EDIT THE MODEL YOU DEPLOY: QUANTIZATION WEAKENS KNOWLEDGE EDITS THROUGH THEIR CONDITIONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge edits and unlearning updates are computed on full-precision weights, but the served model is quantized, which can weaken or undo them. For unlearning, the explanation given is that an update smaller than the quantization step is rounded away. We show that this explanation does not carry over to closed-form (locate-and-edit) edits. By a classical dither argument, rounding on a fixed, evenly spaced grid without clipping is *unbiased* on an update of any size when the weights' bin positions are uniform and independent of it, and on four models single edits below the bin width keep at least 99% of their margin, on average, when only the edited matrix is quantized. The damage comes instead from the model the edit was solved against: quantized layers below the edit move the key it fires at, and quantized layers above change the read-out its value was fitted to. Quantizing everything except the edited matrix reproduces 90–101% of the damage to a single MEMIT edit of one layer (33–91% of it generic at int3), on seven models (124M–8B), two datasets and weight-only round-to-nearest, NF4, AWQ and served GPTQ and AWQ at 3 and 4 bits. At the served int4, an edit loses 4–16% of its margin, and at 3 bits 20–56%. Solving the same closed form inside the quantized model returns 90–100% of the 3-bit single-edit loss on four of five models (72–96% for released MEMIT), 77–104% under 3-bit GPTQ and AWQ, and 72–95% on held-out paraphrases; on Qwen2.5-3B, whose symmetric int3 model is itself broken, it buys efficacy at the cost of locality. Where the dither's assumptions fail, the explanation holds: gradient unlearning at a small learning rate, stored in bf16, changes mainly near-zero weights, whose bin positions are not uniform, and int4 rounding delivers 5–29% of such an update for seven of eight methods on a 1B model. All code and result files will be public at `https://github.com/xxx/xxx` upon acceptance.

1 INTRODUCTION

Locate-and-edit methods (Meng et al., 2022; 2023; Fang et al., 2025) rewrite a fact by adding a low-rank update ΔW to one MLP matrix or a band of them, and the same closed form with a suppressing target removes one. The update is computed on full-precision weights, but the deployed model is almost never full precision: compression is known to hinder editing (Kolbeinsson et al., 2025), and 4-bit quantization can reverse gradient-based unlearning (Zhang et al., 2025). The explanation given for unlearning, since formalized (Sadhu et al., 2026) and turned into training objectives (Mishra & Mehreen, 2026; Khan & Sohel, 2026), is that the update is small: its entries fall below the quantization step Δ, so rounding maps $W + \Delta W$ back onto the grid point of W. The design rule that follows is to make updates *large*.

For closed-form edits, whose mechanism Kolbeinsson et al. (2025) left open, this is the only explanation on offer, and we show that it does not apply to them unless the weights sit on or near the grid. Rounding on a fixed, evenly spaced grid, with weights uniform in their bins and independent of the update, is unbiased on a displacement of any size (Lemma 1): a sub-bin update is not erased but turned into sparse $\pm\Delta$ jumps whose mean is the update. What quantization changes is the model the edit was solved against. A locate-and-edit update is a rank-one map $\Delta W \propto a\,u^\top$ that fires at a key k^*, which every quantized layer below moves, and its value was fitted to a read-out that every quantized layer above changes. Our contributions:

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.

- **Rounding does not erase sub-bin edits.** On average, single edits of six closed-form editors move the edited entries by less than a bin at int3, int4 and NF4 and keep at least 99% of their margin when only the edited matrix is quantized (four models, §4.3). The rounding error on the update is noise whose size a closed form, the edit signal-to-noise ratio (Proposition 1), matches to within 3% for MEMIT's single edits on the four main models (in sample; Appendix B); it transfers to AWQ with a fixed scale and to FT-L's dense update, but not to GPTQ and AWQ as served, which move the grid (§4.4).
- **The damage is in the conditioning.** Quantizing everything *except* the edited matrix reproduces 90–101% of the damage to a single edit of one layer (§4.1, Table 11), including at the int4 of serving stacks, where GPTQ and AWQ cost an edit 4–16% of its margin (§4.2). The edited matrix carries about a tenth or less of such an edit's damage and matters mainly where a batch overloads it. Part of the conditioning loss is generic: at equal confidence, a fact the model already knows loses 33–91% as much as an edit on five models (§4.1).
- **Editing the deployed model returns most of the loss.** Quantization-aware editing (QAE) solves the same closed form at the deployed key, with a value re-optimized through the quantized network (§3.3). At 3 bits it returns 90–100% of the single-edit loss on four of five models and 77–104% under GPTQ and AWQ; it returns 72–95% on paraphrases its objective never sees, and 83–101% on fresh facts' paraphrases on three models (§4.5). At int4 the loss is small and QAE's share imprecise; on Qwen2.5-3B's symmetric int3 model the fix fails, and QAE's edit is specific to its precision.

