A DEAFENING SILENCE: CATASTROPHIC FORGETTING LIVES IN THE OUTPUT EMBEDDINGS OF TOKENS THE DATA NEVER SPEAKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Continual pre-training and fine-tuning in Large Language Models (LLMs) inevitably induce catastrophic forgetting, typically mitigated by replay using often-inaccessible original data. In this data-free regime, we analyze where forgetting occurs and why. Systematic parameter freezing across five settings up to 1.4B reveals that forgetting concentrates selectively in the output embeddings of tokens rarely seen in the new corpus, whereas the same $\sqrt{\hat{v}}$ band of the body is inert and new learning resides elsewhere. This localization is governed by the vocabulary deficiency of the corpus rather than the training mode, allowing pre-retraining risk ranking from token counts alone within a fixed base model. Mechanistically, absent tokens receive persistent one-sided softmax gradients that Adam's second-moment ($\sqrt{\hat{v}}$) normalization amplifies into full-sized updates. We therefore propose an intervention: raising Adam's ϵ exclusively for the output projection during training. Across eight settings spanning 160M to 12B parameters and four model families, this removes 39.4% to 67.9% of forgetting across all seven stable configurations without degrading target learning or requiring per-model tuning. The defense combines additively or better with replay (79.8% on Qwen/Korean) and rescues released-head LoRA from a 23-fold forgetting surge. Because post-hoc editing of the drifted rows recovers under 5% of forgetting, the intervention must operate during training. Our findings indicate that a single-line optimizer adjustment may serve as the primary defense against catastrophic forgetting where the corpus starves the vocabulary.

1 INTRODUCTION

Adapting pre-trained foundation models via continual pre-training or fine-tuning inevitably induces catastrophic forgetting (Kirkpatrick et al., 2017). While replay mitigates this by mixing original pre-training data (Ibrahim et al., 2024), such data is rarely accessible. In this data-free regime, we causally localize where forgetting occurs and target that exact site to prevent it (Figure 1).

Prior research on forgetting divides into mitigation and localization. Beyond replay (Ibrahim et al., 2024; Hickok, 2025), data-free defenses (including parameter selection (Chen et al., 2025), low-rank adapters (Hu et al., 2022; Biderman et al., 2024), and orthogonal subspace projections (Saha et al., 2021; Wang et al., 2023)) indiscriminately constrain aggregate parameter drift, sacrificing plasticity for retention. Conversely, localization studies report conflicting sites across early attention, lower layers, and FFNs (Gu & Feng, 2020; Zheng et al., 2025; Laitinen-Fredriksson Lundstrom-Imanov, 2026), primarily due to correlational metrics like weight displacement. Although theory suggests forgetting concentrates in low-rank output modes (Hidekel & Raviv, 2026), the specific tensors holding these modes remain unidentified.

Instead of unguided regularization, we intervene directly: freezing parameter groups to their pre-trained values isolates each group's causal share of forgetting. Varying sites, second-moment ($\sqrt{\hat{v}}$) bands, and corpora pinpoints forgetting to under-exposed output embeddings. Because these embeddings drift when Adam assigns full step sizes to minute softmax gradients, raising ϵ exclusively on this tensor suppresses drift without impeding learning.

Our contributions are as follows:

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

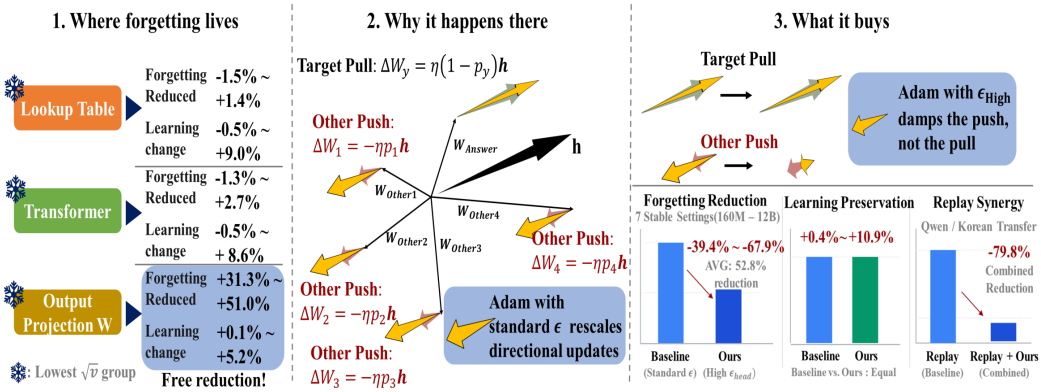

Figure 1: Overview: where the forgetting lives, why it lives there, and what acting on it buys.

- **Causal localization**: By training-time freezing, we show that forgetting concentrates in output embeddings of rare tokens (31–51% at five stable settings; the body in the same band remains $< 10\%$). This site is governed by corpus vocabulary deficiency, enabling pre-retraining risk ranking simply by counting tokens.
- **Underlying mechanism**: Absent token embeddings receive persistent one-sided softmax gradients, which Adam's $\sqrt{\hat{v}}$ normalization amplifies into full-sized updates that default ϵ fails to curb.
- **Output-scoped** ϵ: Elevating ϵ exclusively on the output projection eliminates 39.4–67.9% of forgetting across seven stable settings of an eight-setting grid, 160M to 12B parameters and four model families without degrading target learning or requiring original data. It composes additively with replay and prevents catastrophic forgetting in released-head LoRA.
- **Boundary conditions**: Post-hoc editing of the drifted rows recovers $< 5\%$ of forgetting, so the intervention must operate during training; when forgetting shifts to the body under in-distribution tuning, damping it costs learning at one of the two cells we measure, because body coordinates conflate drift with new knowledge (Section 5.5).

2 Related Work

Replay and defenses without original data. Continual pre-training defenses typically rely on replay and learning rate schedules (Ibrahim et al., 2024; Hickok, 2025). Without original data, existing methods restrict parameter drift via weight regularization, parameter selection (Chen et al., 2025), or low-rank adapters (Biderman et al., 2024), yet these blindly constrain aggregate movement without separating forgetting from learning. Indeed, releasing the output head alone under fixed-rank LoRA multiplies forgetting 23-fold on a tied model and 16-fold on an untied one (Section 5.3, Appendix A.9). We focus on data-free settings and view replay as complementary.

Localizing forgetting. Reported sites of forgetting conflict across early attention, lower layers, and FFNs (Gu & Feng, 2020; Zheng et al., 2025; Laitinen-Fredriksson Lundstrom-Imanov, 2026), largely due to correlational metrics like weight displacement. While output freezing mitigates degradation (Gu & Feng, 2020), the specific rows responsible and their causal mechanisms remained unexplained. Unlike class-incremental learning where output bias stems from task imbalance (Wu et al., 2019; Masana et al., 2023), continual pre-training lacks explicit label skew. Using training-time freezing, we show that corpus vocabulary composition causally dictates where forgetting resides.

The geometry of rare tokens and remedies. Persistent negative softmax gradients push rare token embeddings into a narrow cone (Gao et al., 2019; Yu et al., 2022). While low-rank dynamics are well-documented (Aghajanyan et al., 2021; Hidekel & Raviv, 2026), existing subspace methods (Saha et al., 2021; Wang et al., 2023) require past data, and ASR token freezing (Kwok et al., 2024) relies on binary heuristics. We show this drift directly carries forgetting, identify Adam as the amplifier and ϵ as the brake, and measure rank directly from drift. Unlike post-hoc editing (Ilharco et al., 2023; Wortsman et al., 2022) or optimizer tuning for model scaling (Choi et al., 2019; Wortsman et al., 2024; Yang & Littwin, 2023; Everett et al., 2024), ours operates during training.

3 PRELIMINARIES

3.1 PROBLEM SETTING

Setting and constraint. We adapt a pre-trained foundation model θ_{base} (trained on an inaccessible corpus $\mathcal{D}_{\text{old}}$) to a new corpus $\mathcal{D}_{\text{new}}$ via continual pre-training or fine-tuning, aiming to acquire new domain knowledge without catastrophic forgetting. We assume zero access to $\mathcal{D}_{\text{old}}$, requiring strictly data-free adaptation, although combinations with replay are evaluated in Section 5.2.

Notation. Consider a transformer with vocabulary size V and hidden dimension d, trained with AdamW. Given context representation h, the unembedding matrix $W \in \mathbb{R}^{V\times d}$ produces logits $z_t = h^\top w_t$, where $w_t \in \mathbb{R}^d$ is row t of W, the output embedding of token t. Let $c(t)$ denote the target occurrence count of token t in $\mathcal{D}_{\text{new}}$; tokens with $c(t) < K$ ($K = 1{,}000$) are designated as under-exposed. *Vocabulary deficiency* is the occurrence-weighted fraction of old-domain evaluation positions whose target token is under-exposed in $\mathcal{D}_{\text{new}}$, counted over the four-domain pool of Section 3.3 minus the injected domain, the wider $\mathcal{E}$ of Table 3. Computed simply by tokenizing both corpora with the base tokenizer without retraining, it separates tasks into domain injection (high deficiency) and in-distribution tuning (low deficiency).

3.2 BACKGROUND

Softmax gradients and Adam's ϵ. Given context h, output probabilities are $p = \mathrm{softmax}(Wh)$ with target loss $L = -\log p_y$. The gradient for output embedding w_t is

$$\frac{\partial L}{\partial w_t} = \frac{\partial L}{\partial z_t}\frac{\partial z_t}{\partial w_t} = \big(p_t - \mathbf{1}[t = y]\big)\, h. \tag{1}$$

Target embeddings are pulled toward h, whereas unselected tokens are pushed along $-p_t h$, persistently depressing their logits. In contrast, input embeddings update only upon occurrence.

AdamW pairs Adam's adaptive step with a decoupled decay (Kingma & Ba, 2015; Loshchilov & Hutter, 2019):

$$\Delta\theta = -\,\eta\left(\frac{\hat{m}}{\sqrt{\hat{v}} + \epsilon} + \lambda\theta\right). \tag{2}$$

For sign-consistent gradients, $\hat{m} \approx g$ and $\sqrt{\hat{v}} \approx |g|$, pushing the effective step size toward η regardless of gradient scale $|g|$. Oscillating gradients cancel in $\hat{m}$, so update magnitude reflects directional consistency rather than gradient norm. The denominator ϵ acts as a floor: when $\sqrt{\hat{v}} \lesssim \epsilon$, it scales updates by $\sqrt{\hat{v}}/(\sqrt{\hat{v}} + \epsilon)$, selectively braking parameters with small $\sqrt{\hat{v}}$ (default: $\epsilon = 1 \times 10^{-8}$).

3.3 EXPERIMENTAL SETUP

The grid of settings. Each setting pairs a foundation model with a new domain (Table 1). The Pythia suite (160M–12B) enables controlled scaling comparisons with an openly verifiable $\mathcal{D}_{\text{old}}$ (Biderman et al., 2023). Qwen2.5-0.5B and TinyLlama-1.1B provide tied and untied embeddings, respectively, while OLMo-2-1B represents a modern post-2024 architecture. Target domains span Korean, code, math, and instruction tuning (Alpaca and KoAlpaca), with Korean and Alpaca at opposite extremes of vocabulary deficiency.

Training procedure. All models are trained with AdamW. Continual pre-training uses a 40M token budget (Table 1), instruction tuning uses 12.6M, and budget checks extend to 160M and 400M tokens; all comparisons are strictly budget-matched. Interventions halt (freeze) or dampen (raise ϵ) updates

Table 1: The grid of settings. Korean is the primary injection domain across all base models; other domains vary vocabulary deficiency at fixed models (Section 4.2). All settings run replay-free for 40M tokens across three seeds (12.6M for instruction tuning; Table 4) unless noted.

Base model (architecture)	Primary domain	Additional domains
Pythia 160M–12B (GPT-NeoX)	Korean	math, code, Alpaca, KoAlpaca (410M and 1.4B only)
Qwen2.5-0.5B (tied)	Korean	math, code, Alpaca, KoAlpaca
TinyLlama-1.1B (Llama, untied)	Korean	–
OLMo-2-1B (olmo2, untied)	Korean	–

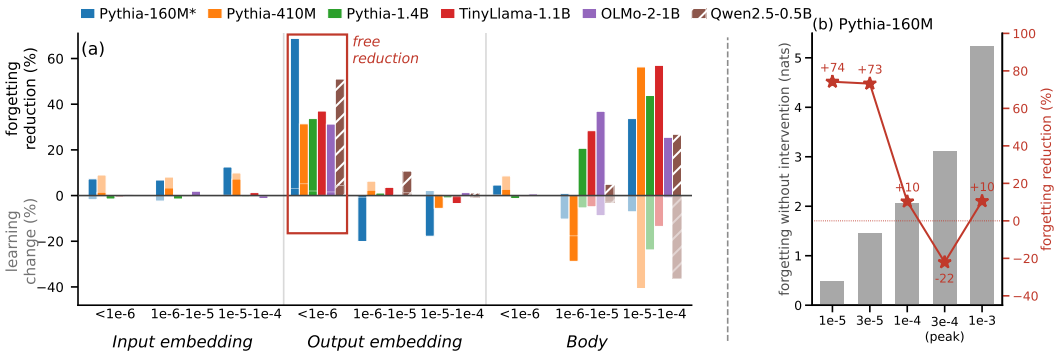

Figure 2: (a) Causal share by (site, $\sqrt{\hat{v}}$ band): bars denote forgetting reduction (up) and learning change (down) from freezing one group (Korean, replay-free). Hatched: Qwen (tied) at 1×10^{-4}. Pythia-160M ($*$) is at 1×10^{-5} due to peak-rate collapse. (b) Pythia-160M across rates, measured with output-scoped ϵ rather than freezing (Table A2). Full grid: Table A3.

to specific parameter groups throughout training. We separately sweep learning rates on a log grid (10^{-5} to 10^{-3} for pre-training, 10^{-6} to 3×10^{-4} for instruction tuning), selecting the peak on target learning and breaking ties within seed variance toward lower rates (Table A11, Table A1).

