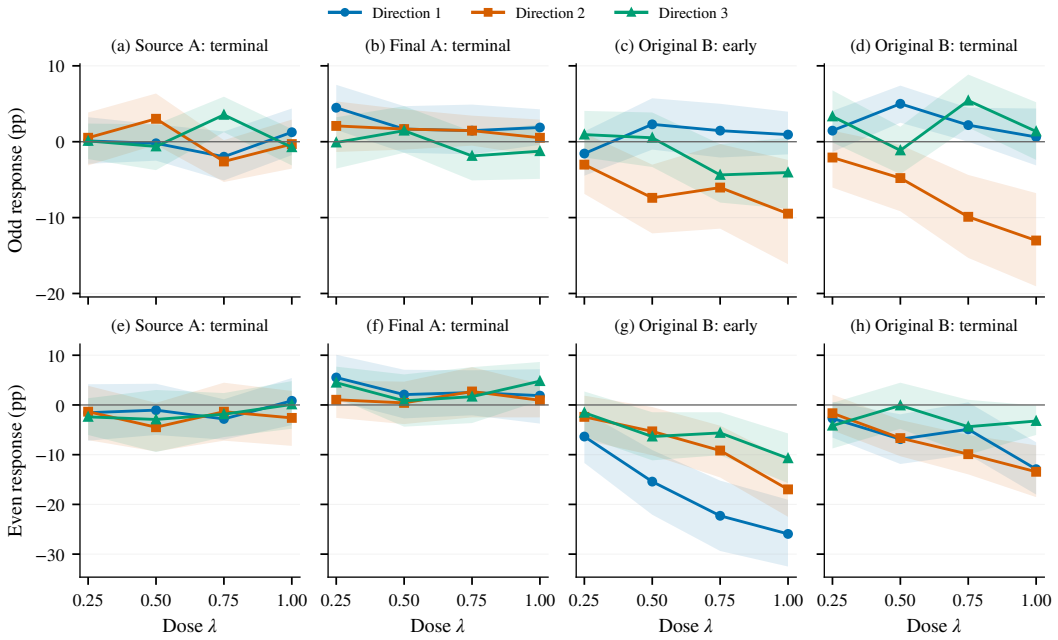

Figure 14: **Signed dose responses in the remaining archived settings.** The same decomposition as Figure 13, applied to both A policies in the terminal region and the original B evaluation in both regions. All use 30 episodes per condition. These panels retain the full regional and directional variation; they are not pooled with the B precision extension.

each sign at a common training state. Evaluating old-task success after each sibling makes support and opposition directly observable; continuing only FULL then places those local effects alongside an actual learning path. The following subsection reports the resulting balance and its relation to forgetting.

The pick-and-place credit interventions use `pick-place-v3` with a shared reward and context ranges split along x. The concatenated object/goal context bounds are

$$A_{\text{low}} = (-0.1, 0.6, 0.02, -0.1, 0.8, 0.05), \qquad A_{\text{high}} = (0, 0.7, 0.02, 0, 0.9, 0.3),$$
$$B_{\text{low}} = (0, 0.6, 0.02, 0, 0.8, 0.05), \qquad B_{\text{high}} = (0.1, 0.7, 0.02, 0.1, 0.9, 0.3).$$

The frozen specification lists 39 observation dimensions, four action dimensions, two hidden layers of width 64, horizon 150, and A/B training budgets of two million environment steps each. PPO uses eight environments, 2048 steps per environment per update, minibatches of 256, ten epochs, learning rate 3×10^{-4}, discount 0.99, GAE parameter 0.95, clipping 0.2, value coefficient 0.5, no entropy bonus, and maximum gradient norm 0.5. Actor and critic gradient clipping are separated.

The functional-accounting study enrolls 17 seeds from 5033–5072 using criteria fixed before intervention outcomes. Eligibility requires source-A success at least 0.85 and a checkpoint in the normalized B-progress band $|P_B - 0.60| \le 0.05$. For a checkpoint with mean B-task episode return R_B, progress is $P_B = (R_B - R_{B,\text{low}})/(R_{B,\text{ref}} - R_{B,\text{low}})$. Here $R_{B,\text{low}}$ is that seed's base-model B return and $R_{B,\text{ref}}$ the maximum final B-only return among completed lineages in the 40-seed candidate pool. The checkpoint closest to 0.60 is selected, breaking ties toward earlier checkpoints; enrollment retains it when it falls in the stated band. The progress band aligns the stage of new-task acquisition across seeds, so functional effects are evaluated at comparable points in learning.

For a sampled state–action pair (s, a) at update t, the control mask is

$$\ell_{\text{ctrl},t}(s,a) = \log \pi_{\theta_t}(a \mid s) - \log \pi_{B,*}(a \mid s), \qquad m_t(s,a) = \min\left\{ \frac{[\ell_{\text{ctrl},t}(s,a)]_+}{\tau_{\text{ctrl}} + \epsilon}, 1 \right\}. \quad (24)$$

Here θ_t is the current learner, $\pi_{B,*}$ the frozen B-only policy at the selected checkpoint, and $\epsilon = 10^{-12}$. The per-seed threshold τ_{ctrl} is the 95th percentile of positive scores in independent calibration rollouts at the selected AB checkpoint. The threshold stays fixed; scores are recomputed before each update and its mask is shared across sibling updates.

A.5 Local functional accounting

The key comparison is between the signed contributions, their net effect, and the observed change in old-task success. A small net effect can arise because neither contribution matters or because substantial opposing effects cancel. We distinguish these cases within each run before averaging across runs.

At each reference state, FULL uses an unweighted actor loss, POS_OFF applies SHUF_POS attenuation, NEG_OFF applies SHUF_NEG attenuation, and BOTH_OFF applies attenuation on both shuffled signs. Attenuation dose is $\lambda = 0.5$. For minibatch size n_{mb}, the actor loss is $n_{\text{mb}}^{-1} \sum_{i=1}^{n_{\text{mb}}} w_i \mathcal{L}_i$, where $\mathcal{L}_i$ is the per-sample PPO loss and w_i the intervention weight. The sign split uses rollout-normalized advantages. Only FULL continues along the reference path; all sibling updates share the saved pre-state and batch and receive independent optimizer-state copies.

For rollout-normalized advantage $\widehat{A}_i$, let $\widetilde{m}_i^+$ and $\widetilde{m}_i^-$ be the two shuffled masks, shared across cells that attenuate the same sign. The four weights are

$$w_i^{pn} = 1 - \lambda \left[(1-p)\widetilde{m}_i^+ \mathbf{1}\{\widehat{A}_i > 0\} + (1-n)\widetilde{m}_i^- \mathbf{1}\{\widehat{A}_i < 0\} \right], \qquad p, n \in \{0, 1\}. \quad (25)$$

Masks are permuted within strata formed by quartile bins of behavior log probability, absolute normalized advantage, and absolute raw advantage. Before sibling updates, the implementation searches up to 300 permutations for a relative removed-credit-mass difference within 2%, where removed mass is $\lambda \sum_i m_i |\widehat{A}_i|$ over samples of the selected sign. It accepts the first match or, if none is found, the lowest-mismatch candidate. Across all 1088 updates, positive and negative masks exceed 2% in 58 and 51 cases, respectively (84 updates for either sign); maxima are 7.18% and 8.11%. All updates remain in the analysis.

Credit-mass matching concerns shuffled versus original masks, not update-size matching to FULL. The recorded actor norm covers only `action_net` and `log_std`, omitting the trained `policy_net` hidden layers. For this output-head/log-standard-deviation subset, median relative update-norm differences from FULL are 19.82%, 24.55%, and 2.14% for POS_OFF, NEG_OFF, and BOTH_OFF. A recorded diagnostic additionally requires both mass tolerances and all three subset-norm differences within 10%; 29/1088 updates meet that joint condition. This flag does not control inclusion or establish whole-actor matching. Critic-update norms coincide across cells. Functional estimates concern the complete finite attenuation operations, including update-magnitude effects, rather than credit placement alone. The supplement provides the diagnostic calculation and source definitions.

Functional evaluation uses the same 128 contexts with eight repetitions throughout each run. Context-reset and action-sampling seeds are shared across cells and updates; each FULL outcome array becomes the next pre-update array. Only FULL continues training. All cell means and local contrasts use this paired evaluation panel, including the initial measurement.