The phenomenon was first reported by Kolbeinsson et al. (2025), and solving inside a quantized model has been done on device to save memory (Lu et al., 2026); we claim the mechanism and the evidence that solving there returns most of the 3-bit loss. Of twelve predictions logged before their runs, five failed and one held in part; all are reported (Appendix Z).

2 Related work

Editing under compression. Kolbeinsson et al. (2025) found, across 417 compositions of interventions, that compression hinders editing. Zhang et al. (2025) showed that 4-bit quantization recovers unlearned knowledge and attributed it to weight changes smaller than the step; the argument has become a proposition with a magnitude floor (Sadhu et al., 2026), a logit-space hinge with a half-step margin (Mishra & Mehreen, 2026) and straight-through training at the target precision (Khan & Sohel, 2026). We run the floor and a weight-space analogue of the hinge as post-hoc transforms of MEMIT's update (Appendix K). Compression also degrades what a model knows in general (Namburi et al., 2023; Jaiswal et al., 2024; Wang et al., 2026). Basani & Chhabra (2026) examine how edits respond to perturbation; the key displacement we measure is coherent, and our isotropic key-robust control fails (Appendices U, N). Closest to our fix, Lu et al. (2026) solve ROME inside the W8A16 model they serve on phones, with the edited projection in floating point; their motivation is memory, not durability. We quantize the edited matrix too and measure each half of the fix. Lakra & Feng (2026) edit quantized models sequentially; Egashira et al. (2024) and Yan et al. (2025) study quantization-conditioned behaviour and full-precision key perturbation, respectively. Gupta et al. (2024) traced ROME's model-breaking edits to a solve with two different keys; the key channel is the deployment-time version of that mismatch.

Quantization theory. That a uniform quantizer is unbiased on a dithered input is classical (Schuchman, 1964; Gray & Neuhoff, 1998) and underlies stochastic rounding (Gupta et al., 2015); we apply it with the roles reversed, the weights being the dither and the update the signal. Zhou et al. (2026) and Duan (2026) predict the quantization damage of unedited models from full-precision statistics; ours is per edit.

3 Two channels of quantization damage

Setup. A locate-and-edit method modifies one MLP down-projection $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, $v = Wk$, where k is the MLP hidden activation at the subject's last token (the *key*) and v is written to the residual stream. For a single fact the update is rank one,

$$\Delta W = \frac{a\, u^\top}{u^\top k^*}, \qquad a = v^* - Wk^*, \qquad u = C^{-1}k^*, \qquad C = \mathbb{E}[kk^\top], \tag{1}$$

so that $(W + \Delta W)k^* = v^*$. For n facts with keys K and residuals R, MEMIT solves $\Delta W = RK^\top(C + KK^\top + \lambda I)^{-1}$, with per-fact direction $u_i = (C + KK^\top + \lambda I)^{-1}k_i$; at $n = 1$ this is $s/(1+s)$ times (1) with $(C+\lambda I)^{-1}$ for C^{-1}, where $s = {k^*}^\top(C+\lambda I)^{-1}k^* \gg 1$ (published MEMIT weights C and spreads the edit over layers; Appendix A). Write $\mathcal{Q}$ for a group-wise round-to-nearest quantizer with per-entry step Δ_{ij}, and $\delta_{ij} := \Delta W_{ij}$. The deployed model is $\mathcal{Q}(W + \Delta W)$ and its baseline $\mathcal{Q}(W)$, so what matters is their difference and, separately, what quantized layers below and above do to k^* and the read-out.