Evaluation corpora. Validation losses are evaluated in nats (cross-entropy per token) on fixed held-out corpora. The old-domain set $\mathcal{E}$ excludes the adaptation target: English and code for Korean/math injections; English and math for code; English, code, and math for instruction tuning. Reductions are strictly compared within one $\mathcal{E}$; the domain grid reads each row on the pool minus that row's injection, so its corpora are wider and differ by row (Table 3). These corpora sample $\mathcal{D}_{\text{old}}$ directly for Pythia and OLMo, and act as proxies for closed base models.

Metrics. Target learning is evaluated on $\mathcal{D}_{\text{new}}$ validation loss, while forgetting averages loss increases across all domains in $\mathcal{E}$:

$$\text{learn} = L_{\text{base}}(\mathcal{D}_{\text{new}}) - L_{\text{after}}(\mathcal{D}_{\text{new}}), \qquad \text{forget} = \frac{1}{|\mathcal{E}|} \sum_{d \in \mathcal{E}} \big[L_{\text{after}}(d) - L_{\text{base}}(d)\big], \tag{3}$$

$$\text{reduction} = 1 - \text{forget}_{\text{int}}/\text{forget}_{\text{ref}}, \qquad \text{change} = (\text{learn}_{\text{int}} - \text{learn}_{\text{ref}})/\text{learn}_{\text{ref}}, \tag{4}$$

where L_{base} and L_{after} denote pre- and post-retraining losses, and subscripts ref and int indicate reference and intervention runs at tuned learning rates. Positive learn and forget denote capability gains and incurred losses, while reduction and change are relative to the baseline.

Seeds and verdicts. We report three-seed averages unless noted; arms with seed disagreement were expanded to six or more. An intervention is judged to preserve learning if change remains within baseline seed variation: $(\max - \min)/\text{mean}$ of reference learn across seeds. This is 2–4% at most settings but far wider where the intervention arm is bimodal, reaching 18.9% at Pythia-410M on Korean.

4 Where Forgetting Lives, and Why There

4.1 Freezing One Group at a Time Locates the Forgetting

Measurement protocol: causal share. We define a parameter group's causal share as the forgetting reduction obtained by freezing only that group to its pre-trained values during training. Unlike correlational drift metrics, this directly tests necessity: a parameter shift holds causal share only if preventing it reduces forgetting.

We partition three components (input embeddings, output embeddings, body) across three $\sqrt{\hat{v}}$ bands (0–1×10^{-6}, 1×10^{-6}–1×10^{-5}, 1×10^{-5}–1×10^{-4}), freezing one pair at a time using fixed baseline masks. To separate location from parameter count, we include mass-matched controls freezing equivalent body parameters. This grid costs eleven arms per setting, so it runs at the six settings up to 1.4B rather than the eight of Section 5.2, on replay-free Korean adaptation at baseline peak learning rates, except Pythia-160M which collapses at its peak and is evaluated at 1×10^{-5} (detailed in (iv)). Tied Qwen treats embeddings as one component.

Table 2: Count-matched site contrast: parameters frozen (M) and the forgetting reduction (%) from freezing an equal mass at each site's low end (matched to the lowest output band). Each row divides by one reference trained at that setting's peak rate. n/a: tied embeddings.

Setting	Params frozen (M)	Output emb.	Input emb.	Body
Pythia-160M	30.7	−6.2	9.8	28.5
Pythia-410M	38.7	31.4	2.6	11.2
Pythia-1.4B	80.6	33.7	−0.5	0.4
TinyLlama-1.1B	48.2	37.0	0.4	2.9
OLMo-2-1B	176.6	31.3	−1.0	4.4
Qwen2.5-0.5B	110.0	51.0	n/a	12.3

Table 3: Reference forgetting strictly follows vocabulary deficiency, while reduction does not (output $\epsilon = 1 \times 10^{-4}$; references three seeds, interventions two, 410M/ko six per arm). Each cell is at its own peak rate, so rows are not rate-matched (Appendix A.1). Learning changes, where negative, stay within each reference's seed spread (18.9% for 410M/ko). Here $\mathcal{E}$ is the four-domain pool minus each row's injected domain, wider than the $\mathcal{E}$ of Table 5 and not identical across rows, so cells are comparable with neither.

Base model	Injected	Vocab. deficiency (%)	Ref. forget (nats)	Reduction (%)	Learning change (%)
Pythia-410M	math	33.9	0.0431	36.4	−3.7
	code	55.9	0.0803	55.7	+0.4
	ko	63.2	0.6565	31.0	−5.4
Pythia-1.4B	math	33.9	0.0560	24.6	−0.3
	code	55.9	0.0832	53.1	+4.5
	ko	63.2	0.5121	40.5	+2.2
Qwen2.5-0.5B	math	50.6	0.0258	67.8	+10.0
	code	52.5	0.1806	72.7	+18.4
	ko	61.3	0.6255	62.8	+5.7

Result: forgetting and learning live in different places. Causal shares reveal four primary findings:

(i) Forgetting localizes to the bottom output band. Freezing output embeddings with $\sqrt{\hat{v}} < 1 \times 10^{-6}$ eliminates 31.3–51.0% of forgetting across five settings, with target learning shifting by $+0.1\%$ to $+5.2\%$ (no degradation). Freezing input embeddings in the same band yields negligible reduction (-1.5% to $+1.4\%$; four untied settings), confirming that output embeddings drive forgetting.

(ii) Location governs forgetting, not low $\sqrt{\hat{v}}$ alone. The low-$\sqrt{\hat{v}}$ body band holds negligible share (-1.3% to $+2.7\%$). Although this band covers 74–86% of output parameters versus only 0.3–1.4% in the body, mass-matched body freezing yields only 0.4–12.3% reduction across five stable settings (Table 2; Pythia-160M is the exception at 28.5%, detailed in (iv)). Selection matters within the table too: at Qwen, random output rows remove 12.1% at a 4.5% learning cost against 42.7% for free from the lowest-$\sqrt{\hat{v}}$ coordinates (Appendix A.2). Disabling weight decay leaves forgetting and reduction essentially unchanged in Qwen and shifts Pythia-410M only within its seed spread (Appendix A.1, Table A7). Thus, forgetting is causally governed by row location, not parameter count or regularization artifacts.

(iii) Learning localizes to upper body bands. Freezing upper body bands eliminates 8.8–40.7% of target learning (the costlier upper band per setting), while freezing output embeddings leaves learning intact. Because forgetting and learning occupy disjoint parameter regions, halting output drift imposes no plasticity penalty.

(iv) Total collapse overrides localized mitigation. At its peak rate (3×10^{-4}), Pythia-160M suffers total collapse (3.1 nats baseline forgetting), where freezing fails. Lowering the learning rate restores mitigation. Measured with output-scoped ϵ rather than with freezing, reductions reach 74.2% (1×10^{-5}) and 73.2% (3×10^{-5}) where baseline forgetting stays below 1.5 nats, against 10.3% at 2.1 nats and -22.1% at 3.1 nats (Figure 2b, Table A2). Rerunning the freezing grid at 1×10^{-5} recovers the standard pattern (Table A3, column †): the bottom output band achieves a 68.8% share ($+3.1\%$ learning), input bands remain below 13%, the bottom body band holds 4.6%, and the cost of freezing reappears in the upper body bands. Breakdown occurs only under extreme baseline forgetting, not due to model scale. Summary ranges in (i)–(iii) exclude this outlier.

4.2 The Data, Not the Training Mode, Decides That Location

Vocabulary deficiency sets how much is forgotten. Holding the model, the budget and the four-domain pool fixed while varying the target domain across math, code, and Korean (Table 3; each cell keeps its own peak rate and drops its own injection from $\mathcal{E}$, so rows match on neither), baseline forgetting strictly follows vocabulary deficiency in all three models: 0.043, 0.080, 0.657 nats (410M); 0.056, 0.083, 0.512 nats (1.4B); and 0.026, 0.181, 0.626 nats (Qwen). Because vocabulary deficiency is computed simply by tokenizing both corpora beforehand, it correctly ranks injections within each model, although it does not predict cross-model absolute magnitudes (Spearman rank correlation

Table 4: Instruction fine-tuning (replay-free, 12.6M tokens): forgetting reduction (learning change). Freezes hold modules at base; ϵ slows low-$\sqrt{\hat{v}}$ output rows († freezes Qwen's tied table). Each cell is at its own peak rate; only 410M is rate-matched across the Alpaca/KoAlpaca pair (Appendix A.1).

	Pythia-410M		Pythia-1.4B		Qwen2.5-0.5B	
	Alpaca	KoAlpaca	Alpaca	KoAlpaca	Alpaca	KoAlpaca
Attention freeze	16.0 (−1.0)	42.7 (−0.7)	7.0 (−0.5)	30.2 (+0.4)	17.2 (−1.5)	15.5 (+0.1)
MLP freeze	27.9 (−5.8)	8.2 (−23.1)	39.7 (−3.7)	47.8 (−16.3)	44.8 (−15.4)	26.1 (−26.3)
Output-embedding freeze	2.7 (−1.2)	−3.7 (+3.0)	0.3 (+0.1)	−1.6 (+0.5)	11.6 (−3.1)†	46.9 (−3.9)†
Input-embedding freeze	1.6 (−0.0)	6.6 (+1.1)	0.8 (+0.2)	−1.4 (+0.5)	n/a	n/a
Output-scoped ϵ	14.6 (−0.1)	29.8 (+6.2)	5.6 (−0.3)	33.1 (+0.9)	6.0 (−1.0)	43.3 (−1.4)

0.60; Appendix A.3). Relative mitigation is decoupled from deficiency, with code yielding the largest reduction. At 410M on Korean, six seeds yield a wide reduction spread (4.6% to 46.6%, mean 31.0%) coupled with learning change: ordered by reduction, the same seeds order almost exactly by learning change (-32.9% to $+16.8\%$), so low-mitigation runs reflect a failure to take hold, not a retention-plasticity trade-off.

Separating training mode from data. Comparing Alpaca and KoAlpaca under instruction fine-tuning varies the data distribution at a fixed task format (Table 4), with the caveat that only Pythia-410M is rate-matched across the pair. In-distribution Alpaca concentrates causal share in MLPs, with output embeddings in single digits for untied models (11.6% in tied Qwen). KoAlpaca introduces novel vocabulary, shifting share to under-exposed output embedding rows. While freezing whole output modules in untied models yields negligible reduction (-3.7%, -1.6%), output-specific ϵ on low-$\sqrt{\hat{v}}$ rows eliminates 29.8–43.3% of forgetting without sacrificing target learning. The apparent 46.9% module freeze share in Qwen is an artifact of parameter tying. Furthermore, freezing MLPs on KoAlpaca reduces forgetting by 8.2–47.8% but sacrifices 16.3–26.3% of target learning, confirming that body freezing severely impairs acquisition.

What the attention column does not settle. Freezing attention on KoAlpaca removes 42.7% of forgetting at 410M (against 29.8% for output ϵ) and 30.2% at 1.4B (against 33.1%), at negligible learning cost. Because causal shares do not partition across unnormalized modules, an unmatched module freeze that holds a large share of the weights cannot be directly contrasted with a one-group optimizer change. Our claim is specific: the share recoverable by a row-level rule shifts to output embeddings when new vocabulary is introduced. Whether attention holds an independent share remains open (Appendix A.7).