For training seed s, let $C_{jr}^{(s)}$ be any cumulative paired contrast at context j and repetition r, after summing all updates. Its context-cluster precision is

$$\widehat{\mathrm{SE}}_{\mathrm{ctx}}(\widehat{C}^{(s)}) = \frac{\mathrm{sd}_{j=1}^{128}\left(\frac{1}{8}\sum_{r=1}^{8} C_{jr}^{(s)}\right)}{\sqrt{128}}. \tag{26}$$

Here $\widehat{C}^{(s)}$ averages the cumulative contrasts across contexts and repetitions, and sd is the sample standard deviation across context means. This retains correlations across repetitions, cells, and updates. Table 9 reports the resulting precision. Summing within each context before estimating precision preserves the dependence created by reusing evaluation contexts across updates. Across runs, we bootstrap complete paired vectors and trajectories. These intervals condition on the observed enrollment and B-progress reference; they do not resample the candidate pool or reselect checkpoints. The descriptive medians summarize finite-panel run estimates. Context precision is reported separately and is not added again to the run-bootstrap variance.

Table 8: Pathwise accounting over the next 64 B updates in all 17 eligible runs. Means and descriptive medians use complete paired run-level estimates. Intervals are 95% seed-bootstrap intervals from 10000 shared resamples (seed 20260925), conditional on the selected cohort and its progress reference.

Quantity	Mean [95% interval]	Median [95% interval]
Positive operation	0.1394 [0.0661, 0.2086]	0.1523 [0.0146, 0.2695]
Negative operation	-0.1464 [-0.2255, -0.0688]	-0.1401 [-0.2476, -0.0532]
Net immediate effect	-0.0070 [-0.0479, 0.0337]	-0.0234 [-0.0605, 0.0361]
Attenuated-update term	-0.1864 [-0.3262, -0.0596]	-0.1611 [-0.3223, -0.0479]
Observed path change	-0.1933 [-0.3139, -0.0795]	-0.2002 [-0.3682, -0.0391]

Table 9: Precision recomputed from the original success arrays, clustering the cumulative paired contrasts by evaluation context. Entries summarize the within-run standard errors across 17 runs.

Quantity	Median context SE	90th-percentile context SE
Positive operation	0.0389	0.0446
Negative operation	0.0363	0.0477
Net immediate effect	0.0453	0.0533
Attenuated-update term	0.0558	0.0613
Observed path change	0.0302	0.0376

Interpreting the net estimate. The small net estimate summarizes a balance of opposing contributions. Table 8 shows positive and negative mean allocations of comparable magnitude, so combining them hides much of the response visible in either component. For seed s, write $\widehat{L}_{64}^{(s)} = \sum_{t=0}^{63}(f_{11,t}^{(s)} - f_{00,t}^{(s)})$. Here f and Φ denote empirical estimates with hats suppressed. Each

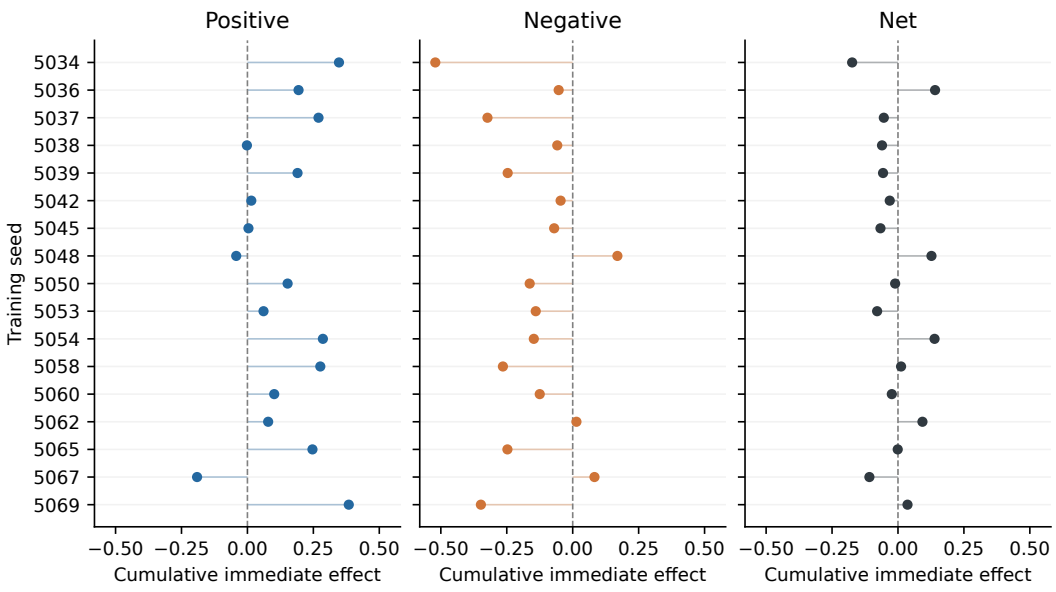

Figure 15: All 17 eligible pick-and-place seeds at update 64, ordered by seed ID. Each point is a cumulative immediate effect; horizontal segments connect it to zero for readability. The net is the within-seed sum of the positive and negative allocations.

seed satisfies $\sum_t \Phi_{+,t} + \sum_t \Phi_{-,t} = \widehat{L}_{64}^{(s)}$, and pairing is preserved before aggregation. The mean and descriptive median summarize this balance across the same complete set of runs. A small net estimate therefore does not imply that new-task credit has little functional influence: support and opposition can be individually appreciable while their sum is small. The interval spanning zero does not establish equivalence.

Table 9 answers a separate measurement question: how much does an estimated cumulative effect vary with the evaluation contexts within one run? The net effect does not automatically have smaller sampling uncertainty simply because its point estimate is small. Context-cluster precision retains the dependence among repeatedly evaluated outcomes, while the paired run intervals summarize variation across trained policies. This is why the signed components, their within-run sums, and the complete seed distribution are reported together.

The remaining update contribution. Let $f_{\mathrm{pre},t}$ denote measured A-success before update t. The final seed-table column records $\Phi_{\mathrm{rest},t} = f_{00,t} - f_{\mathrm{pre},t}$, summed over 64 updates. This attenuated-update term measures both-sign shuffled attenuation relative to the pre-update policy at states visited by FULL. At each update,

$$f_{11,t} - f_{\mathrm{pre},t} = \Phi_{+,t} + \Phi_{-,t} + \Phi_{\mathrm{rest},t}. \tag{27}$$

The fixed evaluation panel carries each FULL outcome forward as the next pre-update outcome. Summing the local identity gives

$$f_{11,N-1} - f_{\mathrm{pre},0} = \sum_{t=0}^{N-1} \left(\Phi_{+,t} + \Phi_{-,t} + \Phi_{\mathrm{rest},t}\right). \tag{28}$$

Most of the observed decrease lies in the attenuated-update term (Table 8). Thus balancing the two signed allocations does not account for all changes produced by training: the remaining update can still reduce old-task success. This term sums sibling changes along FULL states, rather than endpoints of a continued attenuated policy.

Why local support can coexist with forgetting. Figure 16 makes the complete accounting visible. The positive and negative allocations have similar magnitudes and largely offset, leaving a small net immediate contrast. Adding the attenuated-update term recovers the observed decline. Thus

a positive signed contribution is present along a path that still forgets: retention depends on the combined update, including the portion remaining after both attenuations. This distinction connects the local selection question to SFP's analysis of accumulated functional change.

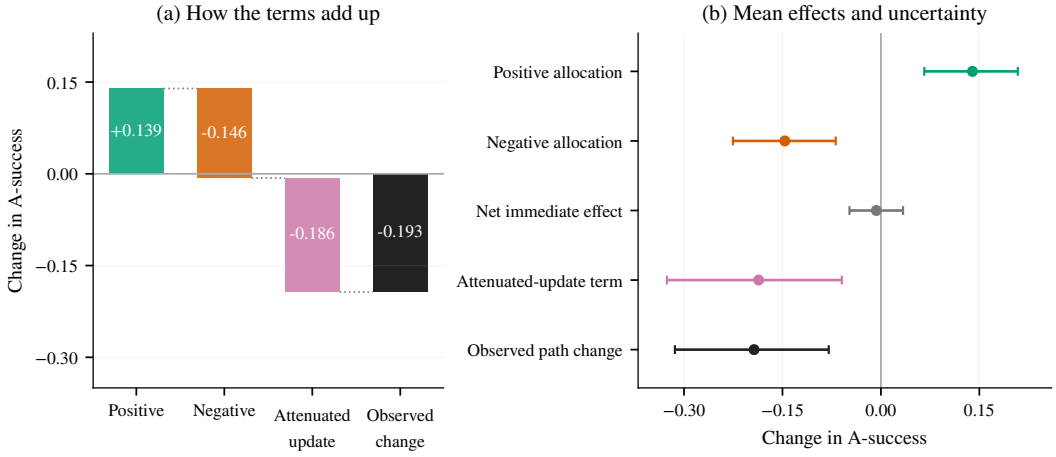

Figure 16: **Local support and total forgetting on the same observed path.** (a) The three cumulative terms in Equation 28 add to the measured change in A-success. Bars represent an additive decomposition of the same path. (b) Mean effects and paired 95% seed-bootstrap intervals from Table 8; all 17 runs contribute to every estimate. The net immediate effect is the sum of the positive and negative allocations.