3.1 The weight channel: rounding is unbiased on the edit

Decompose $\mathcal{Q}(W + \Delta W) = \mathcal{Q}(W) + \Delta W + \tilde{E}$, with $\tilde{E} := [\mathcal{Q}(W + \Delta W) - \mathcal{Q}(W)] - \Delta W$ the rounding error *on the update*.

Lemma 1 (Dither lemma)**.** *Assume (i) the quantizer uses the same grid on W and on $W + \Delta W$, with points at the integer multiples of Δ_{ij}, (ii) the phase $\varphi_{ij} = (W_{ij}/\Delta_{ij} + 1/2) \bmod 1$ is uniform on $[0, 1)$ and independent of δ_{ij}, and (iii) the grid is evenly spaced and neither W_{ij} nor $W_{ij} + \delta_{ij}$ is clipped for any phase. Then for every displacement δ_{ij}, with expectations over the phase,*

$$\mathbb{E}[\tilde{E}_{ij}] = 0, \qquad \mathrm{Var}[\tilde{E}_{ij}] = \Delta_{ij} r_{ij} - r_{ij}^2, \qquad r_{ij} := |\delta_{ij}| \bmod \Delta_{ij}. \tag{2}$$

Proof. Write $\delta = (m + t)\Delta \operatorname{sgn}(\delta)$ with $m \in \mathbb{Z}_{\geq 0}$ and $t = r/\Delta \in [0, 1)$. On an evenly spaced, unclipped grid rounding commutes with whole-step shifts, so $\mathcal{Q}(w+\delta) - \mathcal{Q}(w) = \operatorname{sgn}(\delta)\Delta(m+B)$ with $B \sim \mathrm{Bernoulli}(t)$ under (ii). The mean is δ and the variance $\Delta^2 t(1-t) = \Delta r - r^2$. □

The variance is *periodic* in the displacement: a displacement of exactly one bin is noiseless. Measured with the quantizer's own rounding, a small common displacement is delivered as (ii) predicts to within 0.5% at int3 and int4 on six of the seven main-grid checkpoints, not on Qwen2.5-3B, and fails for each group's absmax entry (Appendix B); a checkpoint stored on the deployment grid violates it outright. NF4's uneven codebook is outside (iii), so its cells are an empirical extension. A dense update changes every group's largest $|W_{ij}|$, so (i) holds only approximately: the step moves negligibly for a single edit on a synthetic matrix (not measured on real ones), by about a fifth for most groups of a batch of 100, and several-fold at $n = 1000$ (Appendix B).

Corollary 1 (Sub-bin updates survive in expectation)**.** *Under Lemma 1, for every $\delta \neq 0$, however far below Δ, $\mathbb{E}[\mathcal{Q}(w + \delta) - \mathcal{Q}(w)] = \delta$: for $|\delta| < \Delta$ an entry jumps by $\operatorname{sgn}(\delta)\Delta$ with probability $|\delta|/\Delta$ and otherwise stays at $\mathcal{Q}(w)$, so the mean is the update. The criterion "an update smaller than the step is rounded away" (its half-step version is exact only for a weight on a grid point) fails for the mean; whether a readout survives is a signal-to-noise question over d_{in} channels.*

The deployed edit's output at its own key is $(\Delta W + \tilde{E})k^* = \Delta W k^* + \tilde{E}k^*$, with $\mathbb{E}[\tilde{E}k^*] = 0$ and, for rounding errors uncorrelated within a row, $\mathrm{Var}[(\tilde{E}k^*)_i] = \sum_j (\Delta_{ij} r_{ij} - r_{ij}^2){k_j^*}^2$.

Proposition 1 (Edit signal-to-noise ratio)**.** *Define*

$$\mathrm{ESNR}^2 = \frac{\|\Delta W k^*\|^2}{\sum_{i,j} \left(\Delta_{ij} r_{ij} - r_{ij}^2\right) {k_j^*}^2}, \qquad r_{ij} = |\delta_{ij}| \bmod \Delta_{ij}. \tag{3}$$

Under Lemma 1, with the rounding errors uncorrelated within each row, the root-mean-square relative readout error of the deployed edit at its own key, $(\mathbb{E}\|\tilde{E}k^*\|^2)^{1/2}/\|\Delta W k^*\|$, equals ESNR^{-1}, and the mean relative error is at most this.

