# EXACT, REVERSIBLE, AND VERIFIABLE LIFECYCLE CONTROL FOR ROUTED PARAMETRIC MEMORY

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Deleting, correcting, or reverting what a deployed language model knows is today an optimisation problem on entangled weights. Unlearning objectives and model editors move parameters until a behavioural probe is satisfied. The result is approximate, depends on the order of operations, and can be verified by no one but the operator. The problem disappears when knowledge is *placed* rather than *optimised*. Block-Rollback stores injected knowledge in the content-routed memory blocks of a hierarchical-memory language model whose backbone, the *anchor*, stays frozen. Every block is a deterministic function of the documents it holds and a public seed. Insert, edit, delete, and revert therefore reduce to rebuilding one block from an append-only log, and each operation lands element-wise on the counterfactual model: the one that never saw the deleted fact, only ever saw the corrected fact, or stopped at the reverted step. We test this on five public anchors from GPT-2-XL to Llama-2-7B and on two anchors pretrained from scratch with the memory pathway, across the KnowEdit, TOFU, and MUSE benchmarks. More than 17,000 locality checks find no violation, and every rebuilt block matches its counterfactual exactly. On the same base, gradient editors halve their locality within ten edits and cannot revert, and optimisation-based unlearners remain separable from the counterfactual by membership, extraction, and jailbreak probes. Block-Rollback matches it on every probe, including after int8 and NF4 quantisation. A third-party certificate rejects 10,000 forged deletions at the cost of rebuilding one block. Per-operation cost tracks the touched block rather than the corpus or the record's age. A read-only routing gate keeps general capability at the anchor's values through 300 edits.

## 1 INTRODUCTION

A deployed language model accumulates obligations to change what it knows. A person invokes the right to be forgotten, a fact goes stale, or a patch turns out to be wrong and must be undone. Current tools treat these operations as optimisation on entangled weights. Unlearning objectives push the model away from the forget set (Zhang et al., 2024b; Fan et al., 2025; Li et al., 2024). Editors solve for a rank-one or side-memory update that satisfies a behavioural probe (Meng et al., 2022; 2023; Wang et al., 2024a; Hartvigsen et al., 2023). The result is *approximate*: the model retrained without the fact is a reference to be approached, never reached, and for multi-stage training no local update can reach it at all (Yu et al., 2025). It is *path-dependent*: each operation perturbs weights shared with every other fact, so *locality*, the share of unrelated facts left unchanged, decays along an edit stream, and the same edits in another order give another model. It is *unverifiable*: a party who does not trust the operator has nothing to check but the operator's own probes.

The exact alternative has been to retrain. SISA shards the training data and retrains a shard (Bourtoule et al., 2021). Deterministic replay records a training run and replays it without the deleted example (X, 2025). Both are exact, and both pay in the wrong currency. Random shards scatter related knowledge, so a query must consult every shard. Replay recomputes every step after the deleted one, so the oldest record is the most expensive to forget. Neither edits or reverts, and neither is cheaper to verify than to redo.



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.

We start from a different diagnosis: the operations are hard because knowledge was *placed* where every other fact can reach it. Hierarchical-memory language models (Pouransari et al., 2026) already separate a frozen backbone, the *anchor*, which holds common knowledge, from content-routed memory *blocks* that hold the long tail. A frozen clustering of the content sends each document and each query to one block. Once injected knowledge lives only in such blocks, exactness becomes a property of the storage layout. Block-Rollback fixes three things. First, a frozen content router, so a document's block never depends on other documents. Second, a deterministic trainer that rebuilds a block from its current documents and a public seed, so a block is a pure function of what it holds. Third, an append-only log of insert, edit, delete, and revert operations that defines every block's document set. Applying an operation means rebuilding, or *re-materialising*, the one block whose document set changed. A deleted fact is not suppressed; its block is rebuilt as if the fact had never been written. A revert is the state of the truncated log, not an inverse update.

Purity buys what optimisation cannot. Throughout the paper the reference point is the *gold* model, trained independently on the counterfactual history, and the primary measurement is the parameter distance $\mathrm{pd}$ to it; $\mathrm{pd} = 0$ means element-wise identity. A deletion is *verifiable*: the operator publishes a seed commitment, the surviving documents, and hashes of the block and anchor, and any party rebuilds the one block and compares, at a $1/K$ fraction of a retrain for $K$ blocks. Cost is *local*: a block's rebuild time depends on its own documents, not on the corpus or on where the fact sits in history. Privacy is *inherited*: the post-deletion memory is the gold model, so every membership, extraction, and jailbreak probe returns the gold's value. Capability is *constant*: the anchor is frozen, and a read-only margin gate sends prompts that belong to no block back to it. Section 4 tests each of these claims with every baseline run on the same base. Our contributions are threefold:

- **Exact lifecycle control by placement.** A construction on routed parametric memory under which insert, edit, delete, and revert land exactly on the gold model, and the state depends on the log's content rather than its order. Across five model families and three editing benchmarks, 17,430 locality checks find no violation, one op-log in 16 orders gives 16 identical parameter sets, and a 1,000-operation stream replays to $\mathrm{pd} = 0$ (Section 4.2).
- **Guarantees no editor or unlearner provides, measured on the same base.** On Llama-2-7B-chat, ROME and MEMIT lose half their locality within ten edits and no editor can revert. Four unlearning objectives applied to the same memory stay 5–12 from the gold and are distinguishable from it by privacy probes. Block-Rollback matches the gold on every probe, quantisation included, and holds MMLU, SST-2, and perplexity at the anchor's values through 300 edits (Sections 4.3–4.4).
- **Verification at the cost of one block; cost that ignores the corpus.** A forgery-resistant certificate rejects 10,000 forged deletions with zero false accepts at a median $3.7\times$ below retraining. Per-operation wall time tracks the touched block ($r = 0.9999$) rather than the corpus, and deleting the oldest record costs the same as the newest, where training-state replay costs $8.1\times$ more (Section 4.5).

## 2 Related Work

**Exact and approximate unlearning.** Exact unlearning retrains: from scratch (Cao & Yang, 2015), per shard in SISA (Bourtoule et al., 2021), per merged fine-tuning model (Kuo et al., 2025), or by replaying a recorded training run without the deleted example (X, 2025). These give the retain-only model, but they shard at random, so related knowledge is scattered and must be aggregated at read time, and their cost scales with the shard or with the suffix of the run. Certified unlearning bounds the distance to retraining with noise (Guo et al., 2020; Pandey et al., 2026). Approximate LLM unlearning optimises away the forget set with gradient ascent, preference objectives, or representation misdirection (Zhang et al., 2024b; Fan et al., 2025; Li et al., 2024) and is scored on TOFU and MUSE (Maini et al., 2024; Shi et al., 2025). The forgotten knowledge is recoverable by relearning and quantisation (Deeb & Roger, 2024; Zhang et al., 2025), retrain equivalence is unreachable for local updates on multi-stage training (Yu et al., 2025), and verifiable unlearning asks for an artefact a third party can check (Eisenhofer et al., 2025). Block-Rollback retrains one block, routes by content rather than at random, and carries a certificate an auditor checks by rebuilding that block.

**Model editing.** Locate-then-edit methods write rank-one or batched updates into feed-forward weights (Meng et al., 2022; 2023; Mitchell et al., 2022a). Side-memory methods route edited inputs to an adapter or codebook (Wang et al., 2024a; Hartvigsen et al., 2023; Mitchell et al., 2022b). KnowEdit, RippleEdits, and MQuAKE standardise reliability, generalization, locality, and ripple evaluation (Zhang et al., 2024a; Cohen et al., 2024; Zhong et al., 2023). All of these optimise a behavioural criterion on shared parameters. Locality is therefore a score that decays along a stream, and no edit can be undone without an externally stored copy of the weights. Block-Rollback re-materialises a block, so locality is exact, edits compose in any order, and every state in the log is recoverable.

**Memory-augmented language models.** Product-key and hierarchical memory layers put long-tail knowledge into large sparse parameter banks fetched by content (Berges et al., 2024; Pouransari et al., 2026). Pouransari et al. (2026) note that such blocks could be removed or edited but leave the mechanism and its guarantees open. Unlearning by design in vision removes an instance's key from an exemplar memory (Laguna et al., 2026), and exact deletion from in-context and recurrent state subtracts or replays the record's contribution (Ramesh, 2026). We give the parametric-memory case its lifecycle algebra, its certificate, and its same-base evidence.