4.3 Why Those Embeddings Move: Softmax and Adam

The softmax selects embeddings, Adam enlarges their step. Output embeddings absent as targets in $\mathcal{D}_{\text{new}}$ receive only the one-sided gradient $p_t h$ at each token position, which drives updates along $-p_t h$. Because context h is shared without opposing pulls, these gradients are small, sign-consistent, and directionally aligned. By Eq. (2), Adam maximally amplifies these updates: $\hat{m}$ and $\sqrt{\hat{v}}$ scale down together, driving effective step sizes toward the full learning rate. In contrast, input embeddings update only on explicit occurrences. In five untied settings, 34.8–51.2% of input parameters retain $\sqrt{\hat{v}} = 0$ post-training (identically 36.59% across the three Pythia models sharing a tokenizer; Table A4), whereas no output coordinate remains zero across all six settings (Table A21). Drift thus concentrates in output rows accumulating persistent negative gradients everywhere.

Empirical evidence: travel distance across ϵ. Travel distances over 120 steps confirm this mechanism. Under default $\epsilon = 10^{-8}$, parameter displacement stays within one order of magnitude across bands despite disparate gradient scales. Directional consistency (net displacement over total path length) peaks in the bottom band at 0.67–0.90, indicating unidirectional drift. Raising ϵ to 10^{-4} introduces sharp selectivity, damping displacement by 48.0–189.2× in the lowest band versus only 1.4–1.8× in the highest (Table A20, which bands the whole model and so cannot attribute damping to a site). Default ϵ remains inactive because only 3.5–15.4% of parameters in Pythia (5.0% in Qwen) fall below 10^{-8} (Table A4). Because median output coordinates sit two to three orders of magnitude below median body coordinates (Appendix A.13), elevating ϵ selectively brakes output drift. Softmax selects which embeddings drift, Adam dictates how far, and default ϵ fails to brake.

5 THE INTERVENTION AND ITS BOUNDARIES

5.1 THE INTERVENTION AND ITS EQUIVALENTS

The intervention. We set $\epsilon = 1 \times 10^{-4}$ exclusively for the output projection in optimizer parameter groups, introducing zero compute or memory overhead without requiring explicit target token lists. Under Eq. (2), ϵ scales each step by $\sqrt{\hat{v}}/(\sqrt{\hat{v}} + \epsilon)$: at $\epsilon = 1 \times 10^{-4}$ a coordinate at the output median (1.1×10^{-8} to 7.8×10^{-8}) keeps under a thousandth of its step, while one at the body median (5.5×10^{-6} to 4.2×10^{-5}) keeps a twentieth to a third. The gap between the sites (Appendix A.13), not a hand-placed threshold, supplies the selectivity; sensitivity is in Table A5.

Equivalents. The key mechanism is arresting under-exposed rows rather than ϵ itself. Two other instruments achieve equivalent results: freezing rows with $c(t) < 1{,}000$ at base values, and per-step gating updating rows only when tokens appear in the batch. Although operating differently, all three agree closely in forgetting reduction across five settings (Appendix A.2, Figure A1 and Table A8). We adopt ϵ because it requires no explicit token tracking.

Controls. Scaling down the embedding learning rate traces the same frontier, and at a matched 96.6% of the output embedding (under 1% replay) random and count-based masks retain similarly but differ sharply in learning cost: mass buys retention, the criterion decides its price (Appendix A.2).

Choosing the scope. The intervention site is predictable prior to retraining. If baseline forgetting is non-positive, no intervention is needed. Otherwise, tokenize the candidate corpus with the base tokenizer to compute the share of old-domain positions with target counts $c(t) < K = 1{,}000$. A high fraction indicates output-dominated forgetting, warranting output-specific ϵ. A low fraction indicates body-dominated forgetting, where freezing harms target learning (Section 4.2, Table 4), making replay or learning rate scaling preferable. This heuristic strictly matched empirical forgetting ranks at the three models of Table 3 and at four earlier (checkpoint, corpus) comparisons; quantitative analyses and failed extensions appear in Appendix A.3.

5.2 MAIN RESULT: FREE REDUCTION AT EIGHT SETTINGS

Generality across architectures and scales. We evaluate output-specific ϵ across eight settings under Korean domain injection (Table 5). Target learning is preserved at three seeds ($+0.4\%$ to $+10.9\%$; the bimodal 410M arm reads -7.0% at six, inside its 18.9% spread), with forgetting reduction spanning 39.4% to 67.9% across seven standard models. Pythia-160M collapses at its peak rate (Section 4.1), so its row reflects 3×10^{-5} and is excluded from summary ranges. Baseline forgetting in the Pythia suite decreases monotonically with tokens-per-parameter exposure, falling from 3.1 nats at 160M's peak rate to 0.10 nats at 12B (Tables A2 and 5). Under a fixed budget, exposure and model size co-vary, so this column alone cannot separate them.

The largest models are quiet because they are underfed, not because they are large. Reference forgetting at 6.9B and 12B is approximately 0.1 nats, which might appear negligible. However, extending the token budget demonstrates that forgetting at scale is deferred rather than absent. At each run's own fixed rate to 160M tokens (3×10^{-5} at 6.9B, 10^{-5} at 12B), baseline forgetting grows from 0.238 to 0.305 nats in 6.9B and from 0.044 to 0.076 nats in 12B between the 40M and 160M checkpoints. Rather than decaying, the reduction rises from 53.9% to 62.3% at 6.9B and from 76.2% to 84.1% at 12B, while learning changes stay within $+2.2\%$ and $+0.6\%$ across all checkpoints (Table A15). These are checkpoints inside 160M-token runs whose schedule has not yet decayed, so the ladder compares one run against itself. Likewise, Pythia-410M trained on code across a tenfold budget (400M tokens) incurs 1.27 nats of baseline forgetting, where the intervention eliminates 56.2% while improving target learning by 10.1% (Appendix A.12). Budget scale is therefore not a flattering confound; it is the axis along which the defense strengthens.

Operating regime and robustness. Mitigation operates within a distinct learning-rate regime: forgetting reduction holds steady across the low-rate plateau, then collapses at excessive rates where target learning fails alongside it, so ϵ cannot replace rate tuning, and at a 40M budget that plateau is narrowest at 6.9B and 12B. Because the optimal rate is invariant to the intervention, a single standard sweep suffices; nor does the method require per-model ϵ tuning. Appendix A.4 reads the rate grid (Table A11) and the ϵ sweep (Table A5). These benefits survive capability benchmarks and tenfold budget expansion (Appendix A.5); math and code injections follow in Table 3.

Table 5: Output-only ϵ (Korean, replay-free, 40M tokens, peak lr; 3 seeds, 2 for 12B/OLMo baselines). *160M is at 3×10^{-5} due to peak collapse (Figure 2b, Table A2). The 410M arm is bimodal across seeds: at six seeds of the same configuration it reads 32.7% and -7.0% (Table A7).

Setting	Tokens/param	Ref. forget (nats)	Interv. forget	Reduction (%)	Learning change (%)
Pythia-160M*	0.2469	1.448	0.388	73.2	+9.9
Pythia-410M	0.0988	0.664	0.373	43.8	+10.9
Qwen2.5-0.5B	0.0810	0.629	0.202	67.9	+5.4
TinyLlama-1.1B	0.0364	0.361	0.187	48.1	+0.4
Pythia-1.4B	0.0284	0.473	0.261	44.8	+2.2
OLMo-2-1B	0.0269	1.137	0.689	39.4	+1.8
Pythia-6.9B	0.0058	0.109	0.044	60.2	+0.8
Pythia-12B	0.0034	0.101	0.035	65.3	+1.1

Table 6: Replay × output ϵ (nats; reduction %; 3 seeds). Separate campaigns, Qwen at lr 3×10^{-5} and 40M, OLMo at lr 1×10^{-4} and 60M. Learning stays in spread.

	ϵ off	ϵ on
Qwen2.5-0.5B / Korean (indep. predicts 78.0%)		
replay 0%	0.1373	0.0418 (69.5%)
replay 1%	0.0993 (27.7%)	0.0277 (79.8%)
OLMo-2-1B / Korean (indep. predicts 64.8%)		
replay 0%	0.3997	0.2283 (42.9%)
replay 1%	0.2460 (38.4%)	0.1023 (74.4%)

Table 7: Best recovery from α-sweeping the shared drift ($c(t) = 0$, replay-free; rates and budgets in Table A13). Post-hoc edits recover $< 5\%$ of forgetting; this is one family of edit, with row reset and uniform interpolation in Appendix A.14.

Setting	Rows	Recovery (%)	Best α	Learn at α=1 (%)
OLMo-2-1B / ko	42,037	+0.7	0.25	−133.6
Qwen-0.5B / ko	85,143	+2.5	0.50	−18.7
Pythia-1.4B / ko	15,542	+4.6	0.50	−7.7
Time course, Qwen / Korean: best recovery by checkpoint (tokens)				
+1.7 (6.7M)	+1.8 (13.4M)	+2.0 (20M)	+2.1 (26.7M)	+2.5 (33.3M)	+2.5% (40M)

Interaction with experience replay. When replay is viable, both defenses combine effectively. On Qwen/Korean, the combined reduction (79.8%) matches the independence prediction (78.0%) within 2 percentage points, with ϵ's individual share remaining invariant to replay (69.5% vs. 72.1%, Table 6). In OLMo, the combination becomes super-additive (74.4% vs. 64.8% predicted). Across both settings, the workflow is identical: activate the zero-cost optimizer modification first, then deploy replay to curb residual forgetting at token expense.

5.3 Why the Replay-Free Baselines Stop Short

Comparison with baselines. Without access to $\mathcal{D}_{\text{old}}$, we evaluate five baselines on Qwen/Korean: L_2-to-base, KL-to-base, a constant low learning rate, bottom-layer freezing, and a low-rank adapter at $r = 16$; EWC is excluded because estimating Fisher weights requires original data. None reaches the free point: both anchors and the low rate buy retention only by sacrificing target learning, and bottom-layer freezing worsens both axes. They share one cause: all restrict aggregate parameter drift without a coordinate metric separating harmful drift from new learning, the axis that per-parameter $\sqrt{\hat{v}}$ supplies and output-scoped ϵ exploits. Appendix A.10 gives the protocol, with per-arm numbers in Table A6 and the frontier in Figure A3; the ordering generalizes to OLMo-2-1B (Appendix A.11).

The adapter is the informative failure. LoRA tests our premise directly. At its optimal rate it reduces forgetting by only 12.0% while giving up 27.1% of target learning, and that retention comes mostly from keeping the output head at base by default: releasing the head at that rate, ten times the full fine-tuning rate, spikes forgetting 23-fold, from 0.121 to 2.820 nats, through one tensor. Output-scoped ϵ on the released head removes 91.1% of that at the same rate, while reaching the highest target learning of the adapter arms. Low rank also contributes, though: on TinyLlama, head freezing without an adapter leaves 5.13 nats at the adapter rate (against 0.062 for LoRA) and turns learning negative, while at the full fine-tuning rate head freezing and output-scoped ϵ converge within 0.001 nats (47.0% and 47.2%), so rank constraints protect only at the tenfold higher adapter rates (Appendix A.9).

5.4 Acting After Training Fails

The edit does not recover what the dial prevents. In our replay-free setting, sweeping α over the shared drift component recovers at most $+4.6\%$ of forgetting across three settings, with the optimal α always representing partial removal; full removal degrades target learning by -7.7% to -133.6%

(Table 7). Editing earlier does not help: across six checkpoints the best recovery moves only from $+1.7\%$ to $+2.5\%$, and shared-direction energy saturates at 66% by 20M tokens.

Why, and it is not an implementation detail. Drift and repair share one tensor, and ablating each component separates their roles (Appendix A.6). Removing the shared direction leaves forgetting essentially unchanged ($0.98\times$ to $1.10\times$) while degrading target learning, so that direction carries new-domain probability mass rather than damage. Ablating the residual instead amplifies forgetting by $1.6\times$, $1.7\times$ and $2.4\times$ while leaving learning within 1.8%, so the residual stores the compensation that protects legacy capabilities under a drifting head. Any post-hoc edit strands or erases that compensation; raising ϵ during training prevents the drift, so no compensation is needed.

A second regime, outside our setting. Under a 1% replay regime (outside our data-free scope, Section 3.1), tracking edit dynamics over time confirms this limitation. Partial row resets recover at most $+3.5\%$ with tied embeddings and nothing with untied ones, while full resets cost rather than recover, monotonically more as training proceeds (-12.3% to -68.7% over six checkpoints; Appendix A.14). Neither regime provides an actionable repair window.

5.5 When the Forgetting Is in the Body

When $\mathcal{D}_{\text{new}}$ introduces no new vocabulary, causal share shifts into the body (Section 4.2). Body-scoped ϵ on Alpaca at two base models shows the site transfers but the free lunch does not: the body dial removes several times more forgetting than the output dial at both cells, yet is free at only one, paying beyond seed variance at the other. The mechanism explains the asymmetry: an output row for an absent token accumulates pure drift, so damping it costs nothing, whereas a body coordinate encodes new representations and prior capabilities together, so damping it suppresses acquisition. Losslessness is a property of the site, not of ϵ; where sites mix, replay and rate scaling remain the remedies (Appendix A.8).