Table 10: All eligible seeds in the fresh-seed functional accounting experiment. Entries are cumulative immediate effects under the implemented shuffled-credit interventions.

Seed	$\sum \Phi_+$	$\sum \Phi_-$	Net	$\sum_t \Phi_{\mathrm{rest},t}$
5034	+0.3472	-0.5210	-0.1738	+0.2363
5036	+0.1938	-0.0532	+0.1406	-0.3936
5037	+0.2695	-0.3232	-0.0537	-0.3809
5038	-0.0020	-0.0586	-0.0605	-0.3076
5039	+0.1899	-0.2466	-0.0566	-0.3223
5042	+0.0146	-0.0459	-0.0312	+0.0986
5045	+0.0039	-0.0703	-0.0664	-0.0479
5048	-0.0425	+0.1694	+0.1270	+0.0293
5050	+0.1523	-0.1631	-0.0107	+0.2021
5053	+0.0610	-0.1401	-0.0791	-0.1611
5054	+0.2861	-0.1475	+0.1387	-0.6787
5058	+0.2764	-0.2646	+0.0117	-0.0508
5060	+0.1016	-0.1250	-0.0234	-0.1768
5062	+0.0786	+0.0142	+0.0928	-0.8691
5065	+0.2466	-0.2476	-0.0010	-0.0586
5067	-0.1909	+0.0825	-0.1084	-0.1230
5069	+0.3843	-0.3481	+0.0361	-0.1641

All 17 runs are retained. The paired bootstrap uses 10000 shared run-index draws with generator seed 20260925 for means, descriptive medians, and trajectory bands. All components and observed success share these resamples; trajectory bands are pointwise percentile intervals for means. The original directional sign-test values, computed on finite-panel estimates, are 0.0064 for each signed component and 0.9283 for the positive-net hypothesis, without multiplicity adjustment. A sum of marginal medians need not equal the median of within-run sums. Since the pre-update panel carries forward the preceding FULL outcome, summing Equation 27 gives the exact panel endpoint identity in Equation 28. This closure is an accounting property, not a continuation contrast between training rules.

Figure 15 and Table 10 show that the balance also occurs within individual runs: the two allocations have opposite signs in 15/17 seeds. Their relative magnitudes differ, producing both positive and negative net estimates. Cancellation therefore occurs within trained policies, rather than only after separately averaging disjoint groups of runs. This supports studying the balance of functional effects in each training history instead of treating the cohort's near-zero mean as an absence of old-task support.

A.6 How the comparisons inform SFP

The comparisons address successive questions about implicit rehearsal. Historical geometry locates a component that remains expressed; credit interventions test whether new-task updates sustain it; functional probes ask what capability depends on it. Signed accounting then places local support alongside the other changes that determine observed forgetting.

Table 11: Mechanistic questions, experimental comparisons, and the inference supported by each.

Question	Comparison	Inference
What historical component persists during new-task learning?	LLM accuracy and position-wise decomposition on fixed prefixes	A largely maintained acquisition-aligned component coexists with growth in orthogonal differences
Does new-task credit affect historical expression?	Eight reset batches and ten on-policy branch updates	Historical-credit attenuation reduces expression under the implemented update rules
Do both tasks depend on the reference direction?	A- and B-task lesions versus equal-state-KL orthogonal controls	Both task performances depend on the specified early-region direction
Does provenance determine usefulness?	The same lesion construction in two declared regions	Old-task functional dependence differs by region
Does one historical readout determine capability?	Baseline versus projection-preserving orthogonal policies	Identical historical projection can coexist with different task success
What does a finite response contain?	Paired positive/negative perturbations	Functional change remains after averaging the two perturbation signs
How does local support relate to actual forgetting?	Four sibling updates with signed-credit attenuation and the continuing reference path	Positive local support coexists with decline; complete path accounting includes the attenuated-update term

A.7 Additional LLM outcomes and evaluation

The following evaluations place the historical readouts alongside task performance. The main training paths ask whether prior Math performance coexists with learning Code; the branch evaluations ask how changing historical expression relates to each independently measured capability.

Table 12: Final language-model performance (percent). AIME averages verifier scores over 32 responses per problem; code evaluation is greedy. Base and strengthened tests evaluate the same generated code (Liu et al., 2023).

Evaluation at update 500	Math→Code	Base→Code
AIME 2025	24.0	17.2
HumanEval base tests	63.4	65.9
HumanEval+ strengthened tests	61.6	60.4
MBPP base tests	70.9	71.4
MBPP+ strengthened tests	63.0	60.6

Learning Code while retaining Math performance. Table 12 gives the endpoints; Figure 17 shows how they are reached. Along Math→Code, AIME performance stays near its initial level while both code benchmarks finish above their starting scores. Retention therefore accompanies measurable new-task acquisition along this path. Across the two histories, the AIME ordering is maintained throughout the recorded checkpoints, whereas code-benchmark ordering changes during

learning. Prior learning is associated with a lasting old-task advantage, without uniformly ordering all new-task scores. This motivates SFP's capability-specific question: which parts of prior learning remain useful as the policy adapts?

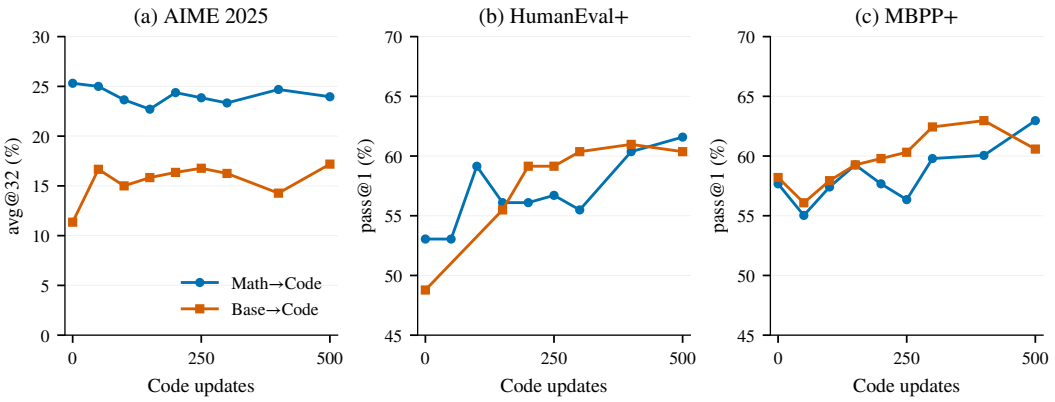

Figure 17: **Old- and new-task performance along the two Code-training paths.** (a) AIME 2025 avg@32; (b,c) greedy pass@1 on strengthened code tests. Markers denote valid recorded evaluations through update 500; lines connect adjacent valid checkpoints. In (b), invalid Base→Code records at updates 50 and 100 are omitted. Table 12 gives the final values, including base code tests.

AIME accuracy first averages verifier scores over 32 responses per problem, then over the 30 problems. The verifier requires a parseable answer in the task format. Parsing failures range from 0.2–2.9% along the Math-initialized path and 52.4–62.5% along the Base-initialized path. Consequently, the complete-response performance in Figure 2a reflects both solving the problem and delivering an answer the verifier can read. Retaining this measured task capability includes retaining the required answer behavior; the trajectory is not a separate measurement of reasoning conditional on successful parsing.

The AIME trajectories in Figure 2a use the exported evaluation summaries, including initialization scores at update 0 and evaluations at updates 50, 100, 150, 200, 250, 300, 400, and 500. Code evaluations use training-style prompts and a common length budget of 16384 tokens. At update 500, HumanEval truncation rates are 30.5% for Math→Code and 22.0% for Base→Code. During training, responses reaching the training length cap are excluded, amounting to approximately one fifth of long responses.