ESNR uses ΔW, k^* and the step map of $\mathcal{Q}$ on $W + \Delta W$, at cost $O(d_{\mathrm{in}} d_{\mathrm{out}})$, and applies to any ΔW under (i)–(iii) and uncorrelated row errors, including fixed-scale AWQ; served AWQ, which re-searches its scale and clip on $W + \Delta W$, breaks (i), and GPTQ's correlated errors are outside it. Its value lies not in cost but in being a closed form in the update, which the next corollary relates to the update's alignment with the key.

Corollary 2 (Alignment)**.** *For a rank-one edit on one step Δ with almost every entry below one bin, whose $|u_j|$ and u_j^2 are uncorrelated with ${k_j^*}^2$, with $\bar{\delta}$ the mean $|\delta_{ij}|$, $\kappa = \kappa_a \kappa_u$ and $\kappa_x = \overline{x^2}/\overline{|x|}^2$ over the entries of x,*

$$\mathrm{ESNR}^2 \approx \frac{\kappa\, d_{\mathrm{in}} \cos^2(u, k^*)\, \bar{\delta}/\Delta}{1 - \kappa\, \bar{\delta}/\Delta}, \tag{4}$$

so the readout error equals the edit ($\mathrm{ESNR} = 1$*) at* $r^* = \bar{\delta}/\Delta \approx \left[\kappa\left(d_{\text{in}}\cos^2(u, k^*) + 1\right)\right]^{-1}$ *(the formula needs* $\kappa\bar{\delta}/\Delta < 1$*).*

For Gaussian entries $\kappa = \pi^2/4$, so an update with $d_{\text{in}}\cos^2 = 1$, the mean for a random direction, has $r^* \approx 1/(2\kappa) \approx 0.20$ (about 2% of the entries then exceed a bin; the measured threshold is 0.21–0.22): a fixed fraction of a bin set by the entries' shape, *which is the form of the bin-width criterion*. An update aligned with the key survives at a per-entry magnitude about $d_{\text{in}}/4$ times smaller than a random direction for Gaussian keys (alignment makes $|u_j|$ track $|k_j^*|$, which the correlated form of Appendix C accounts for; $(d_{\text{in}} + 1)/2$ holds only for keys of constant magnitude). $u = C^{-1}k^*$ is not randomly oriented. On synthetic W the threshold is predicted to within 11% for $\cos \leq 0.2$ where $d_{\text{in}}\cos^2 \geq 1$ and under-predicted by up to $2\times$ at $\cos \geq 0.7$, where $|u_j|$ and k_j^* correlate. Since $\|\Delta W\|_F = \|a\|/(\|k^*\|\,|\cos(u, k^*)|)$, raising cos at a fixed residual a shrinks the update and raises ESNR when $\kappa\bar{\delta}/\Delta < \frac{1}{2}$ (equivalently $d_{\text{in}}\cos^2 > 1$ at the threshold); MEMIT's single edits at int3 and int4 have $\bar{\delta}/\Delta \leq 0.10$ on the four main models (defined in §4) and 0.17 on Qwen2.5-3B, inside that condition for Gaussian κ. The corollary is a design heuristic, not a calibrated predictor: on real MEMIT edits, whose $|u_j|$ and k_j^* correlate, (4) with Gaussian κ under-predicts the measured readout error by 1.15–1.56× (Appendix C). For single edits the weight channel is already negligible (§4.1), so the corollary governs the channel that matters least; where that channel is large, in overloaded batches outside its hypotheses, the alignment ridge removes 73–95% of it (Appendix N).

3.2 The key channel

Upstream layers are quantized too, so the deployed model reads $k_q^* := k^* + \delta k$ instead of k^*.

Proposition 2 (Key signal-to-noise ratio)**.** *For a rank-one edit of the form* (1)*,* $\Delta W(k^* + \delta k) = \rho\, a$ *with* $\rho := 1 + u^\top \delta k / u^\top k^*$*, so with* $\mathrm{KSNR} := |u^\top k^*|/|u^\top \delta k|$ *the relative error of the delivered edit is* $\mathrm{KSNR}^{-1} = |\rho - 1| = \left(|\cos(u, \delta k)|/|\cos(u, k^*)|\right)\|\delta k\|/\|k^*\|$.