6 Discussion and Limitations

Consistency with other remedies. Our framework unifies disparate findings along one variable, vocabulary deficit: rare-token gating (Yu et al., 2022) matches output ϵ, bottom-layer freezing (Zheng et al., 2025) fails under output-dominated forgetting, and weight interpolation (Wortsman et al., 2022) degrades gracefully where selective edits collapse because uniform scaling moves drift and compensation together, though it still trades learning for retention (Appendix A.14).

Limitations. First, the pre-retraining diagnostic predicts rank order only; absolute forgetting depends on base capacity and resists cross-tokenizer calibration (Appendix A.3). Second, in-distribution defenses are preliminary: the inward shift to the body reproduces at both tested cells, but lossless body ϵ holds at one (Section 5.5). Third, reduction is not uniform within a setting: at Pythia-410M on Korean two of six seeds recover little and forfeit target learning (Section 4.2), so cost-free mitigation describes most runs, not all. Fourth, scaling comparisons are Pythia-only, the ϵ sweep covers three Pythia checkpoints (Table A5), and the capability check predates the grid and runs at $\epsilon = 1 \times 10^{-5}$ on math with 1% replay, not data-free Korean (Appendix A.5). Fifth, the adapter ablation uses one rank ($r = 16$) at two base models (Appendix A.9). Sixth, the mechanism needs coordinate-wise adaptive optimizers, leaving Muon (Jordan et al., 2024) outside its scope.

Practical summary. Assess vocabulary deficit before retraining; if it is high, raise output ϵ, run a standard rate sweep without tuning ϵ, and add replay when available. No post-hoc edit we tried recovers what prevention achieves.

7 Conclusion

We localize catastrophic forgetting under domain shift to the output embeddings of under-exposed tokens, where non-target softmax gradients meet Adam's scale normalization. Raising ϵ there alone removes 39.4–67.9% of forgetting across seven stable settings, without original data or lost target learning, and it has to be done during training: drift and compensation share one tensor, so a later selective edit strands or destroys what sustains the old capabilities. The first defense against catastrophic forgetting may lie in optimizer configuration rather than data curation.

AUTHOR DISCLOSURE OF GENERATIVE AI TOOLS

In this work, large language model (LLM) tools were used for figure generation, final checking, translation, and LaTeX formatting. The authors take full responsibility for the content of this paper.

REFERENCES

Armen Aghajanyan, Luke Zettlemoyer, and Sonal Gupta. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021. arXiv:2012.13255.

Dan Biderman, Jacob Portes, Jose Gonzalez Ortiz, et al. Lora learns less and forgets less. *Transactions on Machine Learning Research*, 2024. arXiv:2405.09673.

Stella Biderman et al. Pythia: A suite for analyzing large language models across training and scaling. *International Conference on Machine Learning (ICML)*, 2023.

Yupeng Chen, Senmiao Wang, Yushun Zhang, Zhihang Lin, Haozhe Zhang, Weijian Sun, Tian Ding, and Ruoyu Sun. Mofo: Momentum-filtered optimizer for mitigating forgetting in llm fine-tuning. *Transactions on Machine Learning Research (TMLR)*, 2025. arXiv:2407.20999.

Dami Choi, Christopher J Shallue, Zachary Nado, Jaehoon Lee, Chris J Maddison, and George E Dahl. On empirical comparisons of optimizers for deep learning. *arXiv preprint arXiv:1910.05446*, 2019.

Katie Everett, Lechao Xiao, Mitchell Wortsman, et al. Scaling exponents across parameterizations and optimizers. In *International Conference on Machine Learning (ICML)*, 2024.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tie-Yan Liu. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*, 2019.

Shuhao Gu and Yang Feng. Investigating catastrophic forgetting during continual training for neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING)*, 2020. arXiv:2011.00678.

Truman Hickok. Scalable strategies for continual learning with replay. *arXiv preprint arXiv:2505.12512*, 2025.

Ido Nitzan Hidekel and Dan Raviv. Catastrophic forgetting is low-rank: A function-space theory for continual adaptation. In *International Conference on Machine Learning (ICML)*, 2026. arXiv:2606.18024.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022. arXiv:2106.09685.

Adam Ibrahim, Benjamin Thérien, Kshitij Gupta, Mats L. Richter, Quentin Anthony, Timothée Lesort, Eugene Belilovsky, and Irina Rish. Simple and scalable strategies to continually pre-train large language models. *Transactions on Machine Learning Research*, 2024.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. Editing models with task arithmetic. In *International Conference on Learning Representations (ICLR)*, 2023. arXiv:2212.04089.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks. https://kellerjordan.github.io/posts/muon/, 2024.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. arXiv:1612.00796.

Chin Yuen Kwok, Jia Qi Yip, and Eng Siong Chng. Continual learning optimizations for auto-regressive decoder of multilingual asr systems. In *Interspeech*, 2024. arXiv:2407.03645.

Gustav Olaf Yunus Laitinen-Fredriksson Lundstrom-Imanov. Mechanistic analysis of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2601.18699*, 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, and Joost van de Weijer. Class-incremental learning: Survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. arXiv:2010.15277.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. In *International Conference on Learning Representations (ICLR)*, 2021. arXiv:2103.09762.

Xiao Wang, Tianze Chen, Qiming Ge, Han Xia, Rong Bao, Rui Zheng, Qi Zhang, Tao Gui, and Xuanjing Huang. Orthogonal subspace learning for language model continual learning. In *Findings of the Association for Computational Linguistics: EMNLP*, 2023. arXiv:2310.14152.

Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo-Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. arXiv:2109.01903.

Mitchell Wortsman et al. Small-scale proxies for large-scale transformer training instabilities. In *International Conference on Learning Representations (ICLR)*, 2024.

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. arXiv:1905.13260.

Greg Yang and Etai Littwin. Tensor programs ivb: Adaptive optimization in the infinite-width limit. *arXiv preprint arXiv:2308.01814*, 2023.

Sangwon Yu, Jongyoon Song, Heeseung Kim, Seong-min Lee, Woo-Jong Ryu, and Sungroh Yoon. Rare tokens degenerate all tokens: Improving neural text generation via adaptive gradient gating for rare token embeddings. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

Junhao Zheng, Xidi Cai, Shengjie Qiu, and Qianli Ma. Spurious forgetting in continual learning of language models. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2501.13453.

Table A1: Reference forgetting and the composition of the lowest band ($\sqrt{\hat{v}} < 1 \times 10^{-6}$) per site: parameters held (M) and, for the input embedding, the share of that set whose $\sqrt{\hat{v}}$ is exactly zero. Reference forgetting is this campaign's own pair; at 160M, whose runs scatter widely at 3×10^{-4}, the six-seed mean of Table A2 is 3.108 nats against the 3.392 here.

Setting	lr	Ref. forget	Out. params (M)	In. params (M)	In. zero (%)	Body params (M)
160M	3×10^{-4}	3.392	30.7	30.9	45.8	1.2
410M	1×10^{-4}	0.656	38.7	41.3	45.7	2.7
1.4B	1×10^{-4}	0.472	80.6	88.0	42.8	15.9
TinyLlama	1×10^{-4}	0.355	48.2	50.4	45.3	13.1
OLMo	3×10^{-4}	1.160	176.6	199.6	52.8	6.9
Qwen	1×10^{-4}	0.628	110.0	n/a	n/a	1.0

Table A2: Pythia-160M across learning rates: output-only ϵ, Korean, replay-free, 40M tokens. The 1×10^{-3} uptick sits inside the seed spread. Cells here are matched pairs drawn from one campaign, with the same seed count on both arms, whereas the 160M row of Table A11 pools every available reference at each rate into a deeper but unbalanced denominator; that is why the two differ by up to four points at the shared rates.

lr	Ref. forget (nats)	Interv. forget	Reduction (%)	Learning change (%)	n (ref/interv)
1×10^{-5}	0.491	0.126	74.2	-2.5	3/3
3×10^{-5}	1.448	0.388	73.2	$+9.9$	3/3
1×10^{-4}	2.068	1.855	10.3	-7.5	6/6
3×10^{-4}	3.108	3.795	-22.1	-17.3	6/6
1×10^{-3}	5.233	4.681	10.5	$+20.5$	2/3

A Appendix

The appendix opens with the tables the main text cites and then groups the deferred details by topic. Section A.1 compiles those supplementary tables, while subsequent sections take the details in this order: method equivalence, the pre-retraining readout, the learning-rate operating regime, capability and length validation, drift geometry and component ablations, the attention column of the instruction grid, the body dial at an in-distribution corpus, low-rank adapters, the replay-free baseline suite and its repetition at a second base model, a second injected domain, the site-by-site second moment, and post-hoc editing.

A.1 Supporting tables

Learning rates behind the two domain-varying tables. Both follow the peak-on-learning rule of Section 3.3, which selects per cell, so neither table is rate-matched along a row and neither should be differenced across one. For Table 3: math and code at 3×10^{-5} in all three models except Qwen/math at 1×10^{-5}, and Korean at 1×10^{-4} throughout. For Table 4, as Alpaca/KoAlpaca: Pythia-410M 3×10^{-5} and 3×10^{-5}, Pythia-1.4B 1×10^{-5} and 1×10^{-4}, Qwen 3×10^{-6} and 3×10^{-5}. Only the 410M column is therefore rate-matched across the two corpora; the Korean cell of Table 3 also sits at a rate that buys 0.6% of target learning for roughly four times the forgetting of 3×10^{-5}, which inflates its denominator.

A.2 Instrument equivalence in detail

Controls beyond the three instruments. Table A8 reports the full comparison across models, and Figure A1 plots the three instruments side by side; for Pythia-160M, all instruments yield near-zero or noisy metrics due to optimization collapse. Freezing the same parameter mass under different criteria separates retention from selectivity, and the separation shows up in learning rather than in retention (Table A9). At a matched 96.6% of the output embedding, in the one campaign that carries 1% replay, a random mask removes 66.4% of forgetting and a count mask removes 68.2%, a gap of under two points; freezing the whole projection removes 74.4%, so mass is what buys retention. The criterion decides what the mass costs. The random mask pays -4.4% of target learning, the same price as freezing everything, while the count mask pays -0.2% and ϵ pays nothing at all. Which rows stay free, not how many are held, is what makes the retention free. Furthermore, scaling down the embedding learning rate traces the exact trade-off frontier of elevating ϵ: a learning-rate

Table A3: Causal share by (site, $\sqrt{\hat{v}}$ band): forgetting reduction (learning change in parentheses, %) from freezing that group. Korean, replay-free, 40M tokens, peak lr, two seeds per arm except Qwen at three. Every column is at that setting's peak rate except †, which repeats the 160M grid at 1×10^{-5}: the 160M peak collapses and every band row in its column goes negative (the two count-matched controls do not), so † is what Figure 2a plots for that model and what (iv) of Section 4.1 reads. Each column divides by its own campaign's three-seed reference trained at that column's rate. Dashes are arms not run. Freezing a band and freezing by token count are different membership rules, not two estimates of one quantity. Count-matched rows freeze the output lowest band's mass from the named site's low end. Stats: Table A1.

Site	Band	160M	160M†	410M	1.4B	TinyLlama	OLMo	Qwen
Output emb.	$<1 \times 10^{-6}$	-6.2 (-13.4)	68.8 (+3.1)	31.4 (+5.2)	33.7 (+2.0)	37.0 (+0.1)	31.3 (+1.6)	51.0 (+4.4)
	1×10^{-6}–1×10^{-5}	-13.6 (-9.7)	-20.1 (-0.8)	2.3 (+6.3)	1.1 (+0.1)	3.6 (-0.4)	0.3 (-0.1)	10.7 (+1.3)
	1×10^{-5}–1×10^{-4}	-10.9 (-7.7)	-17.8 (+2.2)	-5.6 (+0.6)	-1.0 (-0.2)	-3.5 (-0.5)	1.3 (-0.4)	1.1 (-0.8)
Input emb.	$<1 \times 10^{-6}$	-3.6 (-4.6)	7.3 (-1.8)	1.4 (+9.0)	-1.5 (+0.1)	-0.6 (-0.5)	0.4 (-0.2)	n/a
	1×10^{-6}–1×10^{-5}	-17.9 (-6.5)	6.8 (-2.4)	3.3 (+8.1)	-1.5 (+0.2)	-0.3 (-0.4)	1.9 (-0.2)	n/a
	1×10^{-5}–1×10^{-4}	-18.7 (-9.3)	12.5 (+0.2)	7.2 (+9.9)	-0.5 (+0.1)	1.3 (-0.2)	-1.3 (-0.2)	n/a
	count-matched	9.8 (+2.3)	—	2.6 (+7.2)	-0.5 (+0.0)	0.4 (-0.3)	-1.0 (-0.2)	n/a
Body	$<1 \times 10^{-6}$	-0.1 (-5.1)	4.6 (+0.4)	2.7 (+8.6)	-1.3 (+0.3)	0.2 (-0.5)	0.7 (-0.3)	0.2 (+0.0)
	1×10^{-6}–1×10^{-5}	-2.6 (-6.4)	0.8 (-10.3)	-28.8 (-17.6)	20.7 (-5.4)	28.4 (-4.9)	36.9 (-8.8)	4.9 (-2.9)
	1×10^{-5}–1×10^{-4}	-5.8 (-39.6)	33.7 (-7.1)	56.3 (-40.7)	43.8 (-23.8)	57.0 (-13.5)	25.5 (-1.0)	26.8 (-36.3)
	count-matched	28.5 (+4.2)	—	11.2 (+7.5)	0.4 (+0.2)	2.9 (-0.4)	4.4 (-0.4)	12.3 (-9.4)

Table A4: Distribution of $\sqrt{\hat{v}}$ at the end of the reference run (peak lr, Korean, replay-free, 40M tokens, seed 0). Zero coordinates never received a gradient (unfed input rows; tied Qwen has none) and carry no update for ϵ to damp; the two threshold columns exclude them. The last column shares the zeros of the input table alone (Table A1's In-zero column instead shares them of the input's lowest band).