# 3 METHOD

## 3.1 SETUP: A FROZEN ANCHOR WITH CONTENT-ROUTED MEMORY BLOCKS

The model is a frozen decoder-only *anchor* $f_\theta$ augmented with a hierarchical parametric memory in the sense of Pouransari et al. (2026). At every transformer layer $i$, $L$ memory levels each hold $K_\ell$ *blocks*. A block $(\ell, c)$ is one row of three matrices $W_1^{(i,\ell)}, W_2^{(i,\ell)} \in \mathbb{R}^{K_\ell \times d \times w}$ and $W_3^{(i,\ell)} \in \mathbb{R}^{K_\ell \times w \times d}$ of width $w$, with $w \in [16, 256]$ and $w = 64$ for every 7B run. A forward pass carries one routed block per level, $\mathbf{c} = (c_1, \ldots, c_L)$, and adds a gated memory branch to the layer's feed-forward output on hidden state $h$:

$$\mathrm{FFN}_{\mathbf{c}}^{(i)}(h) \;=\; \mathrm{FFN}_{\theta}^{(i)}(h) \;+\; \sum_{\ell=1}^{L} \big(\sigma\big(W_1^{(i,\ell)}[c_\ell]^\top h\big) \odot W_2^{(i,\ell)}[c_\ell]^\top h\big)\, W_3^{(i,\ell)}[c_\ell], \tag{1}$$

where $W[c]$ selects row $c$. Only the memory rows are trainable. The anchor parameters $\theta$ are frozen and never enter an optimiser. With $\mathbf{c} = \varnothing$ the branch is absent and the network is the bare anchor; this is the read path for prompts the memory should not touch (Section 3.4). Every deletable piece of knowledge lives in these rows and nowhere else.

A *content router* $R$ decides which block a document trains and which block a query reads. It is fixed before any knowledge is injected and consists of a frozen text embedder $E$ (MiniLM, unit-normalised outputs) and, per level, a frozen centroid set $C_\ell \in \mathbb{R}^{K_\ell \times e}$. The centroids are fitted once by $k$-means on retain or public text and never refitted:

$$R_\ell(x) \;=\; \arg\min_{c \in [K_\ell]} \|E(x) - C_\ell[c]\|_2, \qquad R(x) \;=\; \big(R_1(x), \ldots, R_L(x)\big). \tag{2}$$

$R$ depends on the content of $x$ alone. It does not depend on the order in which documents arrive, on which other documents exist, or on which documents are later deleted. Every guarantee below rests on this property, and it is why the centroids are frozen rather than learned or refitted. A document's route is computed once at injection and stored with it, so later edits to its text cannot move it to another block.

## 3.2 BLOCKS ARE PURE FUNCTIONS OF THEIR DOCUMENTS

A block is never updated in place. Whenever the document set $D_c$ assigned to block $c$ changes, the block is *re-materialised* by a deterministic trainer $T$ from a seed that is a public function of the block's identity:

$$\theta_c \;=\; T\big(D_c,\, s_c\big), \qquad s_c \;=\; \mathrm{BLAKE2b}\big(s_0 \,\|\, \ell \,\|\, c\big). \tag{3}$$

The trainer $T$ re-initialises row $c$ from $s_c$ (truncated normal for $W_1, W_2$, a small warm start for $W_3$) and builds a fresh AdamW optimiser without weight decay over the memory parameters. It then

presents the documents of $D_c$ in an order derived from $s_c$, with the block's level as the only active level, and minimises the answer-token cross-entropy of (1) for a fixed number of epochs.

Three facts about (1) make $T$ local. Row selection is an index gather, so the backward pass scatters gradient only into row $c$, and every other row receives exactly zero. The anchor is frozen, so it receives none. The optimiser is created fresh for each call, so no moment estimate carries information from another block or from an earlier version of this block. (An optimiser that owns state for row $c$ alone is byte-identical and $K\times$ smaller; Appendix G.) Consequently

$$\theta_c \text{ depends on } (D_c, s_c) \text{ only}, \qquad \theta_{c'} \text{ is unchanged for every } c' \neq c, \qquad \theta \text{ is unchanged}, \tag{4}$$

and two calls of $T$ with the same $(D_c, s_c)$ on the same deterministic hardware return the same bytes. We measure this against a *gold* block, trained independently from the same inputs, as the parameter distance $\mathrm{pd} = \max_j |\theta_{c,j} - \theta^{\mathrm{gold}}_{c,j}|$. A value of $\mathrm{pd} = 0$ means element-wise identity, and $\mathrm{pd} = 0$ throughout the paper is the empirical face of (4).

### 3.3 Lifecycle operators and their normal form

Knowledge enters and leaves the memory through four operators on an append-only *op-log* $\mathcal{L} = (o_1, \ldots, o_t)$. Its entries are $\mathrm{insert}(u, x)$, $\mathrm{edit}(u, x')$, $\mathrm{delete}(u)$, and $\mathrm{revert}(t')$ for a fact identifier $u$ and text $x$. The log determines each block's document set by a fold. The last write to $u$ wins, a delete removes $u$, and $u$ contributes to the block it was routed to at its first insertion:

$$D_c(\mathcal{L}) \;=\; \big\{\, x_u \;:\; u \text{ with } R(u) = c,\ \text{last op on } u \text{ in } \mathcal{L} \text{ is } \mathrm{insert}(u, x_u) \text{ or } \mathrm{edit}(u, x_u) \,\big\}. \tag{5}$$

The materialised state is then $S(\mathcal{L}) = \{T(D_c(\mathcal{L}), s_c)\}_c$. A block whose document set is empty is left at its seed initialisation. Applying an operator means appending it to $\mathcal{L}$ and re-materialising the blocks whose $D_c$ changed; we call these the *affected set* $\mathcal{B}^*$. For a single fact, $|\mathcal{B}^*| = 1$. Because $S$ is defined through (5) rather than through the sequence of parameter updates, three identities hold by construction:

$$\begin{aligned} S(\mathcal{L}{\cdot}\mathrm{delete}(u)) &= S(\mathcal{L} \setminus u), & S(\mathcal{L}{\cdot}\mathrm{revert}(t')) &= S(\mathcal{L}_{\le t'}), \\ S(\mathcal{L}{\cdot}\mathrm{edit}(u, x')) &= S\big((\mathcal{L} \setminus u){\cdot}\mathrm{insert}(u, x')\big), \end{aligned} \tag{6}$$

where $\mathcal{L} \setminus u$ is the history in which $u$ was never written. A deletion therefore lands on the model that never saw the fact, an edit on the model that only ever saw the corrected fact, and a revert on the exact historical state. None of the three refers to the pre-operation weights, and Section 4 measures exactly these identities.

The same fold makes $S$ insensitive to the order of operations on distinct facts. Any two logs with the same per-fact last write and the same delete set materialise identical bytes, and a block untouched by an operation is untouched byte for byte. Wall-clock cost follows the same locality:

$$\mathrm{cost}(o) \;=\; \sum_{c \in \mathcal{B}^*(o)} E \cdot |D_c| \;=\; E \cdot |\mathcal{B}^*|\, w_{\mathcal{B}^*}, \tag{7}$$

for $E$ epochs and mean occupancy $w_{\mathcal{B}^*}$ of the touched blocks. The cost does not depend on the corpus size $N$ or on where the fact sits in the history. Multi-hop compositions are handled by the same algebra on a second memory level whose blocks are keyed by the base facts they combine. Deleting a base fact re-materialises every composition that depends on it (Appendix I).

### 3.4 Verification and the deployed read path

Purity (4) turns a deletion into something a third party can check without the operator's cooperation. Before any operation on block $c$ the operator publishes a seed commitment. After $\mathrm{delete}(u)$ it publishes a certificate

$$\mathrm{cert} \;=\; \Big(c,\ \underbrace{H(s_c \| c \| \mathrm{nonce})}_{\text{seed commitment}},\ \underbrace{H(D_c(\mathcal{L}))}_{\text{survivors}},\ \underbrace{H(\theta_c)}_{\text{block}},\ \underbrace{H(\theta)}_{\text{anchor}}\Big), \tag{8}$$

where $H$ is SHA-256 over a canonical serialisation. The survivor set is hashed as an identifier-sorted set with a per-document content hash, and the block is hashed both over its raw bytes and

over its fp32 values. A verifier holds the public log, the surviving documents, and its own copy of the anchor. It checks four things in order: that the commitment predates the deletion and matches the revealed seed; that the deleted document is absent from the committed survivor set; that its anchor digest matches; and finally that re-materialising $T(D_c(\mathcal{L}), s_c)$ reproduces $H(\theta_c)$. The audit costs one block, a $1/K$ fraction of retraining the memory, and the hash checks run first, so most forgeries are rejected before any training. A forger who controls the log cannot choose a convenient seed after seeing the weights, cannot reorder or edit the survivors without changing the set hash, and cannot substitute an anchor.