The measured displacement is coherent with the update: at int3 the median $|\cos(u, \delta k)|$ is 0.29–0.42 on the four main models, 42–74 times the median $|\cos|$ of a random direction, and $\cos(u, \delta k)$ is negative on average (-0.29 to -0.40; -0.14 on Qwen2.5-3B; Appendix U). The delivered edit is therefore scaled by ρ, which one forward pass through the quantized backbone gives before any edit is made: 0.49–0.71 on the four main models, and -0.014 on Qwen2.5-3B, where the edit is annihilated. Across the five models ρ tracks the fraction of its margin MEMIT keeps at int3 (Spearman $+0.90$, exact two-sided $p = 0.08$; $+0.80$, $p = 0.33$, without Qwen2.5-3B; weaker per edit). Dividing MEMIT's update by ρ comes within 0.41 nats of re-solving at k_q^* in all eight cells tested (two models), though the difference is significant in three; the match is empirical, since the rescaling corrects the edit's output at k_q^* but not the unedited layer's shift $W\delta k$ (Appendix U).

3.3 Design consequences

Besides its own weights, a locate-and-edit update depends on two things the deployed model changes, both measurable before deployment. It fires at a *key*: quantizing the layers below the edit replaces k^* by k_q^*, which one forward pass through the quantized backbone reads off. It is fitted against a *read-out*: v^* is chosen so that the full-precision layers above turn the edited layer's output into the new object; running the same optimization through the quantized model, with the edited matrix in full precision, gives v_q^*. **Quantization-aware editing (QAE)** substitutes both into (1): it solves with k_q^* and residual $v_q^* - Wk_q^*$, with W and C unchanged and no new hyperparameter. Its halves can be taken separately: key-aware editing (KAE) uses (k_q^*, v^*) and value-aware editing (k^*, v_q^*). Two controls modify the solve: the **alignment ridge** $u_\lambda = (C + \lambda I)^{-1}k^*$ with $\lambda = 3\bar{\lambda}$ against MEMIT's $10^{-3}\bar{\lambda}$ ($\bar{\lambda}$ the mean diagonal of C), which raises $\cos(u, k^*)$ to 0.88–0.95 for single edits, and **key-robust editing (KRE)**, which adds four isotropic noisy copies of k^* to K and does not work (Appendix N). Post-hoc floors and hinges enlarge MEMIT's $n = 1$ update 3–15×; they add deployed margin on GPT-2-XL although MEMIT's own weight channel there is at most 0.05 nats, so not by reducing rounding error (mechanism not measured), lose margin on Llama-3.2-3B (significantly in four of six cells), and cost locality (Appendix K).

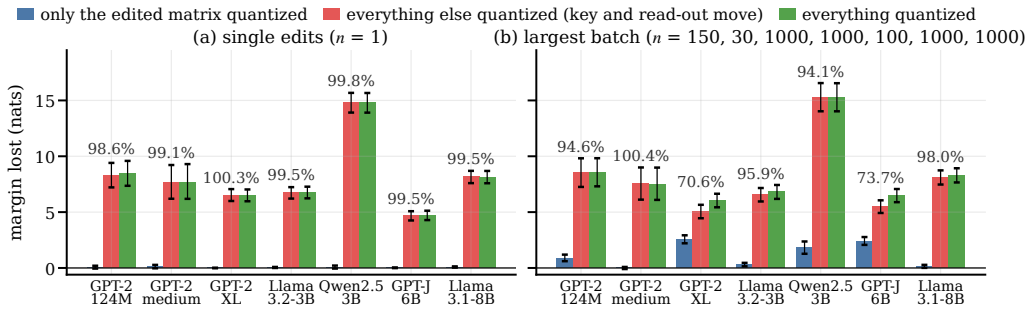


Figure 1: **Where the damage comes from.** MEMIT, int3, rewrite margin lost relative to the unquantized edit (nats, paired 95% CI) when only the edited matrix is quantized (blue), when everything *except* it is (red: key and read-out move), and when everything is (green); the label is the even split, which gives the channels' interaction half to each. (a) Single edits, seven models. (b) The largest batch in each model's grid; GPT-2-XL and GPT-J at $n = 1000$ and Qwen2.5-3B at $n = 100$ overload the edited matrix.