Setting	Exactly 0 (%)	$< 1 \times 10^{-8}$ (%)	$< 1 \times 10^{-6}$ (%)	Zero, input emb. (%)
Pythia-160M	8.7	15.4	32.8	36.6
Pythia-410M	4.7	4.6	16.5	36.6
Pythia-1.4B	2.7	3.5	10.7	36.6
TinyLlama-1.1B	2.1	3.0	8.3	34.8
OLMo-2-1B	7.1	11.2	20.1	51.2
Qwen2.5-0.5B	0.0	5.0	22.5	n/a

factor of $\times 0.10$ achieves 67.5% reduction and $\times 0.36$ achieves 51.3%, with the ϵ sweep spanning the intermediate range (Table A10). Finally, freezing body parameters across matched $\sqrt{\hat{v}}$ bands yields negligible mitigation until reaching bands that severely degrade target acquisition (41.6% learning penalty), confirming that the mechanism is strictly confined to under-exposed output embeddings.

Random rows against the lowest band. The contrast quoted in Section 4.1(ii) comes from one replay-free Qwen campaign at lr 3×10^{-5}, three seeds per arm, all divided by the same reference (forget $+0.1374$). Holding 85,143 randomly chosen output rows at base removes 12.1% of forgetting and costs 4.5% of target learning; holding the lowest-$\sqrt{\hat{v}}$ coordinates of the same table removes 42.7% and gains 1.7%. Two cautions attach. The masks are not count-matched: the random arm holds 85,143 of 151,936 rows, or 76.3M parameters, while the band arm holds 108.8M, so the pair separates selection criteria and not parameter counts, and the count-controlled statement rests on Table 2 and Table A9 instead. The band arm also holds coordinates rather than whole rows, since $\sqrt{\hat{v}}$ membership is per parameter; against the wider ten-seed reference pool of the same cell the same arm reads 43.1% rather than 42.7%.

A.3 Reading the risk before retraining

What the deficiency predicts, and on what evidence. Vocabulary deficit is computed entirely without training. Following Section 3.1, it is the share of old-domain token positions, over the four-domain held-out set minus the injected domain, whose target has fewer than 1,000 occurrences in the 40M-token new stream, amounting to 63.2% for Korean, 55.9% for code, and 33.9% for math under the Pythia tokenizer (61.3%, 52.5%, and 50.6% for Qwen). On a shared base model, this fraction accurately predicts the relative order of baseline forgetting. Across a domain-axis grid where model architecture and training recipe are held constant, the predicted rank strictly matches

Table A5: Sensitivity to the ϵ value, Korean injection, replay-free, 40M tokens: forgetting reduction (learning change in parentheses, both %) against the same reference. At 1.4B the response is flat across the hundredfold span; 410M gives up eleven points at the 1×10^{-3} edge; in 160M's collapse regime no value rescues the run. The sweep exists only for these three Pythia checkpoints, so the flatness claim is not tested on the other three families.

Setting	lr	$\epsilon = 1 \times 10^{-5}$	$\epsilon = 1 \times 10^{-4}$	$\epsilon = 1 \times 10^{-3}$
Pythia-160M	3×10^{-4}	−6.2 (−10.7)	−22.1 (−17.3)	−30.0 (−24.4)
Pythia-410M	1×10^{-4}	41.5 (+13.2)	43.8 (+10.9)	32.7 (+2.5)
Pythia-1.4B	1×10^{-4}	40.9 (+2.1)	44.8 (+2.2)	45.6 (+2.2)

Table A6: No baseline reaches the free point. Qwen / Korean, replay-free, 40M tokens, three seeds per arm, all measured against one reference trained at lr 3×10^{-5} (forget $+0.1373$ nats, learn $+0.3189$). Arms whose own sweep peaks elsewhere carry their rate in the Setting column. Qwen's reference learns most at 1×10^{-4}; 3×10^{-5} is the rate at which this baseline campaign was run. Qwen ties its input and output embeddings, so on this model "output-only" scopes ϵ to the one shared table and the input/output separation of Table A18 cannot be drawn here.

Method	Setting	Forgetting reduction (%)	Learning change (%)
Output-only ϵ (ours)	$\epsilon = 1 \times 10^{-4}$	69.5	+0.4
Under-exposed freeze	$K = 1{,}000$	70.7	−1.4
Step-wise gating		70.2	−0.4
L_2-to-base	$\lambda = 10$	11.2	−1.3
L_2-to-base	$\lambda = 30$	29.3	−4.4
L_2-to-base	$\lambda = 100$	64.3	−15.4
KL-to-base	$\beta = 0.1$	96.5	−95.3
KL-to-base	$\beta = 0.3$	96.4	−97.3
KL-to-base	$\beta = 1$	96.4	−98.0
Constant low LR	3×10^{-6}	98.8	−53.8
Bottom-layer freeze	6 blocks	−2.5	−11.4
LoRA, $r = 16$	lr 1×10^{-3}	12.0	−27.1

empirical observations for all three models (Table 3), consistent with all four prior (checkpoint, corpus) comparisons.

Two registered failures bound the claim. First, attempting to predict absolute forgetting magnitudes rather than rank order failed. Across ten configurations spanning diverse base models, the Spearman rank correlation between deficit and forgetting is 0.60, falling short of our pre-registered threshold of 0.7. Because forgetting varies across base architectures under identical data, a data-only diagnostic cannot account for model-level factors. Second, attempting to cheaply compensate for these model factors using base-model loss per token also failed. Discrepant tokenization schemes render per-token cross-entropy incomparable across tokenizers, as exemplified by OLMo. Although OLMo's per-token Korean loss appears close to that of English (2.59 vs. 2.46 nats), converting to loss per byte reveals it as the checkpoint least familiar with Korean among the four families evaluated (1.93× vs. 1.35–1.74×). Even with byte normalization, rank predictive power was not restored. The diagnostic remains strictly a data-side, rank-only tool.

A.4 The operating regime across learning rates

The plateau and its edges. Table A11 sweeps output-only ϵ across learning rates in all eight settings. Forgetting reduction holds steady across the low-rate plateau (10^{-5} to 3×10^{-5}: 50.7% and 50.6% in 410M; 57.6% and 52.1% in 1.4B), but collapses at 10^{-3}, falling to 6.7% in 1.4B and 8.8% in 410M; OLMo maintains a wider plateau, retaining 26.9%. The intermediate 3×10^{-4} column is where settings part company rather than where they fail together: 410M is already down to 15.3% and 12B to 3.4%, while 1.4B holds 32.5%, OLMo 34.2% and Qwen 60.9%. Target learning mirrors this threshold: 410M changes remain within $\pm 1\%$ across the plateau, but drop by 25.6% at 3×10^{-4} and 49.1% at 10^{-3}. The plateau also ensures measurement stability: across the 37 cells with a readable reduction, every cell with a seed spread within 8 percentage points exceeds 27% reduction, whereas spreads over 17 points yield reductions below 17%. Two cells break monotonic ordering: Pythia-6.9B at 3×10^{-4} (47.0% vs. 41.6% at 10^{-4}) reflects optimization breakdown where target learning collapses to -108%, and the Pythia-160M uptick at 10^{-3} lies within seed variance (Table A2). The same shape recurs at the largest scale: 12B falls from 65.3% at 3×10^{-5} to 3.4% and 1.6% at the

Table A7: Disabling weight decay leaves forgetting and reduction unchanged, confirming drift is not a decay artifact (Korean, replay-free, 40M tokens, lr 10^{-4}). The decoupled term $\lambda\theta$ of Eq. (2) is therefore inert at this site, which is why Section 3.2 reads the adaptive step alone. The 410M gap reverses sign as seeds are added: at three seeds $\mathrm{wd} = 0.1$ led, 39.2% against 29.0% for $\mathrm{wd} = 0$, while at six seeds $\mathrm{wd} = 0$ leads, 38.4% against 32.7%. A gap that changes sign with three more seeds is noise rather than an effect.

Base model	wd	Reduct.	Learn.	n
Pythia-410M	0	38.4%	−5.7%	6
Pythia-410M	0.1	32.7%	−7.0%	6
Qwen2.5-0.5B	0	68.0%	+5.7%	3
Qwen2.5-0.5B	0.1	67.9%	+5.6%	3

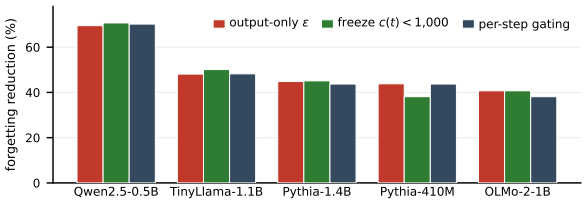

Figure A1: Three instruments stopping the same rows achieve equivalent reduction (Korean, replay-free; rates in Table A8).

two highest rates, where the reference itself collapses to 6.0 and 6.6 nats. Both failure modes fall on the same two models: at 10^{-5} their references forget too little to divide by, and at the top two rates they break down, so at a 40M budget the window that reads at all is 3×10^{-5} to 10^{-4}, narrower than at 410M or 1.4B where every rate is defined. Outside the plateau both retention and new learning fail, which is why ϵ cannot replace rate tuning.

No per-model ϵ tuning. The ϵ value itself requires no per-model adjustment: reduction is flat across a hundredfold span in 1.4B (40.9% to 45.6%) and gives up only 11 points at the 10^{-3} edge in 410M (Table A5).

A.5 Capability and length checks

Loss is a conservative measure. All measurements thus far are loss-based, so we separately verified whether the lower loss reflects recovered capability rather than mere recalibration. This check predates the grid and differs from the main result in four conditions: $\epsilon = 1 \times 10^{-5}$ rather than 1×10^{-4}, a math injection rather than Korean, 1% replay rather than none, and a 60M budget at 1.4B. Its loss-based reduction at 6.9B (73.9% on its own reference pair) is therefore not the 60.2% of Table 5; what it compares is two measures of one pair of runs, not two settings. On Pythia-6.9B across five tasks, the default baseline loses 3.21 capability points whereas the intervention loses only 0.52, so the capability-based reduction (83.6%) exceeds the loss-based 73.9% on the same pair (Table A12). On Pythia-1.4B, where each arm carries three benchmark evaluations, the two measures agree within 0.2 points (49.0% against 49.1%). Measuring in terms of loss therefore does not overstate the benefit of the intervention, and it may understate it.

The gain does not wash out with extended training. The main-result budget is 40M tokens. As a tenfold validation check, we trained Pythia-410M/code up to 400M tokens (replay-free, across three seeds). While the reference baseline forgets 1.2701 nats, output-only ϵ reduces forgetting by 56.2% and improves learning by 10.1%, with seed variance remaining within 2% along both axes. The advantage is not late-emerging: across all six evaluation checkpoints, the intervention's new-domain loss remains strictly below that of the reference baseline.

The same shape on twenty evaluation points. A second long run measures the budget axis more finely. On Qwen2.5-0.5B-Instruct with Korean injected to 400M tokens (lr 5×10^{-5}, 1% replay, $\epsilon = 1 \times 10^{-5}$, three seeds per arm, twenty evaluation points), the reduction rises monotonically at every point, reading 75.1% at 20M, 82.1% at 160M and 84.7% at 400M with seed spreads within two points, while the intervention's new-domain loss stays at or below the reference from 60M on (learning change −1.5% at 20M against +3.9% at 400M). Its conditions are the capability check's

Table A8: Comparison of three equivalence instruments (output-specific $\epsilon = 1 \times 10^{-4}$, freezing rows with $c(t) < 1{,}000$, and step-wise gating) under Korean injection (replay-free, 40M tokens, two seeds per arm except Qwen and every ϵ arm at three). The rate column is the campaign rate, which is the peak rate except at Qwen, whose reference learns most at 1×10^{-4}, and at 160M, which has no stable peak. Cells keep this campaign's seed pool; main-table rows pool more seeds, hence small offsets (OLMo 40.7 here vs. 39.4 in Table 5).