The deployed read path routes every prompt through $R$. A prompt that belongs to no injected block would otherwise read whichever block is nearest. A read-only gate turns the router's distances into a confidence over blocks and abstains to the anchor when that confidence is low:

$$p(x) = \mathrm{softmax}\big(\tfrac{1}{T_g}\langle E(x), C_\ell\rangle\big), \quad \mathbf{c}(x) = \begin{cases} R(x) & p_{(1)}(x) \geq \tau \text{ and } p_{(1)}(x) - p_{(2)}(x) \geq \delta, \\ \varnothing & \text{otherwise.} \end{cases} \tag{9}$$

The thresholds $(\tau, \delta)$ are chosen once on a held-out split of injected facts and out-of-domain prompts and then frozen. The gate reads centroid distances and writes nothing. It only chooses which of two already exact forward paths answers a prompt, and it leaves every statement in Sections 3.2–3.3 untouched.

All results are reported under one determinism regime. With CPU/fp32 or deterministic GPU kernels, $T$ is byte-reproducible. Under bf16 with default kernels, every result in the paper is value-identical after fp32 export; the one configuration that is not, together with its repair, is isolated in Appendix D.

# 4 Experiments

The evidence below comes from seven anchors, six benchmarks, streams of up to 1,000 operations, and baselines that all run on the same base as Block-Rollback.

## 4.1 Evaluation protocol

**Substrates.** The public anchors are GPT-2-XL, Qwen-2.5-0.5B/1.5B/7B-Instruct, and Llama-2-7B (Radford et al., 2019; Qwen Team, 2024; Touvron et al., 2023). Native-124M and Native-257M are Llama-architecture anchors pretrained from scratch on 1.5B and 7B FineWeb tokens *jointly* with the hierarchical-memory pathway of Pouransari et al. (2026). Every anchor is frozen. Knowledge enters only the memory blocks, which a frozen MiniLM router (Reimers & Gurevych, 2019) assigns by content into $K \in [8, 64]$ blocks.

**Benchmarks and baselines.** Editing uses KnowEdit WikiRecent, ZsRE, and CounterFact (Zhang et al., 2024a). Deletion uses TOFU forget01/05/10 (Maini et al., 2024) and MUSE News and Books (Shi et al., 2025). Capability uses MMLU, SST-2, and WikiText-103 (Hendrycks et al., 2021; Socher et al., 2013; Merity et al., 2017). The editors ROME, MEMIT, WISE, and GRACE run through EasyEdit (Meng et al., 2022; 2023; Wang et al., 2024a; Hartvigsen et al., 2023; Wang et al., 2024b) on the *same* Llama-2-7B-chat base as Block-Rollback. The unlearners NPO, SimNPO, gradient ascent, RMU, and a gradient weight editor (Zhang et al., 2024b; Fan et al., 2025; Li et al., 2024) optimise the *same* memory parameters that Block-Rollback re-materialises. The exact peers are SISA-style random sharding (Bourtoule et al., 2021) and deterministic tail replay of the training history (X, 2025). A retrieval store marks the non-parametric boundary (Appendix J).

**Metrics.** The gold model for an operation is trained independently on the counterfactual history: the survivors, the corrected fact, or the truncated op-log. Nothing is copied from the pre-operation model. The parameter distance $\mathrm{pd}$ of Section 3.2, taken over the affected block, is the primary quantity. TOFU forget quality (FQ) is the KS $p$-value between the candidate's and the gold's truth-ratio distributions; FQ $= 1$ is a corollary of $\mathrm{pd} = 0$.

Table 1: **Exactness holds across five model families, three editing benchmarks, and every operation order.** **(a)** Content-routed KnowEdit editing with up to 10 facts per block. pd Affected compares the re-materialised block with its gold; pd Replay compares the replayed op-log with the pre-operation state. **(b)** Order permutations of one op-log, certified lifecycle streams, and a forget-ratio sweep.

**(a) Content-routed editing: every affected block equals its counterfactual**

| | | | | Behaviour | | Locality | | pd | |
|---|---|---|---|---|---|---|---|---|---|
| Model | Benchmark | Facts/Blocks | Max/blk | Rel. | Gen. | Checks | Viol. | Affected | Replay |
| GPT-2-XL | WikiRecent | 60/15 | 8 | 1.00 | 1.00 | 661 | 0 | 0 | 0 |
| | ZsRE | 60/15 | 7 | 1.00 | 1.00 | 662 | 0 | 0 | 0 |
| Qwen-2.5-0.5B | WikiRecent | 100/25 | 10 | 1.00 | 1.00 | 1896 | 0 | 0 | 0 |
| | ZsRE | 100/25 | 10 | 1.00 | 1.00 | 1905 | 0 | 0 | 0 |
| | CounterFact | 100/25 | 8 | 0.99 | 0.99 | 1920 | 0 | 0 | 0 |
| | WikiRecent | 200/64 | 10 | 1.00 | 1.00 | 5857 | 0 | 0 | 0 |
| Qwen-2.5-1.5B | WikiRecent | 60/15 | 8 | 1.00 | 1.00 | 661 | 0 | 0 | 0 |
| Qwen-2.5-7B | WikiRecent | 40/10 | 9 | 1.00 | 1.00 | 358 | 0 | 0 | 0 |
| Llama-2-7B | WikiRecent | 80/20 | 9 | 1.00 | 1.00 | 1181 | 0 | 0 | 0 |
| | ZsRE | 80/20 | 8 | 1.00 | 1.00 | 1192 | 0 | 0 | 0 |
| | CounterFact | 40/10 | 7 | 1.00 | 1.00 | 351 | 0 | 0 | 0 |
| **All routed runs (14)** | | | | | | **17,430** | **0** | **0** | **0** |

**(b) The state does not depend on operation order or stream length**

| Setting | Model | Ops | Evidence | max pd |
|---|---|---|---|---|
| Block-Rollback, 16 orderings of one op-log | Qwen-2.5-0.5B | 25 | 16 orderings | 0 |
| Warm-start additive editor, same orderings | Qwen-2.5-0.5B | 25 | 16 orderings | $5.7 \times 10^{-7}$ |
| Lifecycle stream, certificate every 50 ops | Qwen-2.5-0.5B | 1000 | 21/21 verified | 0 |
| Lifecycle stream, certificate every 5 ops | Llama-2-7B | 80 | 17/17 verified | 0 |
| Forget-ratio sweep 1–90% of a 120-fact pool | Llama-2-7B | 7 ratios | FQ $= 1.00$ at every ratio | 0 |

### 4.2 Every edit and deletion lands on its counterfactual state

Table 1a covers 11 model–benchmark pairs. In every pair the affected block is element-wise identical to its gold, and the replayed op-log reproduces the state exactly. Across all 14 routed runs, 17,430 locality checks find 0 violations. The routing is load-bearing: the 200-fact Qwen run packs 64 blocks with up to 10 facts each, and deleting one fact from a shared block leaves its survivors recallable while the deleted fact is gone (its NLL rises from 1.7 to 15.5).

Table 1b shows that the state is a function of the op-log's *content*, not of its order or length. One 25-operation log applied in 16 distinct orders yields 16 identical parameter sets, whereas a warm-start additive editor on the same orders diverges by $5.7 \times 10^{-7}$. A 1,000-operation stream passes every interleaved certificate check and replays to $\mathrm{pd} = 0$, and so does the 80-operation Llama-2-7B stream. Deleting between 1% and 90% of a 120-fact pool on Llama-2-7B leaves FQ at 1.00 and $\mathrm{pd}$ at 0 at every ratio, while the number of retrained blocks grows from 1 to 15 (Figure 1a). The amount deleted changes cost, never exactness.

### 4.3 Editors on the same base cannot match, revert, or hold capability

Gradient editors overwrite `down_proj` rows in place. On the shared base, ROME and MEMIT start at locality 0.44 and 0.41 after a single edit and fall to 0.22 and 0.23 after ten (Table 2a). Half of the neighbouring facts change within a ten-edit stream. WISE, the side-memory editor closest to ours, keeps locality but loses reliability, from 0.95 to 0.80. None of the three can restore the pre-edit parameters after a stream, because none records what each edit did to which parameter. Block-Rollback re-materialises the target's block on the corrected fact. The edit is exact, with no locality violation in 232 checks. Reliability and generalization stay at 1.00 through the stream, and replaying the op-log without the edit restores the previous state to $\mathrm{pd} = 0$.