Expression probes complement the capability evaluation. Table 1 in Section 4.1 reports AIME 2025 pass@128 after branch training. Figure 18 shows the original ten-update evaluation, pairing avg@32 Math accuracy and greedy Code scores with fixed-prefix expression probes. These measures distinguish the availability of a successful solution across multiple attempts from average response correctness, and locate the expression response in its probe context.

AIME 2025 pass@128 evaluation. For each cell in Table 1, all 30 AIME 2025 problems receive 128 responses: the original 32 plus 96 additional samples, combined after collection. Both batches use the RL-Zero Math template, Answer-line scoring, temperature 0.6, top-$p = 0.95$, and a 16384-token limit; the vLLM engine and random seeds differ between batches. Since the sample count equals $k = 128$, pass@128 is the fraction of problems with at least one verifier-correct response. TARGET and SHUFFLED continue from step 410 to 430 with their optimizers reset at 410. The later FULL reference uses main-path step 430. All three conditions are evaluated at steps 410 and 430. The endpoint contrasts describe these observed training paths, with problems as the statistical units.

Continuation and evaluation protocol. The three branches share the step-400 starting checkpoint and are evaluated at step 410 after ten branch updates. Each probe set uses its own fixed prefixes and the checkpoint-400 reference construction. The AIME set contains 100 saved responses to 60 distinct problems (one per problem and a second response for 40); MATH-L5 contains one response to

each of 100 problems. Both use AB-400 generations truncated to their first 6000 tokens for measurement, independently of correctness. Capability evaluation generates separate complete responses.

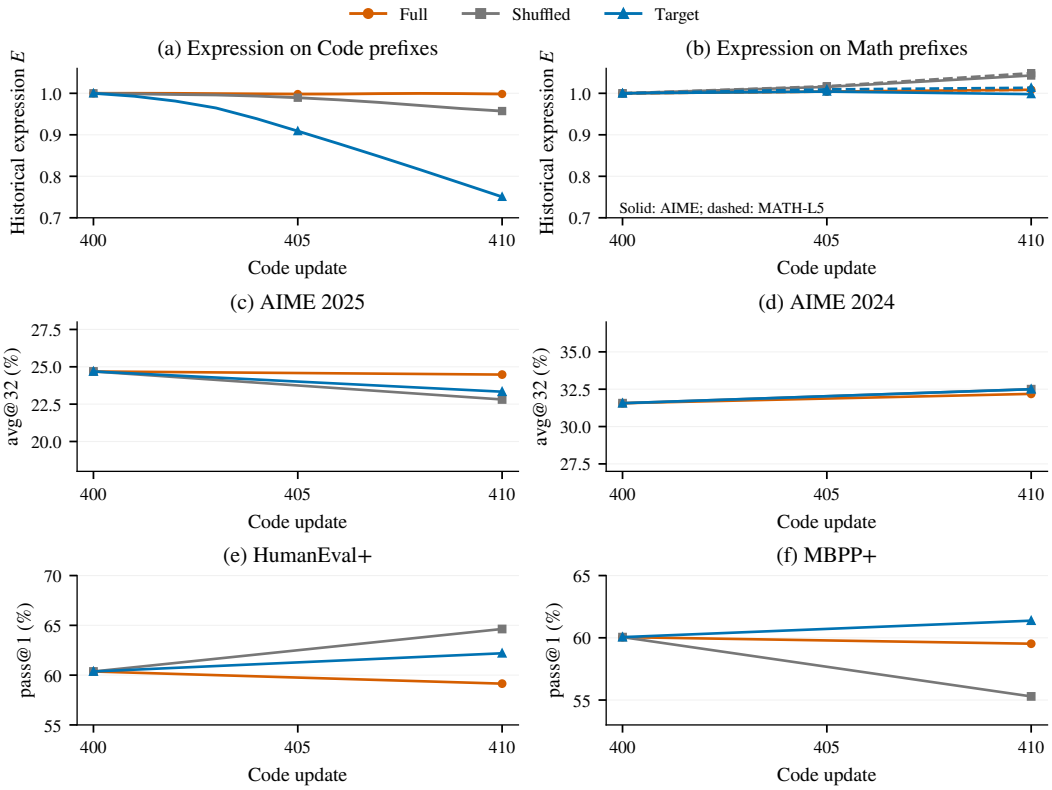

Figure 18: **Historical expression and independently evaluated capability after ten branch updates.** (a) Expression on fixed Code prefixes. (b) Expression on fixed AIME (solid) and MATH-L5 (dashed) prefixes. Panels (a,b) share a vertical scale. (c,d) AIME avg@32 accuracy. (e,f) Greedy pass@1 on HumanEval+ and MBPP+. Colors and markers identify the same update rules in all panels. All branches run from step 400 to 410; capability panels show only the shared start and measured endpoint.

The expression response depends on the probe context. By step 410, TARGET's Code-prefix expression has moved below its starting value, while FULL remains close to the start and SHUFFLED declines less. On the fixed Math prefixes, however, TARGET remains near its starting expression on both probe sets, although below SHUFFLED. Thus the same Code-credit intervention produces different expression responses across the measured contexts. Because the prefixes, historical directions, and reference weights are fixed, these differences reflect policy updates rather than changing probe samples. The ten-update comparison supports the role of Code credit in maintaining historical expression on Code prefixes and shows why that response must be located in its measurement context. Its magnitude alone does not quantify a loss of Math capability; the independent verifier evaluations answer that functional question.

B Detailed Theoretical Derivations

B.1 Notation and measurement dependencies

The framework starts from saved model outputs, training losses, and independently verified task outcomes. Table 13 distinguishes empirical observables from the quantities used in the theoretical identities and bounds.

Table 13: Operational meaning and analytical role of the quantities.

Quantity	Construction or definition	Role in the analysis
$s, y, z_\theta, \pi_\theta$	Saved prompt/prefix, next token, forward logits and softmax probabilities.	Historical forward measurements available.
$\ell, \tau_{\text{hist}}, m_i, q_i$	Log-probability difference; calibration threshold; clipped mask; sign-gated attenuation score.	Defines the historical-credit interventions analyzed in Section 3.1.
c_i, v_i	Differentiate the actual actor objective with respect to token log probability and model parameters.	Credit specifies the intervention; gradients define the selection identity.
W_*, d_*, E	Frozen reference probabilities and logit difference; fixed-prefix projection in Equation 2.	Defines a historical readout for the local analysis.
V_A	Average verifier reward over independently generated A-task completions under the declared decoder.	Independent AIME evaluation at the main checkpoints.
U, h, ψ_i	Parameter displacement; readout gradient; $\psi_i = h^\top M v_i$.	Selection response, with a fixed local gradient-to-update map M.
H, b_A	Differentiate the fixed-probe readouts; estimate the reward-weighted evaluation score gradient.	Theoretical gradients for functional identification.
$F, G, \beta, \varepsilon_A$	Reference-token score outer products, gradient inner products and the functional projection residual.	Theoretical metric, projection coefficients, and coverage error.
S, J_t^0	Complete resumable training state; continue from that state under the specified alternative and evaluate A.	Training state and continuation value in the transport identity.
A_N, B_N, Z_N	Coverage/curvature budget; continuation budget; accumulated functional prediction.	Bound the gap between readout predictions and terminal rule contrasts.
L_N, Δ_N	Sum immediate sibling performance contrasts; compare full training endpoints.	Immediate and terminal estimands in the continuation identity.

Capability gradients and decoding. For a differentiable evaluation policy, the independently defined capability gradient is

$$b_A = \mathbb{E}\left[R_A(x, Y)\nabla_\theta \log p_\theta^{\text{eval}}(Y \mid x)\right]. \tag{29}$$

This estimates a population gradient without differentiating the verifier or a sampled accuracy table. The population definition of V_A uses a fixed prompt distribution and a parameter-independent verifier. For a finite response $Y = (y_1, \ldots, y_T)$ including its stopping convention, the evaluation-policy score is the sum of token log-probability gradients under that decoder. Differentiating the expectation gives Equation 29. Sampling temperature must be included; top-p or top-k truncation requires locally stable support and differentiation of the retained-probability normalization. At support-switch boundaries smoothness can fail. Greedy accuracy tables alone do not provide this gradient estimator. A separate stochastic probe protocol may be used, but its capability target must then be declared separately from greedy accuracy. None of these gradients can be recovered from the published aggregate accuracy numbers alone.

Geometry and feasible estimation. W_* is a vocabulary-space matrix computable from forward probabilities. F is a parameter-space score covariance on a declared reference-prefix distribution. They are not numerically interchangeable. In a chosen finite-dimensional perturbation subspace,

projected score gradients estimate F and the corresponding projected H, b_A. The identification result then applies only to that subspace. Linear solves or Fisher-vector products can avoid constructing a full-model inverse, but their approximation error must be assessed. Adding a ridge changes the metric and therefore the reported residual bound; it must not be silently equated with the undamped Fisher.