		Forgetting reduction (%)			
Setting	lr	Output-only ϵ	Freeze, K=1,000	Step-wise gating	Max gap (pp)
Qwen2.5-0.5B	3×10^{-5}	69.5	70.7	70.2	1.2
Pythia-1.4B	1×10^{-4}	44.8	45.1	43.7	1.5
TinyLlama-1.1B	1×10^{-4}	48.1	50.1	48.2	2.0
OLMo-2-1B	3×10^{-4}	40.7	40.7	38.1	2.6
Pythia-410M	1×10^{-4}	43.8	38.1	43.7	5.7
Pythia-160M	1×10^{-3}	8.0	−7.6	−4.6	15.6

Table A9: Retention under different selection criteria at matched mass: every arm below the first freezes 96.6% of the output embedding, while the first row freezes the whole projection, so only the lower four rows are mass-matched with each other (Qwen/Korean, 40M tokens, lr 3×10^{-5}, three seeds, one reference with forget $+0.0993$ nats). This campaign carries 1% replay, which is why its ϵ row is the 72.1% share of Table 6 and not the replay-free 69.5% of Table A6.

Arm	Forgetting reduction (%)	Learning change (%)
Freeze the whole output projection	74.4	−4.4
Freeze a random 96.6% of the embeddings	66.4	−4.4
Freeze by count, $K = 100$ (96.6%)	68.2	−0.2
Output-only $\epsilon = 1 \times 10^{-4}$	72.1	+0.3
Count-freeze and ϵ together	73.2	−0.8

rather than the grid's, so what this curve carries is the shape, that the dial does not decay as the budget grows, and not a level comparable with Table 5.

A.6 How the drift geometry is computed

The drift is low-rank. If the embeddings of absent tokens accumulate a shared push, their aggregate displacement vectors should align along a small number of shared directions rather than scattering independently across embeddings. Computing a singular value decomposition (SVD) on the output projection displacement before and after retraining confirms that displacement energy concentrates predominantly in a single direction (Table A13). Across three Korean-injection settings, the top singular vector of the absent rows accounts for 66.1% (Qwen), 72.6% (OLMo), and 80.8% (Pythia-1.4B) of total energy, compared to merely 3.2–12.9% in size-matched high-exposure controls ($c(t) \geq 1{,}000$). For embeddings receiving active learning signals, the shared push is diluted by task-specific updates, whereas absent rows retain only the shared push (Figure A2).

Two existing accounts describe the same structure. This low-rank structure aligns with two independently proposed accounts. The first stems from the embedding degeneration literature, which observes that rarely occurring token embeddings collapse toward a narrow cone in a single direction (Gao et al., 2019; Yu et al., 2022). While those studies analyze models trained from scratch, our measurements capture displacement driven by the same gradient geometry when retraining an already converged base model. The second comes from function-space theory predicting that forgetting concentrates along a small number of modes in output space (Hidekel & Raviv, 2026). Although we observe equivalent low-rank behavior in parameter-space displacement matrices, whether low-rank structure in parameter space implies that of function space lies beyond our scope.

Displacement matrix and singular value decomposition. From the pre- and post-retraining output projections W_{base} and W_{after}, we construct the difference matrix $D = W_{\text{after}} - W_{\text{base}}$ restricted to rows corresponding to target token sets: 85,143 rows absent as a target ($c(t) = 0$) for Qwen/Korean and 1,242 for Pythia-1.4B/math (Table A14). From the SVD of D, we report the top-1 energy ratio $\sigma_1^2 / \sum_i \sigma_i^2$. The control group applies identical computations to high-exposure embeddings

Table A10: Equivalence of embedding learning-rate scaling and ϵ elevation, alongside body negative controls across $\sqrt{\hat{v}}$ bands (Qwen/Korean, replay-free, three seeds). All arms share one reference, whose learning is 0.3507 and forgetting 0.1924 nats; that reference belongs to this campaign and is not the one behind Table A6.

Arm	Forgetting reduction (%)	Learning change (%)
embedding learning rate $\times 0.10$	67.5	$+0.7$
embedding $\epsilon = 5 \times 10^{-5}$	66.7	$+2.2$
embedding $\epsilon = 2 \times 10^{-5}$	64.2	$+2.8$
embedding $\epsilon = 5 \times 10^{-6}$	57.9	$+3.0$
embedding $\epsilon = 2 \times 10^{-6}$	52.0	$+2.8$
embedding learning rate $\times 0.36$	51.3	$+1.4$
freeze embedding, $\sqrt{\hat{v}} < 1 \times 10^{-6}$	45.4	$+2.4$
freeze embedding, 1×10^{-6}–1×10^{-5}	16.5	$+0.8$
freeze embedding, 1×10^{-5}–1×10^{-4}	1.2	-0.9
freeze body, $\sqrt{\hat{v}} < 1 \times 10^{-6}$	0.1	-0.0
freeze body, 1×10^{-6}–1×10^{-5}	5.4	-3.9
freeze body, 1×10^{-5}–1×10^{-4}	19.4	-41.6

Table A11: Forgetting reduction and learning change (parentheses, both %) across learning rates (Korean, replay-free, 40M tokens; 3–10 seeds, independently paired vs. Table 5). Values $> 100\%$ denote old loss below base. *n/q* marks a ratio with no readable meaning, and the reason differs by column: in the reduction position the baseline forgets too little to divide by (< 0.025 nats), while in the learning position the reference itself ends below its base model on the new domain (-0.45 and -0.63 nats at 12B), so change in Eq. (4) divides by a negative quantity and returns $+13.3\%$ and $+50.9\%$ for arms that in fact learned less.

Setting	1×10^{-5}	3×10^{-5}	1×10^{-4}	3×10^{-4}	1×10^{-3}
Pythia-160M	73.8 (-2.9)	71.4 ($+6.5$)	10.0 (-7.8)	-17.7 (-16.5)	8.0 ($+6.1$)
Pythia-410M	50.7 (-1.0)	50.6 ($+0.7$)	43.5 ($+10.3$)	15.3 (-25.6)	8.8 (-49.1)
Pythia-1.4B	57.6 (-0.1)	52.1 ($+0.6$)	44.9 ($+2.0$)	32.5 ($+3.8$)	6.7 ($+16.0$)
TinyLlama-1.1B	n/q (-0.9)	111.6 (-0.2)	47.7 ($+0.4$)	27.9 ($+2.3$)	6.1 ($+11.8$)
OLMo-2-1B	44.5 (-3.1)	46.0 (-0.8)	41.9 ($+0.5$)	34.2 ($+1.5$)	26.9 ($+4.5$)
Qwen2.5-0.5B	75.4 (-3.8)	69.5 ($+0.4$)	67.7 ($+5.7$)	60.9 ($+15.9$)	30.8 (-77.5)
Pythia-6.9B	n/q (-0.1)	60.2 ($+0.8$)	41.6 ($+2.4$)	47.0 (-108.0)	8.5 (-12.9)
Pythia-12B	n/q ($+0.1$)	65.3 ($+0.9$)	45.3 ($+3.3$)	3.4 (n/q)	1.6 (n/q)

($c(t) \geq 1{,}000$) within the same model, and both groups are cut to the smaller one's size before the SVD (2,080, 794, 381 and 1,242 rows by the row order of Table A13).

Direction predicted before retraining. Running a single forward pass over the new corpus with the base model yields the mean hidden state $\bar{h}$ at the final layer, defining the predicted direction as $-\bar{h}$. A coordinate-scaled variant accounts for Adam by dividing each coordinate by its mean $\sqrt{\hat{v}}$ before unit normalization. Directional alignment is measured by the absolute cosine similarity $|\cos|$ with the empirical top singular vector; the random expectation in dimension d is approximately $1/\sqrt{d}$.

Component ablation. We decompose each embedding's displacement into its projection along the top singular vector and an orthogonal residual, subtract one component from the retrained weights, and re-evaluate on the validation corpus. Ablating the shared component projects out the top-1 direction from D, whereas ablating the residual retains only the top-1 direction. Both variants compare post-ablation target learning and forgetting against the unablated retrained model.

How well the predicted direction matches. The cosine between the pre-training prediction $-\bar{h}$ and the empirical top singular vector reaches 0.40 in Qwen/Korean (0.50 under coordinate scaling) and 0.647 in Pythia-1.4B/math, substantially exceeding the random expectation $1/\sqrt{d}$ (0.033 at $d = 896$ for Qwen, 0.022 at $d = 2048$ for 1.4B). Neither setting reaches our pre-registered threshold of 0.7: the consistent negative sign corroborates the mechanism, but the prediction lacks the precision of an operational tool.

The two components do different jobs, but the split is setting-dependent. Residual ablation behaves consistently across all three settings: target learning remains largely intact (-0.9% to -1.8%), whereas forgetting escalates by factors of $1.6\times$ (Pythia-1.4B), $1.7\times$ (OLMo), and $2.4\times$

Table A12: Capability by benchmark rather than loss, at $\epsilon = 1 \times 10^{-5}$ against the default 1×10^{-8}: five likelihood-scored tasks at two Pythia checkpoints. Both campaigns run at lr 1×10^{-4} under 1% replay on a math injection, 6.9B to 40M tokens and 1.4B to 60M, three seeded training runs per arm; the 1.4B columns average three benchmark evaluations per arm and the 6.9B columns one. Accuracy in percent; higher is better.

	Pythia-6.9B			Pythia-1.4B		
Task	untouched	1×10^{-8}	1×10^{-5}	untouched	1×10^{-8}	1×10^{-5}
LAMBADA	61.05	60.00	65.17	60.78	52.53	56.79
HellaSwag	63.24	57.89	62.06	52.06	48.08	49.90
PIQA	76.22	72.09	73.23	71.00	67.19	69.04
ARC-easy	67.26	64.02	65.32	60.73	57.00	58.33
WinoGrande	61.64	59.35	61.01	56.99	55.46	56.64
average	65.88	62.67	65.36	60.31	56.05	58.14

Table A13: Drift geometry of the output embedding, replay-free checkpoints. Rows: the size of the absent set ($c(t) = 0$), which is not the number fed to the SVD (Table A14); that runs on a size-matched subsample of this set and of the control ($c(t) \geq 1{,}000$). Top-1: energy share on the group's first singular direction; RMS: mean displacement per row. Rates and budgets, shared with Tables 7 and A14: Qwen 3×10^{-5} at 40M, OLMo 1×10^{-4} at 60M, Pythia-1.4B/ko 1×10^{-4} at 40M, Pythia-1.4B/math 3×10^{-5} at 60M.

Setting	Rows	Top-1, absent (%)	Top-1, control (%)	RMS, absent	RMS, control
Qwen-0.5B / ko	85,143	66.1	3.2	2.17×10^{-3}	5.53×10^{-4}
OLMo-2-1B / ko	47,492	72.6	8.8	5.97×10^{-3}	2.38×10^{-3}
Pythia-1.4B / ko	15,542	80.8	12.9	6.30×10^{-3}	1.67×10^{-3}
Pythia-1.4B / math	1,242	72.0	3.7	3.08×10^{-3}	5.87×10^{-4}

(Qwen). In contrast, shared component ablation leaves forgetting essentially unaffected ($0.98\times$ to $1.10\times$) while degrading target domain learning by widely varying degrees, spanning -7.7% at Pythia-1.4B to -133.6% at OLMo (Table A14). Although the shared drift consistently contributes to boosting new-domain token probabilities across all three settings, the magnitude of this share varies; hence, our main text claims only the protective function of the residual. In 1.4B/math where vocabulary deficit is minimal, absent rows number only 1,242, rendering aggregate metrics insensitive to ablating either component: an exception reflecting measurement limits rather than a contrary mechanism.

A.7 On the attention column of the instruction grid

One column does not fit the summary above and we state it rather than leave it to be found. Freezing attention on KoAlpaca removes 42.7% of forgetting at 410M and 30.2% at 1.4B, at no measurable learning cost. At 410M that exceeds what output-scoped ϵ removes (29.8%); at 1.4B it does not (33.1%), so the two are not separated by a consistent ordering either. Two things follow. First, causal share does not partition: freezing any large group also stops whatever that group would have done downstream to the output rows, so shares measured this way sum past one hundred and a large attention share is not evidence against an output account. The count-matched controls of Section 4.1 exist precisely because unmatched module freezing cannot separate site from mass, and attention here is neither count-matched nor row-resolved. Second, the claim this paper defends about instruction tuning is narrower than "the output dominates". It is that the share which is recoverable by a row-level rule moves to the output when the corpus brings new vocabulary, which is what the ϵ row shows (single digits on Alpaca against 29.8–43.3% on KoAlpaca) and what the module rows cannot show. Whether attention holds an independent share of instruction-tuning forgetting is a question our design does not answer, and we do not claim it does.