Capability is the second axis (Table 2b). The frozen anchor's MMLU, SST-2, and perplexity are constant by construction. A memory that answers *every* prompt through its nearest block does not inherit this. After 100 edits the ungated path loses more than 40% of its MMLU accuracy and half of

Table 2: **On the same Llama-2-7B-chat base, only Block-Rollback has exact locality, exact revert, and the anchor's general capability.** **(a)** EasyEdit protocol, 20 WikiRecent facts, re-evaluated after ten further edits; $\max|\Delta\theta|$ is the largest change to a base weight. **(b)** 300 KnowEdit facts in 64 blocks; the gated read path abstains to the anchor on low router margin (out-of-domain abstention 1.00/1.00/1.00); MMLU $n$=72, SST-2 $n$=80, WikiText-103 $n$=60.

**(a) Same base (Llama-2-7B-chat), same 20 WikiRecent facts, same 10-edit stream**

| | Isolated edit | | | After 10 sequential edits | | | |
|---|---|---|---|---|---|---|---|
| Editor | Rel. | Gen. | Loc. | Rel. | Loc. | $\max\|\Delta\theta\|$ | Exact revert |
| ROME | 0.99 | 0.65 | 0.44 | 0.99→0.98 | 0.44→0.22 | 0.166 | no |
| MEMIT | 0.98 | 0.78 | 0.41 | 0.98→0.99 | 0.41→0.23 | 0.031 | no |
| WISE | 0.95 | 0.82 | 1.00 | 0.95→0.80 | 1.00→0.93 | side memory | no |
| **Block-Rollback** | **1.00** | **1.00** | **exact** (0/232) | **1.00→1.00** | **exact→exact** | **0** | **yes** |

**(b) General capability of Llama-2-7B-chat under a 300-edit stream**

| | 0 edits | | | 100 edits | | | 300 edits | | |
|---|---|---|---|---|---|---|---|---|---|
| Read path | MMLU | SST-2 | PPL | MMLU | SST-2 | PPL | MMLU | SST-2 | PPL |
| Anchor (frozen) | 0.319 | 0.950 | 13.8 | 0.319 | 0.950 | 13.8 | 0.319 | 0.950 | 13.8 |
| Memory, ungated | 0.319 | 0.950 | 13.8 | 0.181 | 0.438 | $3.2 \times 10^6$ | 0.264 | 0.512 | $1.1 \times 10^6$ |
| **Memory, gated (deployed)** | **0.319** | **0.950** | **13.8** | **0.319** | **0.950** | **13.8** | **0.319** | **0.950** | **13.8** |

its SST-2 accuracy, and its WikiText perplexity leaves the measurable range, because out-of-domain prompts land in fact blocks never trained for them. A read-only margin gate that abstains to the anchor when the router is unsure removes the effect. The gated path matches the anchor to the last digit on all three measures at 0, 100, and 300 edits. It abstains on every out-of-domain prompt and still recalls 0.77 / 0.79 of the injected facts at 100/300 edits. The gate touches no weight and no router state, so no $\mathrm{pd} = 0$ result depends on it.

### 4.4 After deletion, the model is the retain-only model

Table 3a is the TOFU protocol read literally. After deleting the forget01 author from a Llama-2-7B memory, every statistic of Block-Rollback equals the gold's, so FQ is 1.00 and $\mathrm{pd}$ is 0. NPO on the same memory under-forgets: its forget probability is 0.081 against the gold's 0.005. SimNPO over-forgets and drives the truth ratio to 0.003. Both end 12.45 from the gold in parameter space. Table 3b widens the comparison to six unlearners on forget10. Every optimisation-based method sits 5.1–5.6 from the gold and is distinguishable from it by at least one probe. Gradient ascent and the weight editor destroy the retain set. The retain-NLL values below the gold's (NPO, SimNPO) are not utility gains, since those methods never retrain the retain set.

Privacy is inherited rather than argued. The post-deletion memory *is* the gold's, so every attack returns the same value against Block-Rollback as against the gold (Table 3c). Every membership, quantised-membership, paraphrase, extraction, and jailbreak probe in the table agrees to three decimals, and fine-tuning on two deleted facts recovers nothing of the other 28. MUSE carries the result to real News and Books text and to repeated deletion on Llama-2-7B. Verbatim memorisation falls to the gold's value, the Min-K membership AUC equals the gold's so PrivLeak is 0.0, and $\mathrm{pd}$ is 0 after each of three deletion rounds on both corpora (Appendix D).

### 4.5 Deletion is verifiable at the cost of one block, and cost ignores the corpus

A certificate names one block, its seed, its survivor set, and the anchor digest. Table 4a shows the four layers doing distinct work. Tampering with the survivors moves the re-materialised block by 4.0–7.2 and fails the closure check. Recommitting the hashes to hide such a change fails the block hash. Changing the seed or the anchor fails the corresponding commitment. Two forgeries leave the block numerically correct and are still rejected, so $\mathrm{pd}$ alone would not suffice. Across 10,000 randomised forgeries from 18 classes the false-accept rate is 0. Measured full-retrain-to-audit wall ratios are 2.5–8.3 (median 3.7) for $K$ from 2 to 8, in line with the $1/K$ cost of rebuilding one block.

Table 3: **Deletion by re-materialisation coincides with the retain-only gold on every statistic and every attack; optimisation-based unlearners on the same memory do not.** **(a)** TOFU forget01. **(b)** TOFU forget10; LiRA uses 16 shadow models; Δ columns are changes relative to gold in the forget/hold-out loss gap and in the jailbreak leakage score. **(c)** Attack battery after deleting 30 of 80 facts; int8 and NF4 via bitsandbytes.

**(a) TOFU forget01 on Llama-2-7B-chat: every statistic of Block-Rollback is the gold's**

| | Forget set | | | Retain set | | |
|---|---|---|---|---|---|---|
| Method | Prob. | Truth ratio | FQ | Prob. | Truth ratio | pd |
| Retain-only gold | 0.005 | 1.086 | 1.00 | 0.729 | 0.428 | 0 |
| **Block-Rollback** | **0.005** | **1.086** | **1.00** | **0.729** | **0.428** | **0** |
| NPO on memory | 0.081 | 0.410 | $1.9 \times 10^{-6}$ | 0.836 | 0.800 | 12.45 |
| SimNPO on memory | <0.001 | 0.003 | $1.9 \times 10^{-23}$ | 0.852 | 1.428 | 12.45 |

**(b) Six unlearners on the same memory substrate (Llama-2-7B, TOFU forget10)**; Δ is relative to gold

| Method | pd | Forget NLL | Retain NLL | LiRA AUC | Min-K AUC | Extraction Δ | Jailbreak Δ |
|---|---|---|---|---|---|---|---|
| Retain-only gold | 0 | 6.38 | 3.50 | 0.438 | 1.000 | 0 | 0 |
| **Block-Rollback** | **0** | **6.38** | **3.50** | **0.438** | **1.000** | **0** | **0** |
| NPO | 5.37 | 10.90 | 2.02 | 0.542 | 0.667 | −0.99 | −4.4 |
| SimNPO | 5.25 | 59.12 | 2.07 | 0.500 | 0.000 | −38.59 | −49.6 |
| Gradient ascent | 5.19 | 142.38 | 24.25 | 0.500 | 0.000 | −15.91 | −135.9 |
| RMU | 5.61 | 17.47 | 6.79 | 0.583 | 0.500 | +2.73 | −11.1 |
| Weight editor | 5.14 | 139.88 | 33.77 | 0.521 | 0.000 | −10.78 | −133.5 |

**(c) Full attack battery on Llama-2-7B-chat** (80-fact pool, 30 deleted, 80 held-out; pd = 0)

| Probe | Block-Rollback | Gold | Δ |
|---|---|---|---|
| Membership AUC (loss / Min-K / Min-K++) | 0.254 / 0.743 / 0.541 | 0.254 / 0.743 / 0.541 | 0 |
| Membership AUC after int8 / NF4 quantisation (Min-K++) | 0.543 / 0.535 | 0.543 / 0.535 | 0 |
| Membership AUC, 30 paraphrased probes | 0.247 / 0.758 / 0.611 | 0.247 / 0.758 / 0.611 | 0 |
| Extraction gap / grad. norm / jailbreak leakage | 4.48 / 1.11 / −4.29 | 4.48 / 1.11 / −4.29 | 0 |
| Relearning recovery of held-out forget facts (NLL drop) | −2.78 | 0.00 | −2.78 |

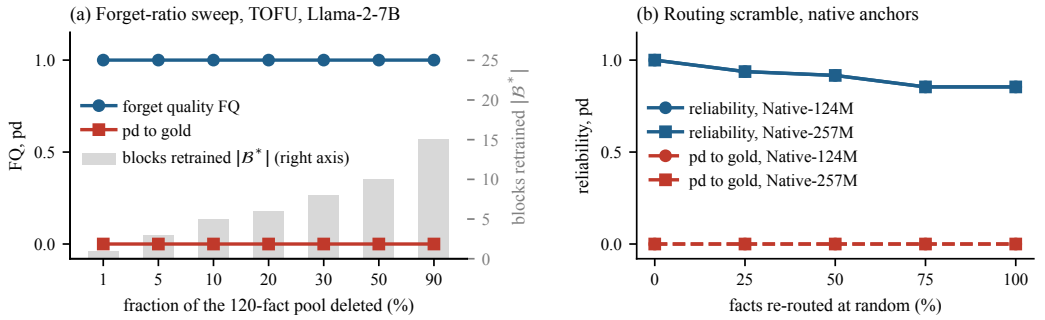


Figure 1: **Exactness is flat where cost and utility move.** **(a)** TOFU forget-ratio sweep on Llama-2-7B: FQ stays at 1 and pd at 0 while the number of retrained blocks rises with the deleted fraction. **(b)** Native anchors, 48 KnowEdit facts in 8 blocks: reliability falls as content routing is replaced by random assignment; pd stays at 0.