4 Experiments

Models, data, quantizers, editors. Seven models from four families, $d_{\text{in}} = 3072$–16384: GPT-2 (124M), GPT-2-medium (355M), GPT-2-XL (1.5B), Llama-3.2-3B, Qwen2.5-3B, GPT-J-6B and Llama-3.1-8B, each edited at one MLP down-projection (layers 5/8/17/4/5/5/5); the Instruct checkpoints (Llamas, Qwen) get no chat template. *The four main models* are GPT-2-XL, Llama-3.2-3B, GPT-J-6B and Llama-3.1-8B, and with Qwen2.5-3B, the five main models. Data are 1,600-record samples of CounterFact (Meng et al., 2022) and of ZsRE (Levy et al., 2017), the latter on GPT-2-XL and Llama-3.2-3B at $n \leq 100$. Quantizers act on every linear layer of the blocks and are weight-only except in §4.2. The main grid uses symmetric absmax round-to-nearest in groups of 64 at 8, 4 and 3 bits, and NF4 (Dettmers et al., 2023); separate runs add AWQ (Lin et al., 2024) with its scale fixed on the unedited matrix (same probes) and GPTQ (Frantar et al., 2023) on the edited matrix alone (12–25 probes) and on the whole backbone (100 probes; Appendices Q and Y). §4.2 uses the serving configuration, zero-point groups of 128 with round-to-nearest, GPTQ or AWQ over the whole model, and W8A8 and W4A8. Editors are MEMIT (single-layer, unweighted C), a single-layer AlphaEdit-style variant, KAE, the alignment ridge, KRE and KAE+KRE (the six editors of the main grid), FT-L (Zhu et al., 2020), and in §4.5 QAE and value-aware editing; MEMIT and AlphaEdit with their released settings are in Appendix T. C is estimated from 500,000 tokens (100,000 on GPT-2 and GPT-2-medium) of WikiText-103 (Merity et al., 2017). Batch sizes are $n \in \{1, 100, 1000\}$ on the four main models, $\{1, 100\}$ on Qwen2.5-3B, $\{1, 150\}$ on GPT-2 and $\{1, 30\}$ on GPT-2-medium.

Channels and metrics. Every grid cell is evaluated in four channels: *fp* (nothing quantized), *weights-only* (the edited matrix quantized; its damage includes the rounding of W itself), *keys-only* (everything but the edited matrix quantized; the *conditioning* channel, moving the key and the read-out) and *full*. Appendix G splits keys-only into an *upstream* channel (only the matrices feeding the key) and a *downstream* one (the rest). The per-edit quantity is the rewrite margin $m = \log P(o_{\text{new}} \mid p) - \log P(o_{\text{true}} \mid p)$ (nats, length-normalized), with efficacy $\text{ES} = \mathbf{1}[m > 0]$, paraphrase success PS the same indicator on the mean margin over two paraphrases, and neighbourhood success NS the indicator that three neighbouring subjects prefer their true object on average. *Damage* is the fp margin minus the channel margin, paired per probe. With D_K, D_W and D_F the keys-only, weights-only and full damages, the *conditioning share* is D_K/D_F (written K/F in the tables), a ratio of cell means that can exceed 100% through interaction or noise, and the *even split* $[D_K + (D_F - D_W)]/(2D_F)$ gives the interaction to both channels equally. A fix's *recovered* fraction is the share of MEMIT's deployed damage it returns, $(m^{\text{fix}}_{\text{dep}} - m^{\text{MEMIT}}_{\text{dep}})/(m^{\text{MEMIT}}_{\text{fp}} - m^{\text{MEMIT}}_{\text{dep}})$. Unless stated, intervals are 95% bootstrap CIs over the probe facts of a cell (paired differences, or ratios of means with probes resampled jointly; 20,000 resamples): 200 probes at $n \in \{1, 1000\}$ and 100 at $n = 100$ and in §4.2 on the main models, 40 and 20 on GPT-2 and GPT-2-medium. A batch cell is one solve, so its interval covers probes, not batch composition; Appendix D adds ten disjoint batches of 100 on three models, three on GPT-J-6B.