The output column of the same table carries a second contrast worth stating. At 410M on KoAlpaca, freezing the whole output module reads -3.7% while scoping ϵ to its low-$\sqrt{\hat{v}}$ rows reads $+29.8\%$, though the freeze is the harder instrument on every row the two share. What separates them is the rows they do not share: a module freeze also pins the fed rows, and Appendix A.6 finds those rows carrying the compensation that holds the old domain up under a drifting head, so stopping the drift

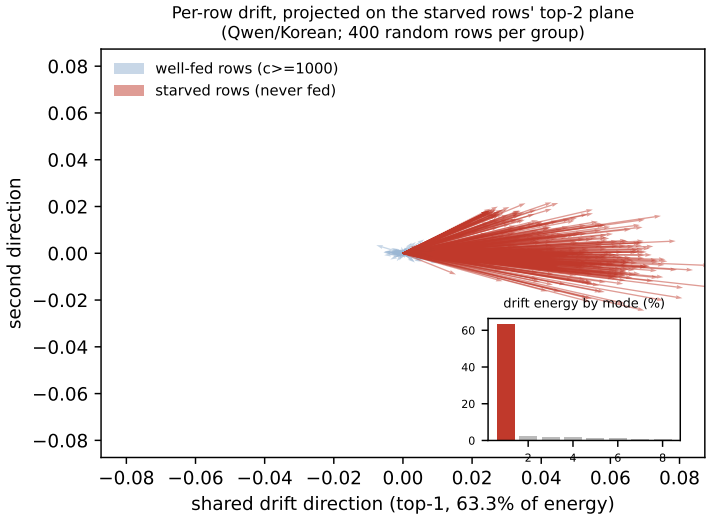

Figure A2: Per-token displacement on the top-2 drift plane (Qwen / Korean, 400 rows per group): absent rows (red) move together, well-fed rows (blue) stay near the origin. Inset: drift energy by mode.

Table A14: Component ablation: the shared (top-1) or residual component is subtracted from the trained weights and the model re-evaluated; negative recovery means forgetting grows. Rows and top-1 differ from Table A13 because that table subsamples both groups to the smaller one (794 rows at OLMo) while the ablation acts on every absent row, and because the OLMo row counts $c(t)$ over a 40M window there against the run's full 60M here.

			Shared removed		Residual removed	
Setting	Rows	Top-1 (%)	Recovery (%)	Learning (%)	Recovery (%)	Learning (%)
Qwen-0.5B / ko	85,143	66.1	-1.7	-18.7	-138.9	-1.8
OLMo-2-1B / ko	42,037	74.6	-9.6	-133.6	-74.4	-1.7
Pythia-1.4B / ko	15,542	80.3	$+1.6$	-7.7	-60.4	-0.9
Pythia-1.4B / math	1,242	72.0	-0.0	-0.1	-1.2	-0.0

and the compensation together nets to nothing. Blocked acquisition is not the explanation, since the module freeze leaves target learning unharmed at both untied models ($+3.0\%$ at 410M and $+0.5\%$ at 1.4B); what it removes is a repair, not a gain.

A.8 In-distribution forgetting and the body dial

Section 4.2 shows that an in-distribution corpus leaves the causal share in the body rather than the output. That raises the obvious follow-up: if the site moves, does the same dial work at the new site? Table A16 answers it in two cells, both Alpaca under instruction fine-tuning at 12.6M tokens and lr 1×10^{-5}, replay-free, three seeds per arm.

In both settings, the body dial eliminates roughly three to seven times more forgetting than the output dial, confirming that in-distribution forgetting resides in the body. However, while the body dial is cost-free within seed noise for Qwen (-0.3% target learning change for 35.6% reduction), it imposes a -6.7% learning penalty on Pythia-1.4B, exceeding measured seed variance. Thus, the localization holds across the two models and losslessness in only one, which is how Section 5.5 reads it.

Three details belong here rather than in the main text. First, each cell is paired inside its own campaign: the Qwen reference is the one its launch script names, and its output-scoped arm reproduces the 12.1% that script predicted before the runs were made. Second, the body dial is dose-dependent in the direction the denominator implies: at Qwen, $\epsilon = 1 \times 10^{-5}$ removes 14.5% against 35.6% at 1×10^{-4}, so the body's larger $\sqrt{\hat{v}}$ needs a larger floor before it brakes at all, which is the same ordering Table A21 shows between the sites. Third, we ran the body dial at 1.4B at two higher rates as well, but those arms have no reference trained at their own rate, so we do not convert them into

Table A15: The reduction does not shrink with budget: evaluation checkpoints taken inside 160M token runs, Korean, replay-free; three seeded runs per arm at 6.9B at its peak rate 3×10^{-5}, one at 12B at 1×10^{-5}. Learning changes at the same checkpoints stay within $+2.2\%$ (6.9B) and $+0.6\%$ (12B). The 40M row is a checkpoint of a run still scheduled to 160M, so its learning rate has not yet decayed; this is why it forgets more than the independent 40M runs of Table 5 and why neither those nor the matching rate column of Table A11 are comparable with it.

	Pythia-6.9B			Pythia-12B		
Budget	Ref. forget	Interv.	Reduction	Ref. forget	Interv.	Reduction
40M	0.2383	0.1100	53.9%	0.0443	0.0105	76.2%
80M	0.3106	0.1283	58.7%	0.0703	0.0149	78.9%
120M	0.3114	0.1192	61.7%	0.0757	0.0125	83.5%
160M	0.3051	0.1152	62.3%	0.0755	0.0120	84.1%

Table A16: Scoped ϵ on in-distribution Alpaca across sites (replay-free, 12.6M tokens, lr 1×10^{-5}, 3 seeds). Each cell is measured against its own reference: Qwen forget $+0.0620$ nats and learn $+0.1685$; Pythia-1.4B forget $+0.1356$ and learn $+0.3256$.

Setting	Arm	Forgetting reduction (%)	Learning change (%)
Qwen2.5-0.5B / Alpaca	output $\epsilon = 1 \times 10^{-4}$	12.1	$+0.8$
	body $\epsilon = 1 \times 10^{-5}$	14.5	$+3.0$
	body $\epsilon = 1 \times 10^{-4}$	35.6	-0.3
Pythia-1.4B / Alpaca	output $\epsilon = 1 \times 10^{-4}$	5.0	-0.3
	body $\epsilon = 1 \times 10^{-4}$	32.5	-6.7

reductions and do not report them; the same rule excludes several other unpaired arms from this paper. Section 6 registers body-scoped ϵ as preliminary on the strength of the one lossless cell.

A.9 Low-rank adapters, and where their retention comes from

The adapter test. A low-rank adapter is the closest data-free baseline to our claim, because it constrains *how much* the weights may move without saying *which* rows may move. If forgetting lives where we say it lives, then an adapter's retention should come from the output head being left alone rather than from the low rank, and it should evaporate the moment the head is allowed to move. Table A17 runs that test at Qwen2.5-0.5B on Korean, replay-free, at 40M tokens. The adapter is swept over its own rates and peaks at 10^{-3} on target learning (0.1224, 0.1822, **0.2329**, 0.2262 at 10^{-4} through 3×10^{-3}); all three adapter arms share that rate, so they are comparable with each other.

What it shows. With the head held at base, the default configuration for an adapter, forgetting is 0.1210 nats. Releasing the head to the same adapter returns 2.8204 nats, twenty-three times as much, from a change that touches one tensor, and output-scoped ϵ at that released head removes 91.1% of that forgetting while leaving the highest target learning of the three arms.

Three caveats bound how far this reads. First, it is a within-adapter contrast. The single variable between the two arms is whether the head may move, so the head is what causes the twenty-three-fold gap; and a matched-rate run that freezes the head *without* an adapter now separates the two terms: at TinyLlama it leaves 5.1293 nats against that model's adapter arm at 0.0623 and a negative target learning, so the rank constraint carries a share of the retention at this rate. The head carries the gap between the two adapter arms, not the whole of the adapter's retention. Second, forgetting at the released head exceeds even the full fine-tuning reference, because the adapter runs at a rate ten times higher; only the three adapter arms may be compared with one another. Third, Qwen ties its embeddings, so releasing the head makes a separate trainable copy of the shared table and breaks the tie, which means the released arm changes two things at once. TinyLlama unties them and reproduces the contrast: releasing the head there multiplies forgetting 16.3-fold, and output-scoped ϵ removes 92.4% of that at the highest target learning of the three arms. The effect is therefore not an artefact of tying.

Table A17: Low-rank adapters at two base models on Korean, replay-free, 40M tokens, $r = 16$, adapter arms at their swept peak lr 1×10^{-3}, three seeds per arm; the head-freeze controls carry no adapter. Qwen ties its embeddings and TinyLlama does not. The Qwen full fine-tuning rows are at lr 1×10^{-4} and are shown for scale only. Forgetting and learning in nats.

Arm	Forgetting	Learning	n
Qwen2.5-0.5B, tied			
Full fine-tuning reference	$+0.6285$	$+0.3764$	3
Full fine-tuning + output ϵ	$+0.2019$	$+0.3968$	3
Adapter, output head frozen	$+0.1210$	$+0.2326$	3
Adapter, output head released	$+2.8204$	$+0.1938$	3
Adapter, head released + output ϵ	$+0.2517$	$+0.2660$	3
TinyLlama-1.1B, untied			
Adapter, output head frozen	$+0.0623$	$+0.3213$	3
Adapter, output head released	$+1.0179$	$+0.3186$	3
Adapter, head released + output ϵ	$+0.0773$	$+0.3356$	3
Head frozen, no adapter	$+5.1293$	-0.1371	3
TinyLlama-1.1B, untied, at full fine-tuning's own lr 1×10^{-4}			
Full fine-tuning reference	$+0.3550$	$+0.4456$	3
Head frozen, no adapter	$+0.1883$	$+0.4384$	3
Full fine-tuning + output ϵ	$+0.1874$	$+0.4474$	3

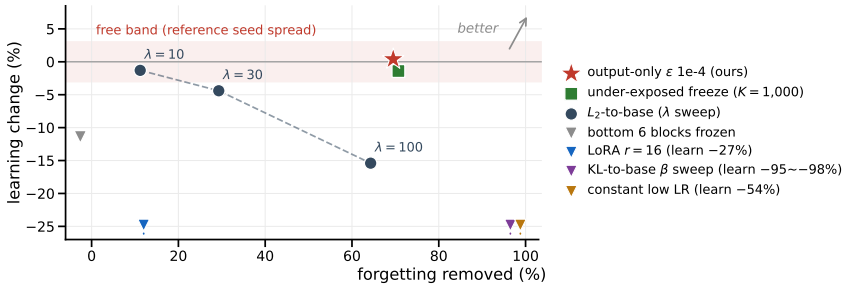

Figure A3: Reduction–learning plane (Qwen/Korean, replay-free, 3 seeds). Shaded band is reference seed spread; methods below the axis are marked at bottom and quantified in legend.

A.10 The replay-free baseline suite in detail

Protocol. The five baselines of Section 5.3 are evaluated on Qwen/Korean (vocabulary deficit 61.3%): L_2-to-base (weight penalty λ caching base weights), KL-to-base (divergence penalty β via parallel base forward passes), a constant low learning rate without reheating, freezing the lowest six blocks, and a low-rank adapter at $r = 16$. EWC is excluded because estimating Fisher weights requires original data that this setting defines as unavailable. Across three seeds at a 40M token budget, each arm trains at its sweep-selected peak rate against a single tuned reference (Figure A3; Table A6).

Four ways to stop short. L_2-to-base presents a rigid trade-off, sacrificing 15.4% of target learning for 64.3% forgetting reduction because it restrains under-exposed embeddings and task-active weights indiscriminately. KL-to-base arrests adaptation entirely, eliminating 96.4% to 96.5% of forgetting but discarding 95.3% to 98.0% of learning, since anchoring to base predictions directly contradicts learning new domain representations. A constant low rate similarly halts adaptation by under-training, forfeiting 53.8% of target learning. Bottom-layer freezing fails on both fronts, worsening forgetting by 2.5% while losing 11.4% of learning.

A.11 The baseline suite at a second base model

Table A6 compares baselines at one setting. Table A18 repeats the suite at OLMo-2-1B, the setting with the largest reference forgetting in the grid. Two conditions differ. The OLMo baseline campaign was run at 1% replay, the only regime in which every arm exists, so its reductions are measured against a reference that already carries a little replay; and it ran to 60M tokens rather than 40M. Neither difference is comparable cell by cell with Table A6. The ordering is what transfers. Output-scoped ϵ, under-exposed freezing and step-wise gating again land together and again cost nothing; the L_2

Table A18: The baseline suite at OLMo-2-1B / Korean, lr 1×10^{-4}, 60M tokens, 1% replay, three seeds per arm, against one reference (forget $+0.2455$ nats, learn $+0.7909$). Reductions above 100% mean the arm left the old-domain loss below the base model's.