The same locality sets the cost of every operation. Over the 1,000-operation stream, wall time correlates with $|\mathcal{B}^*|\,w$ at $r = 0.9999$ and with corpus size at only $r = 0.75$; on Llama-2-7B the two coefficients are 0.999 and 0.34 (Table 4b). Table 4c isolates what separates Block-Rollback from training-state replay (X, 2025). Deleting the first, middle, or last record of a 32-operation history costs Block-Rollback about 21 seconds in every case, whereas tail replay costs 606, 403, and 75 seconds because it recomputes every operation after the deleted one. Both routes reach the same state; replay pays 8.1× more for the oldest record than for the newest, and ours pays the same for both.

Table 4: **An auditor holding only the public log rejects every forged certificate by re-materialising one block, and the cost of an operation is set by that block rather than by the corpus or the record's age.** **(a)** Hand-constructed tampers on a Qwen-2.5-0.5B certificate and 10,000 randomised forgeries from 18 classes; escalated forgeries pass the hash layer and fail re-materialisation (all tampers in Appendix F). **(b)** Insert/edit/delete streams; $|\mathcal{B}^*|\,w$ is blocks touched $\times$ documents per block. **(c)** Deleting the first, middle, and last record of a 32-operation history (Qwen-2.5-0.5B); tail replay restarts from the last checkpoint before the record.

**(a) Every tampered certificate is rejected**

| Tamper | pd of forged block | Rejecting check | Rejected |
|---|---|---|---|
| Alter a surviving document | 4.47 | survivor closure | yes |
| Add a phantom survivor | 7.22 | survivor closure | yes |
| Change the block seed | 4.01 | seed commitment | yes |
| Swap the anchor weights | 3.52 | anchor digest | yes |
| Drop a document, recommit all hashes | 5.10 | block hash | yes |
| Flip one bit of the block hash | 0 | block hash | yes |
| Synthesise the seed after the fact | 0 | seed commitment | yes |
| **10,000 random forgeries, 18 classes** | **–** | **539 escalated** | **10,000/10,000** |

**(b) Per-operation cost follows the touched block, not the corpus**

| | | | Pearson $r$ of wall time with | | | | |
|---|---|---|---|---|---|---|---|
| Stream | Ops | Final $N$ | $\lvert\mathcal{B}^*\rvert\,w$ | $N$ | s per unit | Certificates | Revert pd |
| Qwen-2.5-0.5B, fp32 | 45 | 22 | 0.9903 | 0.697 | 5.30 | 45/45 | 0 |
| Qwen-2.5-0.5B, bf16 | 1000 | 472 | 0.9999 | 0.751 | 2.19 | 21/21 | 0 |
| Llama-2-7B, bf16 | 80 | 44 | 0.9993 | 0.341 | 4.41 | 17/17 | 0 |
| Native-257M, fp32 | 60 | 32 | 0.9993 | 0.524 | 2.22 | 60/60 | 0 |

**(c) Deletion cost is independent of the record's position in history; replay is not**

| | | Block-Rollback | | Tail replay from checkpoint | | |
|---|---|---|---|---|---|---|
| Deleted record | Suffix length | Wall (s) | Work units | Wall (s) | Work units | pd, both routes |
| Early (op 0) | 32 | 20.9 | 45 | 605.8 | 1275 | 0 |
| Mid (op 15) | 17 | 21.5 | 45 | 402.7 | 870 | 0 |
| Late (op 29) | 3 | 21.3 | 45 | 74.8 | 165 | 0 |

### 4.6 Content routing buys reachability at no cost to exactness

Exactness holds for any assignment of facts to blocks: content, oracle $k$-means, random, and one-fact-per-block routing all give $\mathrm{pd} = 0$ and zero locality violations on Qwen-0.5B, Qwen-1.5B, and Llama-2-7B (Appendix H). What routing decides is *utility*. A content router sends a query to the one block that holds its knowledge. Random SISA-style shards scatter related facts: on Qwen-0.5B, one routed read reaches retain NLL 0.40, whereas random shards need an $O(K{=}16)$ aggregation over every shard to reach 0.89. Figure 1b traces the mechanism: as the content assignment is scrambled toward random, reliability falls monotonically from 1.000 to 0.854 while $\mathrm{pd}$ stays at 0 at every point.

The natively pretrained anchors close the loop (Appendix L): a 124M and a 257M anchor trained from scratch *with* the memory pathway pass 60/60 certificates each, revert to $\mathrm{pd} = 0$, and match the gold on every attack. Content routing reaches every injected fact at NLL 0.58/0.60; random assignment loses 12.5% of them. The guarantees belong to the construction, not to a frozen backbone.

## 5 Conclusion

Cast as optimisation on entangled weights, editing and unlearning inherit approximation, path dependence, and unverifiability. Block-Rollback places injected knowledge in content-routed blocks that are pure functions of their documents. Insert, edit, delete, and revert therefore land exactly on the counterfactual model, in any order, at the cost of one block, with a certificate any party can check. Across seven anchors and six benchmarks the construction holds without exception. The contract covers knowledge that enters through the memory; what the frozen anchor learned in pretraining is outside it by design.

## AI USE STATEMENT

We did not use generative AI tools for tasks requiring disclosure, including research ideation, methodological or experimental design, implementation, or interpretation of results. Generative AI tools were used solely to edit author-written manuscript text for grammar, clarity, and concision. All AI-assisted edits were reviewed and revised by the authors, who take full responsibility for the final content of this work

## REFERENCES

Vincent-Pierre Berges, Barlas Oguz, Daniel Haziza, Wen-tau Yih, Luke Zettlemoyer, and Gargi Ghosh. Memory layers at scale. *arXiv preprint arXiv:2412.09764*, 2024.

Lucas Bourtoule, Varun Chandrasekaran, Christopher A Choquette-Choo, et al. Machine unlearning. In *IEEE Symposium on Security and Privacy*, 2021.

Yinzhi Cao and Junfeng Yang. Towards making systems forget with machine unlearning. In *IEEE Symposium on Security and Privacy*, 2015.

Roi Cohen, Eden Biran, Ori Yoran, Amir Globerson, and Mor Geva. Evaluating the ripple effects of knowledge editing in language models. *Transactions of the Association for Computational Linguistics*, 2024.

Aghyad Deeb and Fabien Roger. Do unlearning methods remove information from language model weights? *arXiv preprint arXiv:2410.08827*, 2024.

Thorsten Eisenhofer, Doreen Riepel, Varun Chandrasekaran, Esha Ghosh, Olga Ohrimenko, and Nicolas Papernot. Verifiable and provably secure machine unlearning. In *IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, 2025.

Chongyu Fan, Jiancheng Liu, Licong Lin, et al. Simplicity prevails: Rethinking negative preference optimization for llm unlearning. In *International Conference on Learning Representations*, 2025.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *ICML*, 2020.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. Aging with grace: Lifelong model editing with discrete key-value adaptors. In *NeurIPS*, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, et al. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

Kevin Kuo et al. Exact unlearning of finetuning data via model merging at scale. *arXiv preprint arXiv:2504.04626*, 2025.

Sonia Laguna, Jorge da Silva Gonçalves, Moritz Vandenhirtz, Alain Ryser, Irene Cannistraci, and Julia E Vogt. Rethinking machine unlearning: Models designed to forget via key deletion. *arXiv preprint arXiv:2603.15033*, 2026.

Nathaniel Li, Alexander Pan, Anjali Gopal, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. In *ICML*, 2024.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. Tofu: A task of fictitious unlearning for llms. In *Conference on Language Modeling*, 2024.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. In *NeurIPS*, 2022.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. Mass-editing memory in a transformer. In *International Conference on Learning Representations*, 2023.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. In *International Conference on Learning Representations*, 2022a.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D Manning, and Chelsea Finn. Memory-based model editing at scale. In *ICML*, 2022b.