The dependency order is outputs and task outcomes $\rightarrow$ readouts and gradients $\rightarrow$ functional projection coefficients $\rightarrow$ predicted local contrasts. In particular, β is not defined by observed retention, and J_t^0 is an expected endpoint of a specified continuation procedure, not a function chosen to reconcile local and terminal effects. The bounds in Appendix B.2 additionally assume control of functional curvature and continuation sensitivity. These analytical requirements are separate from the directly observed output and task measurements.

Interpreting changing path geometry. The fixed projection E in Equation 2 measures movement along a frozen reference. By contrast, M_t and κ_t describe the total path difference against the acquisition direction. To see why these have different roles, consider two orthonormal coordinates e_1, e_2 under a fixed local metric W_{ex}, with $a = e_1$ and $d_t = e_1 + \gamma_t e_2$ for $\gamma_t \geq 0$. Using this metric for magnitude and alignment gives

$$M_t = \sqrt{1+\gamma_t^2}, \qquad \kappa_t = \frac{1}{\sqrt{1+\gamma_t^2}}, \qquad \langle a, d_t \rangle_{W_{\text{ex}}} = 1. \tag{30}$$

Increasing the second component raises magnitude and lowers alignment while preserving the first component exactly. If old-task performance depends only on the first coordinate, it remains unchanged in this example. The position-wise decomposition in Equation 18 directly separates these components in the LLM measurements; the control lesions independently test the task relevance of a specified reference direction. Moreover, because trajectory statistics average within prompts before aggregation, multiplying reported mean magnitudes and alignments does not recover a mean acquisition-direction projection.

B.2 Full statements and conditional bounds

The main text defines the operational observables and their estimation routes. Here F is the parameter-space Fisher, not the output-space weight W_*. All local gradients are evaluated at the shared pre-update parameters in a common subspace with $F \succ 0$. Unsubscripted norms are Euclidean, and weighted norms satisfy $\|x\|_\Omega^2 = x^\top \Omega x$ for a positive (semi)definite matrix Ω. A transition kernel acts by $(P_t^r f)(S) = \mathbb{E}[f(S') \mid S, r]$, with rule r and update index t. The terminal contrast Δ_N is defined in Equation 9. We assume the smoothness and integrability specified below.

B.2.1 Credit selection depends on gradient alignment and normalization

Fix a pre-update state and batch. Let v_i be sample i's gradient contribution to the actual actor objective being maximized, including advantage normalization, importance ratios, and clipping. For a frozen, stop-gradient attenuation score $q_i \in [0,1]$, write $w_i = 1 - \lambda q_i$, $0 \leq \lambda < 1$. Use the batch average $\langle \cdot \rangle_n$ from Section 3.1 and set $\bar{q} = \langle q_i \rangle_n$. Write $\mathrm{Cov}_n(q, v) = \langle q_i v_i \rangle_n - \bar{q} \langle v_i \rangle_n$ for empirical covariance. Two common loss conventions give different gradients:

$$g_\lambda^{\text{fixed}} = \langle (1 - \lambda q_i) v_i \rangle_n, \qquad g_\lambda^{\text{renorm}} = \frac{\langle (1-\lambda q_i) v_i \rangle_n}{1 - \lambda \bar{q}}, \qquad g_0 = \langle v_i \rangle_n. \tag{31}$$

Theorem 1 (Credit selection under two loss conventions)**.** *For the fixed state and batch,*

$$g_0 - g_\lambda^{\text{fixed}} = \lambda \langle q_i v_i \rangle_n, \qquad g_0 - g_\lambda^{\text{renorm}} = \frac{\lambda}{1 - \lambda \bar{q}} \mathrm{Cov}_n(q, v). \tag{32}$$

Suppose updates take the form $U = u_0 + \eta M g$ with common u_0 and fixed linear map M. For a scalar readout E with pre-update gradient h, set $\psi_i = h^\top M v_i$. Let q^σ be a uniform permutation within strata C of sizes n_C. Write $U_{\lambda,q}$ and $U_{\lambda,\sigma}$ for updates using the original and permuted scores. Then

$$\mathbb{E}_\sigma h^\top (U_{\lambda,\sigma} - U_{\lambda,q}) = \eta b_\lambda \sum_C \frac{n_C}{n} \mathrm{Cov}_{i \in C}(q_i, \psi_i), \quad b_\lambda = \begin{cases} \lambda, & \textit{fixed denominator}, \\ \lambda/(1-\lambda\bar{q}), & \textit{renormalized}. \end{cases} \tag{33}$$

Equation 33 distinguishes the amount of attenuated credit from its alignment with the measured historical direction. For a fixed batch, the predicted effect is linear in dose with a fixed denominator and rational in dose after renormalization. These are quantitative local predictions, not predictions that a 64-update trajectory follows the same curve. If an implementation permutes a mask and then multiplies by a varying sign or credit factor, it may not preserve $\langle q_i \rangle_n$; each arm's actual denominator must then be retained.

Positive credit alone does not determine the sign of ψ_i: shared parameters and the update map can turn a positive sample contribution away from h. Positive- and negative-credit strata must therefore be tested separately. Nor does merely observing $\ell > 0$ establish reward selection: for actions sampled from π_{AB} at a fixed state, $\mathbb{E}[\log \pi_{AB} - \log \pi_B] = D_{\mathrm{KL}}(\pi_{AB} \| \pi_B) \geq 0$, independently of reward.

The actual optimizer. The theorem's gradient identities are exact, whereas its readout interpretation is local. For a differentiable one-gradient optimizer map $\mathcal{U}_S$, the zero-dose response is

$$\frac{d}{d\lambda} E\big(\theta + \mathcal{U}_S(g_\lambda)\big)\Big|_0 = -h_{\mathrm{post}}^\top D_g \mathcal{U}_S(g_0)\, b, \tag{34}$$

where h_{post} is the readout gradient after the baseline update, D_g is the derivative with respect to its gradient input, and $b = \langle q_i v_i \rangle_n$ or $\mathrm{Cov}_n(q, v)$ under the respective conventions. Adam's response need not equal a shared positive-definite preconditioner. Multiple epochs, minibatches, and clipping require the complete update map or paired finite differences. This distinction is especially relevant to PPO (Schulman et al., 2017b). The controlled comparisons below test realized responses, not exact natural-gradient dynamics.

B.2.2 Historical response identifies function only under coverage

To predict capability from expression changes, approximate the capability gradient by a weighted combination of readout gradients. Let $H = [\nabla E_1, \ldots, \nabla E_K]$, $b_A = \nabla V_A$, and define

$$G = H^\top F^{-1} H, \quad \beta = G^\dagger H^\top F^{-1} b_A, \quad e = b_A - H\beta, \quad \varepsilon_A^2 = e^\top F^{-1} e. \tag{35}$$

The Gram matrix G records overlap between readout gradients in the inverse-Fisher metric. The coefficients β give the least-squares approximation, and e is the unmatched gradient. Here $G^\dagger$ is the Moore–Penrose inverse; linear dependence between readouts is allowed.

Theorem 2 (Functional identification and its residual)**.** *For all local updates u, the historical response $w = H^\top u$ uniquely determines the first-order functional response $b_A^\top u$ if and only if $b_A \in \mathrm{span}(H)$. In general, let $u_\parallel = F^{-1} H G^\dagger w$ and $u_\perp = u - u_\parallel$. Then*

$$b_A^\top u = \beta^\top w + e^\top u_\perp, \tag{36}$$

$$\big|b_A^\top u - \beta^\top w\big| \leq \varepsilon_A \sqrt{u^\top F u - w^\top G^\dagger w}. \tag{37}$$

For feasible w and budget $u^\top F u \leq Q$, the exact attainable interval is $\beta^\top w \pm \varepsilon_A \sqrt{Q - w^\top G^\dagger w}$ when $e \neq 0$; it collapses to a point when $e = 0$.

This is an identification statement over the declared tangent space, not an assertion that PPO can realize every direction in that interval. For a restricted intervention subspace, coverage need hold only on that subspace. A single projection can change while capability remains stable because its functional coefficient is small, an unobserved direction compensates, or higher-order effects dominate. Conversely, a lesion that damages performance does not establish that increasing the same projection is locally beneficial.