Method	Setting	Forgetting reduction (%)	Learning change (%)
Output-only ϵ (ours)	$\epsilon = 1 \times 10^{-4}$	58.3	$+0.7$
Under-exposed freeze	$K = 1{,}000$	58.4	$+0.6$
Step-wise gating		57.6	$+0.7$
Input-only ϵ	$\epsilon = 1 \times 10^{-4}$	-0.5	-0.0
L_2-to-base	$\lambda = 0.1$	0.5	-0.1
L_2-to-base	$\lambda = 1$	5.8	-0.3
L_2-to-base	$\lambda = 10$	42.0	-3.1
KL-to-base	$\beta = 0.1$	87.3	-95.3
KL-to-base	$\beta = 1$	86.2	-97.7
Constant low LR	3×10^{-6}	100.4	-52.9
Bottom-layer freeze	6 blocks	11.3	-10.8

Table A19: Code injection, replay-free, output-only $\epsilon = 1 \times 10^{-4}$, arms paired within campaign. Rows above the rule are at each campaign's own rate. Above 100% the old-domain loss ended below the base model's.

Setting	lr	Budget	Ref. forget	Interv. forget	Reduction (%)	Learning change (%)
Pythia-160M	1×10^{-5}	40M	$+0.1876$	-0.0287	115.3	-0.6
Pythia-410M	3×10^{-5}	40M	$+0.0803$	$+0.0356$	55.7	$+0.4$
Pythia-410M	1×10^{-4}	40M	$+0.3248$	$+0.1938$	40.3	-0.9
Pythia-1.4B	3×10^{-5}	40M	$+0.0832$	$+0.0391$	53.1	$+4.5$
Qwen2.5-0.5B	3×10^{-5}	40M	$+0.1806$	$+0.0493$	72.7	$+18.4$
Pythia-410M	1×10^{-4}	400M	$+1.2701$	$+0.5557$	56.2	$+10.1$

anchor reaches two thirds of that reduction only at $\lambda = 10$ and pays for it; the functional anchor and the constant low rate again trade essentially all of the target domain away; and bottom-layer freezing is again both weak and expensive. Input-scoped ϵ, the negative control, does nothing, which is the same null this paper reports from freezing.

A.12 The same intervention on a second injected domain

Korean is the main injection axis of the grid, chosen so that the base model, the budget and the evaluation set stay fixed while only the corpus changes. Because Korean sits at the vocabulary-deficient extreme, Table A19 collects every code injection we ran, the next most deficient domain (55.9% against 63.2% under the Pythia tokenizer). The reduction holds across four base checkpoints, across a tenfold range of budget and at two learning rates on the same checkpoint, and target learning is never meaningfully lost. Two rows deserve their own reading. At 160M the intervention drives old-domain loss below the base model's, which is why its reduction exceeds 100%. And 410M at 400M tokens is the tenfold budget check of Appendix A.5: the reduction there is larger than at 40M on the same checkpoint and the same rate, not smaller.

A.13 Where the second moment sits, site by site

Table A20 bands the whole model at once, so it shows that Adam erases gradient scale but cannot by itself show that the erasure is an output phenomenon. Table A21 splits the same quantity by site at the end of the reference run. The two distributions barely overlap. 74 to 86 percent of the output embedding sits below 1×10^{-6}, against 0.3 to 1.4 percent of the body, and the median coordinate differs by two to three orders of magnitude between the two: 1.1×10^{-8} to 7.8×10^{-8} in the output against 5.5×10^{-6} to 4.2×10^{-5} in the body. A single scalar placed at 1×10^{-4} therefore leaves an output coordinate under a thousandth of its step while a body coordinate keeps a twentieth to a third of its own, and that hundredfold gap, not a threshold either site sits under, is what makes one number selective without a token list. The causal counterpart of this table is Table A3, which freezes the same sites in the same bands, and Table A10, which does so within one setting. What the present data cannot do is split the travel distances of Table A20 the same way, because the probe that produces them flattens across tensors; that measurement is left open.

Table A20: Mean path length per parameter over 120 steps, whole-model $\sqrt{\hat{v}}$ bands, replay-free, at the campaign rate (410M and 1.4B 1×10^{-4}, Qwen 3×10^{-5}; Qwen's own peak is 1×10^{-4}). Damping is the ratio of the two distances; net/total is at the default ϵ (1.0 = pushed the same way every step, 0.09 = random walk).

Setting	$\sqrt{\hat{v}}$ range	Share	Dist. (ϵ=1×10^{-8})	Dist. (ϵ=1×10^{-4})	Damping	Net/total
Pythia-410M / ko	$< 1 \times 10^{-6}$	6.0%	1.75×10^{-3}	3.59×10^{-5}	48.7	0.702
	1×10^{-6}–1×10^{-5}	14.5%	2.25×10^{-3}	9.95×10^{-5}	22.6	0.453
	1×10^{-5}–1×10^{-4}	77.5%	2.49×10^{-3}	5.60×10^{-4}	4.4	0.320
	$> 1 \times 10^{-4}$	2.0%	2.43×10^{-3}	1.38×10^{-3}	1.8	0.288
Pythia-1.4B / ko	$< 1 \times 10^{-6}$	3.3%	1.57×10^{-3}	3.27×10^{-5}	48.0	0.673
	1×10^{-6}–1×10^{-5}	28.9%	2.36×10^{-3}	1.55×10^{-4}	15.2	0.353
	1×10^{-5}–1×10^{-4}	67.5%	2.47×10^{-3}	3.43×10^{-4}	7.2	0.313
	$> 1 \times 10^{-4}$	0.3%	2.34×10^{-3}	1.43×10^{-3}	1.6	0.249
Qwen2.5-0.5B / ko	$< 1 \times 10^{-6}$	21.3%	1.64×10^{-3}	8.67×10^{-6}	189.2	0.895
	1×10^{-6}–1×10^{-5}	50.4%	8.56×10^{-4}	4.22×10^{-5}	20.3	0.456
	1×10^{-5}–1×10^{-4}	27.7%	7.81×10^{-4}	1.34×10^{-4}	5.8	0.419
	$> 1 \times 10^{-4}$	0.6%	7.64×10^{-4}	5.32×10^{-4}	1.4	0.433

Table A21: Distribution of $\sqrt{\hat{v}}$ within each site at the end of the reference run (Korean, replay-free, 40M tokens, seed 0, campaign rate, which is 3×10^{-5} at Qwen against its peak 1×10^{-4}, so its band shares do not reproduce the peak-rate mass held in Table 2). Band columns are percentages of that site, computed from the float32 dump of the final second moment; the Zero column counts coordinates that are exactly zero and is contained in the $< 1 \times 10^{-6}$ column beside it, so the two must not be added. Qwen ties its embeddings and has a single table. Medians are over nonzero coordinates.

Setting	Site	Params (M)	Zero	$< 1 \times 10^{-6}$	1×10^{-6}–1×10^{-5}	1×10^{-5}–1×10^{-4}	Median
Pythia-160M	output emb.	38.6	**0.0**	79.4	16.3	3.9	2.6×10^{-8}
	input emb.	38.6	36.6	79.9	14.9	4.7	1.1×10^{-7}
	body	85.1	0.0	1.4	8.1	76.5	4.2×10^{-5}
Pythia-410M	output emb.	51.5	**0.0**	75.2	18.2	6.2	6.8×10^{-8}
	input emb.	51.5	36.6	80.1	15.8	3.8	1.0×10^{-7}
	body	302.3	0.0	0.9	8.6	89.7	2.6×10^{-5}
Pythia-1.4B	output emb.	103.0	**0.0**	78.2	17.7	3.7	4.5×10^{-8}
	input emb.	103.0	36.6	85.4	13.3	1.1	4.9×10^{-8}
	body	1208.6	0.0	1.3	56.8	41.8	9.1×10^{-6}
TinyLlama-1.1B	output emb.	65.5	**0.0**	73.5	22.1	3.9	7.8×10^{-8}
	input emb.	65.5	34.8	76.9	18.4	4.2	2.4×10^{-7}
	body	969.0	0.0	1.4	33.1	65.4	1.1×10^{-5}
OLMo-2-1B	output emb.	205.5	**0.0**	85.9	11.6	2.1	1.1×10^{-8}
	input emb.	205.5	51.2	97.1	2.7	0.2	5.1×10^{-9}
	body	1073.9	0.0	0.6	90.7	8.7	5.5×10^{-6}
Qwen2.5-0.5B	embedding	136.1	**0.0**	82.3	11.6	5.1	2.0×10^{-8}
	body	357.9	0.0	0.3	15.1	83.6	1.8×10^{-5}

A.14 POST-HOC EDITS IN DETAIL

Restoration, and its timing. We evaluate post-hoc editing under two formulations. First, subtracting the shared component of absent-row drift by ratio α yields maximum recoveries of merely $+0.7\%$, $+2.5\%$, and $+4.6\%$ across three replay-free settings; in all cases, the optimal setting corresponds to partial removal ($\alpha \in [0.25, 0.5]$), whereas full removal ($\alpha = 1$) degrades target domain learning by -7.7% to -133.6% (Table 7). Second, resetting entire embedding rows to base values achieves at most $+3.5\%$ recovery under 1% replay (with tied embeddings, $\alpha \approx 0.2$) and 0% for untied embeddings, with complete reset strictly degrading performance. Temporal dynamics vary across replay regimes. Under zero replay, maximum recovery remains flat between $+1.7\%$ and $+2.5\%$ across all six checkpoints, while energy along the shared direction saturates at 66% within just 20M tokens. Under 1% replay, a different arm answers the timing question: restoring the never-fed rows in full ($\alpha = 1$) at each of six saved points leaves recovery negative throughout and monotonically worse, from -12.3% at 6.7M tokens to -68.7% at 40M with no inversion (Figure A4). That trajectory is the complete reset, not the partial best-α arm above, which is why its numbers are negative where the best-α recoveries are positive. Neither regime exhibits an early window for viable repair.

Why it fails, and the ordering of the damage. Component ablation in Appendix A.6 exposes the root cause. Because ablating the residual amplifies forgetting by factors of $1.6\times$, $1.7\times$, and $2.4\times$

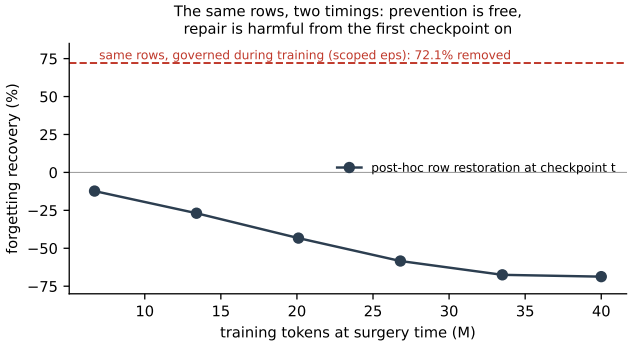

Figure A4: The same embeddings at two moments, Qwen / Korean under 1% replay. Governed during training, 72.1% of the forgetting is removed at no cost (dashed, the ϵ arm of Table A9); restored after training, the edit is a loss from the first checkpoint on and degrades monotonically thereafter.

across the three settings, compensatory adjustments that sustain legacy capabilities reside within the residual of embedding displacement. We also asked whether the body, rather than the embedding residual, is the main site of that compensation, and the answer is that we could not confirm it either way. The alignment between the shift in legacy-domain mean hidden states and the shared drift axis is $\cos = +0.248$ under 1% replay, eight times the random expectation of 0.03 but below the 0.3 we registered in advance. We therefore do not claim the body is uninvolved; we claim only that the embedding residual is where the positive evidence is, since ablating it is what multiplies forgetting while ablating the shared direction does not.

The account makes a prediction that can fail: post-hoc damage should scale with how selectively an edit breaks the relationship between drift and compensation, not with how much weight it moves. The ordering holds (Table 7). Row-level restoration, which replaces some rows entirely and leaves their neighbours untouched, is the most destructive. Ablating the shared subspace, which cuts one direction out of every row, is intermediate. WiSE-FT interpolation (Wortsman et al., 2022), which scales every parameter by the same factor and so moves drift and compensation together, is the one edit that traces a non-catastrophic frontier. Interpolating from the retrained checkpoint back toward base at $\alpha = 0.95, 0.9, 0.8, 0.7, 0.5$ recovers 11.4%, 22.1%, 41.0%, 57.0% and 81.0% of forgetting on Qwen/Korean while giving up 1.0% to 29.9% of target learning, and 14.0% through 100.5% on Pythia-1.4B/math for $+0.5\%$ down to -24.3%. Every point on that frontier still pays, and the free point of output-scoped ϵ sits above it. Two limits keep these numbers out of the main comparison: both sweeps predate the replay-free campaign, carrying 1% replay at a single checkpoint per cell, and the 1.4B cell injects math rather than Korean, so its small baseline (0.0364 nats) is what lets the last step read above 100%. That last case is not a counterexample to this section, it is the case the account predicts.