Aaradhya Pandey et al. Gaussian certified unlearning in high dimensions: A hypothesis testing approach. In *International Conference on Learning Representations*, 2026.

Hadi Pouransari, David Grangier, C Thomas, Michael Kirchhof, and Oncel Tuzel. Pretraining with hierarchical memories: Separating long-tail and common knowledge. In *International Conference on Learning Representations*, 2026.

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI technical report*, 2019.

Vishwajith Ramesh. Subtract or replay? exact deletion from language-model memory. *arXiv preprint arXiv:2607.27539*, 2026.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP*, 2019.

Weijia Shi, Jaechan Lee, Yangsibo Huang, et al. Muse: Machine unlearning six-way evaluation for language models. In *International Conference on Learning Representations*, 2025.

Richard Socher, Alex Perelygin, Jean Wu, et al. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, 2013.

Hugo Touvron, Louis Martin, Kevin Stone, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Peng Wang, Zexi Li, Ningyu Zhang, et al. Wise: Rethinking the knowledge memory for lifelong model editing of large language models. In *NeurIPS*, 2024a.

Peng Wang, Ningyu Zhang, Bozhong Tian, et al. Easyedit: An easy-to-use knowledge editing framework for large language models. In *ACL System Demonstrations*, 2024b.

Abdullah X. Unlearning at scale: State-exact trace-preserving deletion in billion-parameter language models. *arXiv preprint arXiv:2508.12220*, 2025.

Jiatong Yu, Yinghui He, Anirudh Goyal, and Sanjeev Arora. On the impossibility of retrain equivalence in machine unlearning. *arXiv (Cornell University)*, 2025.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, et al. A comprehensive study of knowledge editing for large language models. *arXiv preprint arXiv:2401.01286*, 2024a.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. Negative preference optimization: From catastrophic collapse to effective unlearning. In *Conference on Language Modeling*, 2024b.

Zhiwei Zhang, Fali Wang, Xiaomin Li, et al. Catastrophic failure of llm unlearning via quantization. In *International Conference on Learning Representations*, 2025.

Zexuan Zhong, Zhengxuan Wu, Christopher D Manning, Christopher Potts, and Danqi Chen. Mquake: Assessing knowledge editing in language models via multi-hop questions. In *EMNLP*, 2023.

## A PROTOCOL DETAILS

**Memory blocks and injection.** Each block is one row of the hierarchical-memory FFN of Pouransari et al. (2026) at every layer of the anchor (width 16–256 per experiment, 64 for all 7B headline runs). A block is trained alone: it is re-initialised from its deterministic seed, its output projection is warm-started with standard deviation $0.01$, a fresh AdamW optimiser without weight decay covers only that block's parameters, and the block's documents are presented in a fixed content-sorted order at learning rate $0.05$ for a fixed number of epochs. Nothing outside the block receives a gradient, so a block is a function of (document set, seed) alone; every exactness result follows from this construction.

**Routing and gold models.** The router embeds a document with MiniLM (Reimers & Gurevych, 2019) and assigns it to the nearest of $K$ centroids fitted once on retain or public text and then frozen; a deleted document therefore never moves another document's route. The gold model of an operation is produced by the same trainer from the counterfactual history and is never seeded from the pre-operation state. Locality checks compare every block outside the affected set with its pre-operation copy after every operation.

**Certificate.** A certificate commits, before the operation, to the seed of the affected block, the hash of its surviving document set, the hash of the re-materialised block (raw bytes and fp32 values), and a digest of the frozen anchor weights. The verifier recomputes the closure, re-materialises the one block from the public inputs, and compares all four commitments; a cheap layer checks the hashes first and escalates to re-materialisation only when they pass.

**Determinism.** Under CPU/fp32 execution and under GPU execution with MATH/eager kernels the re-materialised block is byte-identical to the gold's. Under bf16 with default flash-attention kernels every single- and few-round result in the paper is value-identical after fp32 export ($\mathrm{pd} = 0$); the one deviation, and its repair, are in Appendix D.

## B EXACTNESS REPLICATES ACROSS SEEDS, WIDTHS, AND BENCHMARKS

Table 5 lists the one-fact-per-block KnowEdit runs behind the Llama-2-7B rows of Table 2 and the seed replicates on Qwen-2.5-0.5B. Reliability and generalization are 1.00 in every run, no locality check fails, every deleted fact is forgotten, and the op-log replays to $\mathrm{pd} = 0$; the seed changes the reliability NLL (0.34–0.58) and nothing else.

Table 5: Unrouted KnowEdit editing across seeds, block widths, benchmarks, and anchors.

| Model | Benchmark | Seed | Facts/ed./del. | Width | Rel. | Gen. | NLL | Loc. viol. | Forgot | Replay pd |
|---|---|---|---|---|---|---|---|---|---|---|
| Qwen-2.5-0.5B | WikiRecent | 0 | 40/10/5 | 128 | 1.00 | 1.00 | 0.34 | 0/390 | 5/5 | 0 |
| Qwen-2.5-0.5B | WikiRecent | 1 | 40/10/5 | 128 | 1.00 | 1.00 | 0.41 | 0/390 | 5/5 | 0 |
| Qwen-2.5-0.5B | WikiRecent | 2 | 40/10/5 | 128 | 1.00 | 1.00 | 0.47 | 0/390 | 5/5 | 0 |
| Qwen-2.5-0.5B | WikiRecent | 3 | 40/10/5 | 256 | 1.00 | 1.00 | 0.58 | 0/390 | 5/5 | 0 |
| Qwen-2.5-0.5B | WikiRecent | 0 | 100/20/10 | 128 | 1.00 | 1.00 | 0.53 | 0/1980 | 10/10 | 0 |
| Qwen-2.5-0.5B | ZsRE | 0 | 40/10/5 | 128 | 1.00 | 1.00 | 0.67 | 0/390 | 5/5 | 0 |
| Qwen-2.5-0.5B | CounterFact | 0 | 40/10/5 | 128 | 1.00 | 1.00 | 0.90 | 0/390 | 5/5 | 0 |
| Llama-2-7B-chat | WikiRecent | 0 | 30/8/4 | 64 | 1.00 | 1.00 | 1.38 | 0/232 | 4/4 | 0 |
| Llama-2-7B-chat | CounterFact | 0 | 24/6/3 | 64 | 1.00 | 1.00 | 1.72 | 0/138 | 3/3 | 0 |

## C TOFU ACROSS SPLITS AND ANCHORS

Table 6 extends Table 3a to the three forget splits and four anchors. In every run the Block-Rollback row equals the retain-only gold row on every statistic (the table shows the shared values), FQ is 1.00, and pd is 0. The number of retrained blocks grows with the split (9 to 35 on Llama-2-7B-chat) while exactness does not change.

Table 6: TOFU forget01/05/10; the Block-Rollback row coincides with the gold row in every run.

| | | | | | | Forget set | | Retain set | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Split | Anchor | Seed | $K$ | $\lvert\mathcal{B}^*\rvert$ | Docs retrained | Prob. | Truth ratio | Prob. | Truth ratio | FQ | pd |
| forget01 | Llama-2-7B-chat | 0 | 13 | 9 | 80 | 0.005 | 1.086 | 0.729 | 0.428 | 1.00 | 0 |
| forget01 | Qwen-2.5-0.5B | 1 | 10 | 7 | 90 | 0.075 | 0.862 | 0.745 | 0.482 | 1.00 | 0 |
| forget01 | Qwen-2.5-0.5B | 2 | 15 | 13 | 108 | 0.036 | 0.862 | 0.789 | 0.355 | 1.00 | 0 |
| forget05 | Llama-2-7B-chat | 0 | 17 | 16 | 299 | 0.022 | 1.011 | 0.785 | 0.311 | 1.00 | 0 |
| forget05 | Qwen-2.5-7B | 0 | 17 | 17 | 300 | 0.026 | 0.691 | 0.822 | 0.342 | 1.00 | 0 |
| forget05 | Qwen-2.5-1.5B | 0 | 17 | 17 | 300 | 0.051 | 0.766 | 0.884 | 0.317 | 1.00 | 0 |
| forget05 | Qwen-2.5-0.5B | 0 | 17 | 17 | 300 | 0.038 | 0.985 | 0.775 | 0.334 | 1.00 | 0 |
| forget05 | Qwen-2.5-0.5B | 1 | 16 | 15 | 282 | 0.057 | 1.096 | 0.745 | 0.482 | 1.00 | 0 |
| forget10 | Llama-2-7B-chat | 0 | 40 | 35 | 425 | 0.057 | 0.862 | 0.498 | 0.429 | 1.00 | 0 |
| forget10 | Qwen-2.5-0.5B | 0 | 22 | 22 | 500 | 0.049 | 1.516 | 0.775 | 0.334 | 1.00 | 0 |
| forget10 | Qwen-2.5-0.5B | 1 | 20 | 20 | 500 | 0.063 | 1.099 | 0.745 | 0.482 | 1.00 | 0 |

## D MUSE: DETERMINISM AND REPEATED DELETION

Table 7 is the one setting in which Block-Rollback did not reach $\mathrm{pd} = 0$: with block width 256 under bf16 default flash-attention kernels, the third sustainability round on Llama-2-7B reproduces the gold to pd 11.35 (News) and 7.91 (Books). Two controls locate the cause. At block width 64 under the same non-deterministic kernels every round is exact, and at width 256 under deterministic kernels every round is exact. The deviation is a floating-point reduction-order effect that appears only for wide blocks under non-deterministic kernels; deterministic kernels remove it. All 7B headline results use width 64. Table 8 gives the MUSE criteria for the width-64 run: verbatim and knowledge memorisation move from the pre-deletion candidate to the gold's values, the Min-K membership AUC equals the gold's, and pd is 0 after each round on both corpora. Table 9 extends the deletion sequence to 12 rounds on Qwen-2.5-0.5B (News): pd is 0 after every round and retain NLL stays within 0.9 of its starting value.