For two updates U_1, U_0 from the same parameters, with $u = U_1 - U_0$, Taylor expansion gives

$$\big|\Delta V_A - \beta^\top \Delta \mathbf{E}\big| \leq \varepsilon_A \|u_\perp\|_F + \frac{L_A + \sum_j |\beta_j| L_j}{2} \left(\|U_1\|^2 + \|U_0\|^2\right), \tag{38}$$

where $\Delta V_A = V_A(\theta + U_1) - V_A(\theta + U_0)$ and $\Delta \mathbf{E} = \mathbf{E}(\theta + U_1) - \mathbf{E}(\theta + U_0)$, and L_A, L_j bound the corresponding Hessians and β is frozen before updating.

Direction versus update size. At a local maximum of V_A, first-order sensitivity vanishes. Writing $K_A = -\nabla^2 V_A \succeq 0$, forgetting is locally $\frac{1}{2}u^\top K_A u$, whereas reference-policy KL is $\frac{1}{2}u^\top F u$. Equal KL can therefore produce different forgetting. A new-task Fisher F_B, computed on a fixed B-task reference distribution, controls all second-order old-task damage by a finite constant only if $\ker F_B \subseteq \ker K_A$ (Appendix B.6). Thus, controlling update magnitude is useful but cannot replace identifying its functional direction.

Proposition 1 (Finite ambiguity of a historical readout)**.** *Let $J \geq 1$ and let policies $\pi^{(0)}, \ldots, \pi^{(J)}$ share a fixed readout value $\mathbf{E}(\pi^{(j)}) = e$. Write $v_j = V_A(\pi^{(j)})$, $v_{\min} = \min_j v_j$, and $v_{\max} = \max_j v_j$. Then*

$$\inf_{\widehat{v}\in\mathbb{R}} \max_{0\leq j\leq J} |v_j - \widehat{v}| = \frac{v_{\max} - v_{\min}}{2} \geq \frac{1}{2}\left| v_0 - \frac{1}{J}\sum_{j=1}^{J} v_j \right|. \tag{39}$$

Proof. All readout-only predictors return one number at e. The maximum error is at least half the distance between the largest and smallest values, attained by their midpoint. The control average lies between the extreme values, giving the inequality. □

This finite-set statement requires no differentiability. Across training seeds, let $D_s = v_0^{(s)} - J^{-1}\sum_{j=1}^{J} v_j^{(s)}$ and let R_s be the maximum error of a predictor using only the fixed readout within seed s. Then $\mathbb{E}_s R_s \geq \frac{1}{2}\mathbb{E}_s|D_s| \geq \frac{1}{2}|\mathbb{E}_s D_s|$, even if each seed uses its own predictor. Estimates require uncertainty intervals for the success contrast. The result tests finite readout sufficiency, not first-order coverage alone: symmetric positive/negative perturbations cancel their first-order average and can reveal curvature instead. A violation of exact finite prediction therefore does not by itself establish $b_A \notin \operatorname{span}(H)$.

For approximate equality, an L_g-Lipschitz predictor g satisfies the robust version

$$\max_j |v_j - g(\mathbf{E}(\pi^{(j)}))| \geq \frac{1}{2}\left[\left| v_0 - \frac{1}{J}\sum_{j=1}^{J} v_j \right| - \frac{L_g}{J}\sum_{j=1}^{J} \|\mathbf{E}(\pi^{(j)}) - \mathbf{E}(\pi^{(0)})\|_2\right]_+. \tag{40}$$

This follows by adding and subtracting the predictions in the success contrast. Exact readout equality is required for the unrestricted-predictor statement; numerical closeness alone does not replace it.

B.2.3 Local effects require transport through subsequent learning

Let $v(S) = V_A(\theta(S))$ and compare training rules P_t^1, P_t^0 from the same initial state distribution. Define the remaining-training value under rule 0 by $J_N^0 = v$ and $J_t^0 = P_t^0 J_{t+1}^0$. Unlike v, this continuation value can depend on optimizer and critic states even when the current actor is unchanged.

Theorem 3 (Transport of local interventions)**.** *For integrable finite-horizon transitions,*

$$\Delta_N = \sum_{t=0}^{N-1} \mathbb{E}_1\left[((P_t^1 - P_t^0)J_{t+1}^0)(S_t)\right], \tag{41}$$

$$\Delta_N - L_N = \sum_{t=0}^{N-1} \mathbb{E}_1\left[((P_t^1 - P_t^0)(J_{t+1}^0 - v))(S_t)\right], \tag{42}$$

where $\mathbb{E}_1$ averages states visited under rule 1, and $L_N = \sum_t \mathbb{E}_1[((P_t^1 - P_t^0)v)(S_t)]$ is the cumulative immediate effect along the reference path. If $J_{t+1}^0 - v$ is L_t^{cont}-Lipschitz in a specified complete-state metric d, a coupling of sibling next states gives

$$|\Delta_N - L_N| \leq B_N := \sum_t L_t^{\mathrm{cont}} \mathbb{E}_1 \mathsf{d}(S'_{1,t}, S'_{0,t}). \tag{43}$$

This applies the performance-difference telescoping principle to learning states, rather than environment states (Schulman et al., 2017a). It exposes what immediate virtual updates omit: the effect of their changed state on the rest of training. A positive L_N can coexist with negative Δ_N. A sampler comparison fits the same theorem by changing the transition kernels; the identity supplies no inherent on-policy advantage.

B.2.4 A CONDITIONAL CRITERION FOR STUDYING FORGETTING

Combining the preceding results yields a criterion that distinguishes protection, damage, and insufficient information. For same-state interventions evaluated along the reference path, define $Z_N = \sum_t \mathbb{E}_1[\beta_t^\top \Delta \mathbf{E}_t]$. Let A_N be the sum of expected right-hand sides of Equation 38. Then

Corollary 1 (Relative functional effect)**.** *Under the conditions above,*

$$\Delta_N \in [Z_N - A_N - B_N,\ Z_N + A_N + B_N]. \tag{44}$$

Consequently $Z_N > A_N + B_N$ certifies a positive relative effect, and $Z_N < -(A_N + B_N)$ certifies a negative one. An interval containing zero gives no sign certificate.

The criterion is analytical: β maps historical response to its covered functional component, A_N bounds coverage and finite-update errors, and B_N bounds the difference between immediate and terminal effects. Its sign conclusion follows when the resulting interval excludes zero.

The same statements apply to any earlier task in a sequence. Protecting several tasks requires their respective functional conditions; improving one historical coordinate need not protect them all. A general theory of continual learning must therefore explain both which histories are selected and which old-task functions that selection covers. The empirical section tests selected links separately.

B.3 SCOPE, METRICS, AND ACCESSIBLE HISTORICAL READOUTS

The local results assume differentiable population performance and readouts in a finite-dimensional regular coordinate chart. A finite-sample success rate is an estimator of this population quantity, not itself the differentiable object. Twice differentiable functions with bounded Hessians are required for the finite-update bounds. For singular Fisher matrices, the corresponding statements hold on an identifiable constant-rank subspace; we do not identify an arbitrary Euclidean pseudoinverse with an invariant tensor under every reparameterization.

A reference model can combine measurement domains without identifying their distributions. With an observed label j, let $p_\theta^{\mathrm{ref}}(j, x) = \omega_j p_{\theta,j}(x)$ for fixed positive mixture weights satisfying $\sum_j \omega_j = 1$. Then $F = \sum_j \omega_j F_j$. A component-specific expectation is represented by $f^{\mathrm{ref}}(j, x) = \mathbf{1}\{j = j_0\} f(x)/\omega_{j_0}$. This construction makes the reference geometry explicit; it does not imply that an optimizer uses that geometry.

Lemma 1 (Score projection)**.** *For a regular model p_θ, score $\xi = \nabla \log p_\theta$, Fisher $F = \mathbb{E}[\xi\xi^\top]$, and parameter-independent square-integrable f, let $h = \nabla \mathbb{E} f$. Here $F^\dagger$ is the Moore–Penrose inverse, Π projects centered functions onto the score span, and $\|g\|_{L^2}^2 = \mathbb{E}[g^2]$. Then*

$$h = \mathbb{E}[(f - \mathbb{E}f)\xi] \in \mathrm{Range}(F), \quad \Pi f = \xi^\top F^\dagger h, \quad h^\top F^\dagger h = \|\Pi f\|_{L^2}^2 \le \mathrm{Var}(f). \tag{45}$$

Proof. Differentiating the expectation and using $\mathbb{E}\xi = 0$ gives the first identity. If $x \in \ker F$, then $\mathbb{E}(x^\top \xi)^2 = 0$, so $x^\top h = 0$. Hence h lies in the range of F. The displayed Πf solves the normal equations for projection of $f - \mathbb{E}f$ onto the score span. Its squared norm is $h^\top F^\dagger F F^\dagger h = h^\top F^\dagger h$. Orthogonal projection cannot increase squared norm. □

For a fixed reward r, ideal natural-gradient flow gives $d(\mathbb{E}f)/dt = h^\top F^\dagger \nabla \mathbb{E} r = \langle \Pi f, \Pi r \rangle$. This is a geometric reference, following existing natural-gradient analysis (Kakade, 2001); it is not an exact description of the experimental optimizers.