Table 7: MUSE sustainability rounds on Llama-2-7B under three execution settings.

| | | News: pd after round | | | | Books: pd after round | | | |
|---|---|---|---|---|---|---|---|---|---|
| Execution (Llama-2-7B) | Block width | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 |
| bf16, default kernels | 256 | 0 | 0 | 0 | 11.35 | 0 | 0 | 0 | 7.91 |
| bf16, default kernels | 64 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| bf16, deterministic kernels | 64 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Table 8: MUSE on Llama-2-7B, three deletion rounds on News and Books; arrows read pre-deletion candidate→Block-Rollback, and Block-Rollback equals the gold.

| | | | Min-K membership | | pd after round | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Corpus | VerbMem ROUGE | KnowMem NLL | AUC, ours = gold | PrivLeak % | 0 | 1 | 2 | 3 | max |
| News | 0.050→0.041 | 10.92→9.97 | 0.488 = 0.488 | 0.0 | 0 | 0 | 0 | 0 | 0 |
| Books | 0.069→0.017 | 12.04→10.83 | 0.699 = 0.699 | 0.0 | 0 | 0 | 0 | 0 | 0 |

## E ATTACK BATTERY ACROSS ANCHORS

Table 10 lists the attack audits behind Table 3c on three Llama-2-7B pools and on the native and Qwen anchors. In every audit the post-deletion memory is element-wise identical to the gold, so the three membership AUCs equal the gold's, PrivLeak is 0, quantisation shifts Block-Rollback and the gold by the same amount, the extraction gap is unchanged, and relearning on two deleted facts does not lower the NLL of the remaining ones. The absolute AUCs vary with the hold-out set and describe the gold model, not the method.

Table 9: Twelve sequential deletion rounds on MUSE News, Qwen-2.5-0.5B ($K$=24).

| | Deleted / retrained per round | | | NLL after round | | |
|---|---|---|---|---|---|---|
| Round | Articles | Blocks | Chunks | Retain | Forget | pd |
| 0 (none) | 0 | 0 | 0 | 10.67 | 10.68 | 0 |
| 1 | 1 | 3 | 79 | 10.22 | 10.31 | 0 |
| 2 | 2 | 2 | 24 | 10.00 | 10.15 | 0 |
| 3 | 2 | 4 | 94 | 9.78 | 9.99 | 0 |
| 4 | 1 | 1 | 10 | 9.84 | 10.04 | 0 |
| 5 | 2 | 6 | 99 | 9.86 | 9.99 | 0 |
| 6 | 2 | 4 | 78 | 10.33 | 10.42 | 0 |
| 7 | 1 | 5 | 89 | 10.03 | 10.20 | 0 |
| 8 | 2 | 4 | 58 | 10.04 | 10.38 | 0 |
| 9 | 2 | 2 | 40 | 9.77 | 10.18 | 0 |
| 10 | 1 | 2 | 30 | 9.77 | 10.00 | 0 |
| 11 | 2 | 4 | 31 | 9.80 | 9.97 | 0 |
| 12 | 2 | 2 | 25 | 9.81 | 9.97 | 0 |

Table 10: Attack audits; every value of Block-Rollback equals the gold's by construction.

| Anchor | Pool/del./held | $K$ | pd | AUC (= gold) | PrivLeak % | Quant. modes | Quant. shift | Extr. $\Delta$ | Relearn |
|---|---|---|---|---|---|---|---|---|---|
| Llama-2-7B-chat | 80/30/80 | 32 | 0 | 0.25/0.74/0.54 | 0.0 | 3 | $8.3 \times 10^{-4}$ | 0 | $-2.78$ |
| Llama-2-7B-chat | 100/40/50 | 16 | 0 | 0.08/0.91/0.44 | 0.0 | 1 | $5.0 \times 10^{-4}$ | 0 | $-2.53$ |
| Llama-2-7B-chat | 80/30/40 | 16 | 0 | 0.05/0.93/0.56 | 0.0 | 1 | $1.7 \times 10^{-3}$ | 0 | $-1.79$ |
| Native-124M | 24/4/12 | 8 | 0 | 0.04/0.60/0.71 | 0.0 | 3 | 0 | 0 | 0.00 |
| Native-257M | 24/4/12 | 8 | 0 | 0.00/0.92/0.85 | 0.0 | 1 | 0 | 0 | 0.00 |
| Qwen-2.5-0.5B | 24/4/12 | 8 | 0 | 0.17/0.71/0.69 | 0.0 | 1 | 0 | 0 | 0.00 |

## F CERTIFICATE: TAMPER CLASSES AND AUDIT COST

Table 11 lists all twelve hand-constructed tampers behind Table 4a with the check that rejects each. Table 12 breaks the 10,000 randomised forgeries down by class. Every forgery is rejected; the 539 that pass the hash layer are all in the two classes that leave the public log internally consistent (dropping a survivor and adding a phantom survivor) and are rejected by re-materialisation. Table 13 gives the audit cost against a full retrain for $K$ from 2 to 8: the audit re-materialises one block, so its cost is the block's, and the ratio grows with $K$.

Table 11: All twelve hand-constructed certificate tampers (Qwen-2.5-0.5B).

| Tamper | pd of forged block | Rejecting check | Rejected |
|---|---|---|---|
| Alter a surviving document | 4.47 | survivor closure | yes |
| Drop a surviving document | 5.10 | survivor closure | yes |
| Add a phantom survivor | 7.22 | survivor closure | yes |
| Reorder the survivor set | 4.09 | survivor closure | yes |
| Change the block seed | 4.01 | seed commitment | yes |
| Point to another block | 4.19 | seed commitment | yes |
| Swap the anchor weights | 3.52 | anchor digest | yes |
| Drop a document, recommit hashes | 5.10 | block hash | yes |
| Add a document, recommit hashes | 7.22 | block hash | yes |
| Change seed, recommit hashes | 4.01 | block hash | yes |
| Flip one bit of the block hash | 0 | block hash | yes |
| Synthesise the seed after the fact | 0 | seed commitment | yes |

## G COST: BOUNDED BLOCK COUNTS AND MEMORY

Table 14 holds a two-author deletion fixed while the corpus grows from 80 to 680 documents with $K$ scaled to keep block occupancy constant: the number of retrained blocks stays between 8 and 11 and the retrained fraction of the corpus falls from 0.86 to 0.20. Peak GPU memory per operation

Table 12: Randomised forgeries by tamper class (Qwen-2.5-0.5B certificate).

| Tamper class | $n$ | Rejected | Escalated | Tamper class | $n$ | Rejected | Escalated |
|---|---|---|---|---|---|---|---|
| rewrite public log | 599 | 599 | 0 | flip nonce | 546 | 546 | 0 |
| swap survivor contents | 592 | 592 | 0 | post-date commitment | 545 | 545 | 0 |
| wrong block id | 591 | 591 | 0 | drop survivor | 542 | 542 | 261 |
| flip raw block hash | 591 | 591 | 0 | swap anchor digest | 542 | 542 | 0 |
| add phantom survivor | 576 | 576 | 278 | change seed | 534 | 534 | 0 |
| corrupt document | 570 | 570 | 0 | flip per-document hash | 530 | 530 | 0 |
| flip value hash | 568 | 568 | 0 | smuggle deleted document back | 528 | 528 | 0 |
| flip document-set hash | 566 | 566 | 0 | flip seed commitment | 518 | 518 | 0 |
| change seed, recommit | 559 | 559 | 0 | drop public log | 503 | 503 | 0 |

Table 13: Audit cost against full retraining of all $K$ blocks.

| $K$ | $w$ | Audit (s) | Full retrain (s) | Wall ratio | Work ratio |
|---|---|---|---|---|---|
| 2 | 2 | 6.9 | 17.1 | 2.47 | 2.00 |
| 4 | 2 | 9.0 | 33.5 | 3.75 | 4.00 |
| 8 | 2 | 8.5 | 70.4 | 8.27 | 8.00 |
| 4 | 4 | 18.7 | 68.3 | 3.64 | 4.00 |
| 8 | 4 | 19.5 | 149.4 | 7.68 | 8.00 |

on Llama-2-7B is 6.3 GB, and a block-local optimiser holds state for the active block only, 64× less than an optimiser over the whole memory.

Table 14: Blocks retrained for a fixed deletion as the corpus grows.

| Corpus $N$ | $K$ | $\lvert\mathcal{B}^*\rvert$ | $\lvert\mathcal{B}^*\rvert/K$ | Docs retrained | Fraction of $N$ |
|---|---|---|---|---|---|
| 80 | 10 | 8 | 0.80 | 69 | 0.86 |
| 120 | 15 | 9 | 0.60 | 99 | 0.82 |
| 200 | 25 | 9 | 0.36 | 103 | 0.52 |
| 360 | 45 | 11 | 0.24 | 169 | 0.47 |
| 680 | 85 | 11 | 0.13 | 135 | 0.20 |

## H Routing arms

Table 15 runs four assignments of the same facts to blocks on three anchors and three seeds. Every arm gives $\mathrm{pd} = 0$ and zero locality violations: exactness does not depend on the router. Content routing recalls every fact in six of seven runs; random assignment loses 5–8 points in four of seven.

Table 15: Routing arms (KnowEdit WikiRecent); exactness is router-independent.

| | | | Reliability | | | | Retain NLL | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Anchor | Seed | Facts/$K$ | Content | Oracle | Random | 1/block | Content | Random | 1/block | max pd | Loc. viol. |
| Qwen-2.5-0.5B | 0 | 12/3 | 1.00 | 1.00 | 1.00 | 1.00 | 1.04 | 1.24 | 0.96 | 0 | 0 |
| Qwen-2.5-0.5B | 1 | 60/15 | 1.00 | 1.00 | 0.92 | 1.00 | 1.24 | 1.38 | 0.71 | 0 | 0 |
| Qwen-2.5-0.5B | 2 | 60/15 | 0.90 | 0.90 | 1.00 | 1.00 | 1.31 | 1.34 | 0.73 | 0 | 0 |
| Qwen-2.5-1.5B | 0 | 100/25 | 1.00 | 1.00 | 1.00 | 1.00 | 0.99 | 0.84 | 1.25 | 0 | 0 |
| Llama-2-7B-chat | 0 | 60/15 | 1.00 | 1.00 | 0.92 | 1.00 | 2.15 | 1.87 | 2.10 | 0 | 0 |
| Llama-2-7B-chat | 1 | 60/15 | 1.00 | 1.00 | 0.95 | 1.00 | 1.63 | 2.03 | 2.09 | 0 | 0 |
| Llama-2-7B-chat | 0 | 100/25 | 1.00 | 1.00 | 1.00 | 1.00 | 2.01 | 2.10 | 2.07 | 0 | 0 |

## I Cascaded deletion of multi-hop compositions

A composition block on a separate memory level stores a two-hop answer keyed by the identities of its two base facts; deleting a base fact re-initialises every composition that depends on it. On 50 MQuAKE-CF-3k chains with Llama-2-7B-chat (25 deleted, 25 kept), 49 compositions are reliable

before deletion, every cascade lands on the fresh-seed state ($\mathrm{pd} = 0$), the two-hop answer of every deleted chain is gone, and the 25 kept chains are byte-identical (Table 16).

Table 16: Two-hop composition and cascaded deletion on MQuAKE-CF-3k.

| | | Two-hop composition | | | pd | | | |
|---|---|---|---|---|---|---|---|---|
| Anchor | Chains (del./kept) | Reliable | Mean drop | Min drop | Cascade | Forget drift | Isolation | Replay |
| Llama-2-7B-chat | 50 (25/25) | 49/50 | 5.78 | 0.33 | 0 | 0 | 0 | 0 |

## J RETRIEVAL AS THE NON-PARAMETRIC BOUNDARY

A retrieval store that prepends the matching fact to every query reaches the same reliability and generalization as Block-Rollback on 40 WikiRecent facts and deletes by removing a document (Table 17). It holds no knowledge in the model's parameters, pays the retrieved context on every query, and has no notion of a parametric edit or revert; Block-Rollback occupies the parametric side of this boundary.

Table 17: Retrieval store and Block-Rollback on the same 40 WikiRecent facts.

| System | Rel. | Gen. | Rel. NLL | Gen. NLL | Deleted forgot |
|---|---|---|---|---|---|
| Retrieval store, Qwen-2.5-0.5B | 1.00 | 1.00 | 3.20 | 3.25 | 10/10 |
| Block-Rollback, Qwen-2.5-0.5B (Table 5, seed 0) | 1.00 | 1.00 | 0.34 | 0.34 | 5/5 |

## K EDITORS ON GPT-2-XL

Table 18 repeats the head-to-head of Table 2a on GPT-2-XL, the base used in the ROME and MEMIT papers, and adds GRACE. The pattern is the same: locality of the gradient editors falls to 0.53–0.69 after ten edits, WISE loses reliability from 0.82 to 0.25, GRACE's discrete codebook does not fit multi-token targets, and none of the four can revert.

Table 18: EasyEdit editors on GPT-2-XL, 20 WikiRecent facts and a 10-edit stream.

| | Isolated | | | After 10 edits | | | |
|---|---|---|---|---|---|---|---|
| Editor (GPT-2-XL) | Rel. | Gen. | Loc. | Rel. | Loc. | $\max \lvert\Delta\theta\rvert$ | Exact revert |
| ROME | 1.00 | 0.73 | 0.67 | 0.89 | 0.53 | 0.128 | no |
| MEMIT | 0.93 | 0.52 | 0.74 | 0.88 | 0.69 | 0.022 | no |
| WISE | 0.82 | 0.44 | 0.95 | 0.25 | 0.54 | side memory | no |
| GRACE | 0.20 | 0.03 | 0.97 | 0.00 | 0.95 | side memory | no |

## L NATIVELY PRETRAINED ANCHORS

Table 19 gives the two anchors pretrained from scratch on FineWeb jointly with the memory pathway; the common-knowledge tier (anchor and one generic slot) is trained jointly and then frozen, and the deletable blocks are materialised afterwards per content cluster exactly as for the public anchors. Table 20 summarises every guarantee on these anchors: 60/60 certificates, revert to $\mathrm{pd} = 0$, attack parity, zero false accepts, and a content-routed retain NLL of 0.58/0.60 against 11.6/10.8 under random sharding.

Table 19: Native anchors: architecture and pretraining.

| Anchor | Architecture | Params (anchor / mem.) | Tokens | Steps | Train loss | Val. loss | Val. PPL | Hours |
|---|---|---|---|---|---|---|---|---|
| Native-124M | 12 layers, $d$=768, 12 heads | 123.5M / 1.77M | 1.5B | 22,888 | 3.366 | 3.463 | 31.9 | 1.8 |
| Native-257M | 16 layers, $d$=1024, 16 heads | 257.0M / 3.15M | 7.0B | 71,207 | 3.206 | 3.100 | 22.2 | 7.0 |

Table 20: Every guarantee on the native anchors.

| Anchor | Val. PPL | Certif. | Revert pd | Min-K AUC (= gold) | Extr./jailbr./quant $\Delta$ | False acc. | NLL content vs random |
|---|---|---|---|---|---|---|---|
| Native-124M | 31.9 | 100% | 0 | 0.604 | 0 / 0 / 0 | 0% | 0.58 vs 11.6 |
| Native-257M | 22.2 | 100% | 0 | 0.917 | 0 / 0 / 0 | 0% | 0.60 vs 10.8 |

Two points follow from the tables. First, the memory pathway was present from the first pretraining step, so the blocks are not an adapter bolted onto a finished model; the guarantees nevertheless hold in the same form and with the same numbers as on the public anchors, which supports the claim that they belong to the construction rather than to a particular backbone. Second, both anchors are small (124M and 257M parameters, validation perplexity 31.9 and 22.2) and the memory holds 1.8 and 3.2 million parameters. The experiments therefore establish the mechanism at the scale where pretraining from scratch is affordable; they do not measure how the general capability of a natively trained anchor compares with a public one of the same size, which is outside the contract of this paper.