The control mean-projection readout admits a fixed-f representation: given a frozen historical direction d_*, action covariance Σ_*, reference mean $\mu_{\mathrm{ref},*}$, and positive stabilizer ϵ, take at a probe state

$$f_*(s, a) = \frac{d_*^\top \Sigma_*^{-1}(a - \mu_{\mathrm{ref},*})}{d_*^\top \Sigma_*^{-1} d_* + \epsilon}. \tag{46}$$

For pushing, the reference mean is μ_{ctrl}. Expectation over the pre-clipping action distribution replaces a by $\mu_\theta(s)$. The language-model logit projection is a different differentiable readout and is covered directly by Theorems 1–2; no fixed-f representation is assumed for it. If the reference direction or metric changes with time, its derivative contributes an additional explicit readout-drift term. Freezing the readout prevents this term from being mistaken for a training effect.

B.4 Proof of the credit-selection identities

Proof of Theorem 1. With $\bar{v} = \langle v_i \rangle_n$ and $b_q = \langle q_i v_i \rangle_n$, fixed-denominator attenuation gives $g_\lambda = \bar{v} - \lambda b_q$. For renormalization,

$$\bar{v} - \frac{\bar{v} - \lambda b_q}{1 - \lambda \bar{q}} = \frac{\lambda(b_q - \bar{q}\bar{v})}{1 - \lambda \bar{q}} = \frac{\lambda \operatorname{Cov}_n(q, v)}{1 - \lambda \bar{q}}. \tag{47}$$

Left-multiplication by $\eta h^\top M$ gives the respective local readout response. Write $\langle \cdot \rangle_C$ for a uniform average within stratum C. A uniform permutation satisfies

$$\mathbb{E}_\sigma \langle q_i^\sigma \psi_i \rangle_C = \langle q_i \rangle_C \langle \psi_i \rangle_C. \tag{48}$$

Subtracting this from $\langle q_i \psi_i \rangle_C$ and averaging strata with their empirical probabilities yields $\sum_C (n_C/n) \operatorname{Cov}_{i \in C}(q_i, \psi_i)$. Both $\bar{q}$ and the batch-average gradient are unchanged by permuting q. Therefore the same subtraction applies to the centered expression under renormalization, with the stated coefficient. □

Nonideal permutations and finite updates. If the shuffled score has average $\bar{q}_\sigma = \langle q_i^\sigma \rangle_n \neq \bar{q}$, the exact difference of renormalized gradients is

$$g_{\lambda,\sigma} - g_{\lambda,q} = \frac{\lambda\{b_q - b_\sigma + (\bar{q}_\sigma - \bar{q})\bar{v} + \lambda(\bar{q} b_\sigma - \bar{q}_\sigma b_q)\}}{(1 - \lambda \bar{q}_\sigma)(1 - \lambda \bar{q})}, \tag{49}$$

where $b_\sigma = \langle q_i^\sigma v_i \rangle_n$. Thus approximate mass matching is not an exact permutation theorem. Rejection sampling to obtain a mass-matched permutation can also change the permutation distribution: Equation 33 refers specifically to the uniform within-stratum distribution, not every accepted shuffle procedure.

If the empirical loss uses $\sum_i w_i + \epsilon_{\text{num}}$, put $\delta = \epsilon_{\text{num}}/n$. For the consistently normalized baseline $g_0 = \bar{v}/(1 + \delta)$,

$$g_0 - g_\lambda = \frac{\lambda\{(1 + \delta) b_q - \bar{q}\bar{v}\}}{(1 + \delta)(1 + \delta - \lambda \bar{q})}. \tag{50}$$

The simple covariance identity is recovered when $\delta = 0$.

If L_E bounds the readout Hessian in operator norm along both update segments, Taylor's theorem gives

$$|E(\theta + U_1) - E(\theta + U_0) - h^\top (U_1 - U_0)| \leq \tfrac{1}{2} L_E (\|U_1\|^2 + \|U_0\|^2). \tag{51}$$

Equation 34 follows by the chain rule: $g_0' = -\langle q_i v_i \rangle_n$ or $-\operatorname{Cov}_n(q, v)$, followed by the optimizer Jacobian and the readout gradient evaluated after the baseline update. For a multi-minibatch training step, q can affect every intermediate state and gradient, so differentiation must follow that whole computation. Clipping at a boundary can require a one-sided derivative. For PPO, let r_i be the current-to-behavior probability ratio and ϵ_{clip} the clipping radius. The active gradient region is sign-dependent; $|r_i - 1| \leq \epsilon_{\text{clip}}$ alone is not the full clipping rule.

B.5 Functional identification and tightness

Proof of Theorem 2. Let $\widetilde{H} = F^{-1/2} H$ and $\widetilde{b} = F^{-1/2} b_A$. The chosen β solves Euclidean least squares for $\widetilde{H}\beta \approx \widetilde{b}$, so $H^\top F^{-1} e = 0$. Also $\operatorname{Range}(G) = \operatorname{Range}(H^\top)$. For $w = H^\top u$, this implies $GG^\dagger w = w$, hence $H^\top u_\perp = 0$ and

$$u_\parallel^\top F u_\perp = 0, \quad \|u_\parallel\|_F^2 = w^\top G^\dagger w, \quad \|u_\perp\|_F^2 = u^\top F u - w^\top G^\dagger w. \tag{52}$$

Since $e^\top u_\parallel = 0$, substitution of $b_A = H\beta + e$ gives Equation 36. Cauchy–Schwarz between $F^{-1/2} e$ and $F^{1/2} u_\perp$ proves Equation 37.

For feasible w and $Q \geq w^\top G^\dagger w$, let $r = \sqrt{Q - w^\top G^\dagger w}$. If $e \neq 0$, the choices $u = u_\parallel \pm r F^{-1} e / \varepsilon_A$ attain the two interval endpoints and respect the readout constraint. Intermediate multiples attain all points between them. If $e = 0$, every feasible update has response $\beta^\top w$.

If $b_A \in \operatorname{span}(H)$, first-order function is a linear function of w. Otherwise $e \neq 0$, and updates $u = \pm F^{-1} e$ both have $w = 0$ but opposite nonzero functional responses. Thus w cannot uniquely identify function over all local updates. □

Finite-step bound. Expand each $E_j(\theta + U_r)$ and $V_A(\theta + U_r)$ around the shared pre-state, subtract the expansions for $r = 1, 0$, and apply the preceding bound to $u = U_1 - U_0$. Each scalar Taylor remainder is bounded by its Hessian bound times $\|U_r\|^2/2$. Summing their absolute values proves Equation 38. The result requires no assumption that observed $\Delta\mathbf{E}$ equals its first-order approximation exactly.

Equal historical response can hide opposite capability changes. Take $F = I$, $E(\theta) = \theta_1$, and $V_A(\theta) = \theta_1 + \theta_2$ locally. Updates (a, b) and $(a, -b)$ have the same historical response and Fisher norm, but their functional changes are $a + b$ and $a - b$. They have opposite signs if $b > |a|$. Function may be made bounded outside the local neighborhood without changing this example.

A damaging lesion does not imply a positive functional slope. For $E(\theta) = \theta$ and $V_A(\theta) = 1 - (\theta - 1)^2$, a lesion from 1 to $1 - \delta$ damages performance by δ^2. Nevertheless, the gradient at 1 is zero, and increasing the readout to $1 + \delta$ causes the same damage. Lesion dependence therefore does not establish $\beta > 0$ or monotonicity along a learning trajectory.

B.6 Functional curvature and KL coverage

Proposition 2 (Local damage and reference geometry)**.** *At a twice differentiable local maximum θ_* of V_A, write $K_A = -\nabla^2 V_A(\theta_*) \succeq 0$. For a regular reference model with $F \succ 0$,*

$$V_A(\theta_*) - V_A(\theta_* + u) = \tfrac{1}{2} u^\top K_A u + o(\|u\|^2), \tag{53}$$

$$D_{\mathrm{KL}}(p^{\mathrm{ref}}_{\theta_*} \| p^{\mathrm{ref}}_{\theta_* + u}) = \tfrac{1}{2} u^\top F u + o(\|u\|^2). \tag{54}$$

Their quadratic terms are bounded relative to one another by the extreme eigenvalues of $F^{-1/2} K_A F^{-1/2}$. For $F_B \succeq 0$ in finite dimension, a finite Λ satisfying $u^\top K_A u \le \Lambda u^\top F_B u$ for all u exists if and only if $\ker F_B \subseteq \ker K_A$.

Proof. The expansions follow from stationarity and the local Fisher expansion of KL. Substitution of $z = F^{1/2} u$ and the Rayleigh quotient gives the eigenvalue bounds. If a vector lies in $\ker F_B$ but not $\ker K_A$, positive semidefiniteness gives positive damage at zero F_B energy, contradicting a finite bound. Conversely, the kernel inclusion removes all cross terms involving $\ker F_B$; on its orthogonal complement F_B is positive definite and the largest generalized eigenvalue is finite. □

For $F = I$ and $K_A = \mathrm{diag}(1, 100)$, equally sized coordinate updates have equal local KL but a hundredfold difference in quadratic damage. This illustrates a possible explanation, not a fitted account of the experimental heterogeneity.

For a linear historical response on a single fixed geometry, $|h^\top u|/\|u\|_F = \|h\|_{F^{-1}} |\cos_F(u, F^{-1}h)|$, where $\cos_F(u, v) = u^\top F v / (\|u\|_F \|v\|_F)$ for nonzero vectors. This motivates reporting direction per unit change, but does not turn ratios computed on different probe distributions into matched-intervention effects. Near-zero denominators and finite-update curvature further limit that interpretation.

B.7 Four-cell attribution and treatment semantics

Let $f_{pn}(S)$ be the expected next-step old-task performance at a fixed complete state. The binary indicators specify whether the declared positive- and negative-side reference operations are retained. Define interaction $I = f_{11} - f_{10} - f_{01} + f_{00}$. The symmetric allocation is

$$\Phi_+ = \tfrac{1}{2}[(f_{10} - f_{00}) + (f_{11} - f_{01})], \qquad \Phi_- = \tfrac{1}{2}[(f_{01} - f_{00}) + (f_{11} - f_{10})]. \tag{55}$$

It satisfies

$$\Phi_+ = (f_{10} - f_{00}) + I/2, \quad \Phi_- = (f_{01} - f_{00}) + I/2, \quad \Phi_+ + \Phi_- = f_{11} - f_{00}. \tag{56}$$

Expansion proves these identities. They are the two-factor Shapley convention (Shapley, 1952). The sum $(f_{11} - f_{01}) + (f_{11} - f_{10})$ differs from $f_{11} - f_{00}$ by I and generally cannot replace it.

The estimand is determined by the two factor operations. In the functional-accounting experiment, the factors retain positive or negative credit relative to shuffled attenuation, so the allocations measure those operations. TARGET versus SHUFFLED defines a different contrast: the placement of

attenuation at matched removed mass. Each comparison uses its own declared intervention and reference distribution.

B.8 Proof of temporal transport and the combined criterion

Proof of Theorem 3. Using $J_t^0 = P_t^0 J_{t+1}^0$, conditional expectation on the reference path gives

$$\mathbb{E}_1[J_{t+1}^0(S_{t+1}) - J_t^0(S_t)] = \mathbb{E}_1[((P_t^1 - P_t^0)J_{t+1}^0)(S_t)]. \tag{57}$$

The sum telescopes to $\mathbb{E}_1 v(S_N) - \mathbb{E}J_0^0(S_0)$. Because both rules share the initial distribution, the second term is $\mathbb{E}_0 v(S_N)$. This proves Equation 41. Subtracting the definition of L_N gives Equation 42.

For a coupling (S_1', S_0') conditional on S_t, and $f = J_{t+1}^0 - v$, the difference of kernel expectations is $\mathbb{E}[f(S_1') - f(S_0') \mid S_t]$. Its absolute value is at most $L_t^{\mathrm{cont}}\mathbb{E}[\mathsf{d}(S_1', S_0') \mid S_t]$. Taking expectations and summing proves Equation 43. Taking an infimum over admissible couplings gives a Wasserstein-1 version when the relevant first moments exist. □

Proof of Corollary 1. At each reference pre-state, apply Equation 38 to the coupled sibling updates, with the same β_t for both arms. Expectation and summation give $|L_N - Z_N| \le A_N$. The triangle inequality with $|\Delta_N - L_N| \le B_N$ yields $|\Delta_N - Z_N| \le A_N + B_N$. The sign conclusions follow immediately. These are population bounds; estimates of their terms need additional statistical uncertainty control. □

Immediate benefit can reverse after continuation. Let $S = x \in \mathbb{R}$, $x_0 = 0$, and $v(x) = x$. At the first step, rule 1 adds one and rule 0 does nothing. At the second step both apply $x \mapsto -x$. The first immediate contrast is $+1$, the second is zero, so $L_2 = 1$, but $\Delta_2 = -1$. On the finite reachable set, $v(x) = (x+1)/2$ gives a bounded performance example with the same sign reversal.

Relative protection is weaker than no forgetting. The absolute reference-path change equals

$$\mathbb{E}_1 v(S_N) - \mathbb{E}v(S_0) = \underbrace{\mathbb{E}_0 v(S_N) - \mathbb{E}v(S_0)}_{\text{baseline change}} + \Delta_N. \tag{58}$$

A positive relative effect can therefore coexist with substantial absolute forgetting. Fixed-update endpoints, stopping times, and matched-task-progress endpoints also define different estimands; the last two require their own sampling and support assumptions.

B.9 Sampler effects and multiple old tasks

An on-policy two-action bandit already rules out unconditional protection. Let the old task value be $V_A(p) = p$, the new task value $V_B(p) = 1 - p$, and $p \in (0, 1)$. The Fisher is $1/[p(1-p)]$, so exact natural-gradient ascent on B gives $\dot{p} = -p(1-p) < 0$. Compatible reward reverses this sign. Sampling on-policy does not create task compatibility.

Exact importance weighting equates expectations of a fixed integrable gradient when the full density ratio and support conditions hold. Action-only ratios do not generally correct state occupancy. Finite batches, clipping, advantage estimation, and nonlinear optimizers further separate equality of expected gradients from equality of expected updates.

For multiple old tasks, form one b_{A_j} and one identification bound per task. With a fixed readout collection, simultaneous first-order identification requires every b_{A_j} to lie in its span (or the corresponding restricted-space condition). Since task gradients can conflict, no scalar historical-retention score generally orders all old-task performances. This extends the framework's questions to continual learning without claiming that the present two-stage experiments validate long task sequences.

C Related Work

Forgetting under RL and SFT. Lai et al. (2026) report better multimodal retention under RL than SFT, relating it to selective updates on learnable examples. Chen et al. (2026a) show that low forgetting persists without KL penalties or GRPO's advantage estimator; refreshing self-generated SFT data also helps. SFP asks which selected behaviors support old-task function.

On-policy learning and replay. On-policy distillation reduces train–inference mismatch through teacher feedback on student-generated sequences (Agarwal et al., 2024); SDFT reports reduced forgetting through demonstration-conditioned self-distillation (Shenfeld et al., 2026). On-Policy Replay regenerates and filters responses on old-task prompts before SFT (Chen et al., 2026b). Its low-score replay ablation loses the benefit. These results motivate studying both generation and selection, while explicit reuse of old prompts distinguishes replay from rehearsal through new-task trajectories.

Policy drift and functional change. Shenfeld et al. (2025) associate forgetting with base-policy KL drift and derive an on-policy bias toward nearby successful policies. Chen et al. (2026a) report moderate KL–forgetting correlation; Luo et al. (2026) distinguish current-task regularization from prior-task drift. SFP examines the functional directions and subsequent effects of a displacement.

Conditional retention and training history. Self-distillation policy optimization can forget more than GRPO and collapse under some teacher-update settings (Wang et al., 2026); standard RL also forgets substantially on the Multimodal Reasoning Continual Learning benchmark (Luo et al., 2026). Jin et al. (2026) find that RL often restores out-of-distribution performance lost during late SFT, with checkpoint-dependent recovery typically below the earlier SFT peak. These observations motivate the separate selection, function, and continuation questions formalized by SFP.