Table 1: **At the precision that is served.** Single edits (MEMIT, the first 100 of the 200 probes of Table 2, from a separate run), every linear layer in zero-point groups of 128 by our GPTQ/AWQ (each cell: GPTQ/AWQ; Appendix E). ppl: perplexity ratio to the unquantized model; fp: unquantized margin; lost: deployed margin loss (nats, % of fp); K/F: conditioning share (%); flips: edits whose efficacy (ES) or paraphrase success (PS) is lost (counts); QAE: % of the loss that solving in the served model returns (in sample; held-out, 3 bits: Table 21; at 4 bits, where the loss is under 2.6 nats, its intervals span 21–194%). Intervals and batches of 100: Tables 15 and 16.

model	bits	ppl	fp	lost (nats)	lost (%)	K/F	ES flips	PS flips	QAE
GPT-2-XL	4	1.01/1.01	13.21	0.48/0.58	4/4	96/90	0/0	2/3	128/81
	3	1.06/1.05	13.21	2.68/2.98	20/23	98/91	0/0	9/13	104/93
Llama-3.2-3B	4	1.04/1.05	14.03	1.32/1.09	9/8	99/96	0/0	1/2	84/90
	3	1.23/1.28	14.03	5.02/3.83	36/27	100/96	3/0	8/3	99/89
Llama-3.1-8B	4	1.04/1.07	15.33	1.71/1.60	11/10	99/99	0/0	0/0	83/92
	3	1.22/1.36	15.33	6.33/5.16	41/34	100/96	2/0	3/0	98/90
Qwen2.5-3B	4	1.05/1.07	15.61	2.51/2.05	16/13	96/97	1/2	3/2	93/62
	3	1.23/1.34	15.61	8.73/6.73	56/43	99/97	11/7	9/6	77/86

4.1 The damage is in the conditioning, not in the edited weights

Single edits. At int3, quantizing the edited matrix alone changes MEMIT's margin by at most 0.09 nats in the five main models' grid, against full-channel losses of 4.70–14.78 nats, and quantizing everything else reproduces 99.3–100.5% of the full loss (Table 11; even split in Figure 1); on Qwen2.5-3B the conditioning alone erases the edit. At int4 and NF4 the share is 91.5–99.7%, the weight channel reaching 8% of the loss at NF4 on GPT-2-XL; int3 included, it is 97.9–101.0% under fixed-scale AWQ and 94.6–100.2% on ZsRE. The key moves: under an int3 backbone its displacement $\|\delta k\|/\|k^*\|$ is 0.59–0.91 on the four main models and 1.68 on Qwen2.5-3B, against 0.015–0.042 at int8.

The key and the read-out. For single edits at int3, quantizing only the layers below the edit costs MEMIT 43–76% of the keys-only loss on the four main models, and only those above it 39–70% (Appendix G); the halves are not additive, since the layers below also move the residual stream at the prediction position. KAE, which re-solves at k_q^*, returns 73–107% of the upstream loss but only 25–53% of the keys-only loss: most of what it leaves is the read-out half, which no re-solve at the key can reach.

Part of the loss is generic. A quantized backbone also degrades confident predictions the model already had. Within one fp-margin window (Appendix H), adjusting for the edits' higher mean margin, a known fact loses 33–69% as much as an edit in the keys-only channel at int3 on the four main models (91% on Qwen2.5-3B). Most of the loss is edit-specific on GPT-2-XL and, by the point estimate, GPT-J-6B; on Llama-3.2-3B most of it is generic, and on Llama-3.1-8B about half.

Where the weight channel matters. We call batches where it matters *overloaded* (Appendix A). At $n = 1000$ and int3 on GPT-2-XL and GPT-J, where a thousand facts on one unweighted matrix drive $\|\Delta W\|_F/\|W\|_F$ to 11.6 and 7.5 and fp perplexity to $5.4\times$ and $6.5\times$, it costs 2.56 of 6.04 and 2.41 of 6.47 nats (weight share 29% and 26% by the even split; 29–54% at int4/NF4); MEMIT and AlphaEdit with their released settings avoid this regime (int3 share 91.5–96.5%; Appendix T), and the Llamas' $n = 1000$ updates (2.2 and 1.7 times the norm) leave it under 5%. Pooled over ten disjoint batches of 100 (Appendix D), the conditioning share is 95.2% at int3 and 91.3% at int4 on GPT-2-XL, four of whose batches exceed the matrix's norm (up to fourfold), 99.1% and 99.9% on Llama-3.2-3B, and 99.5% and 99.3% on Llama-3.1-8B (at NF4, 84.5% on GPT-2-XL). Five-layer bands keep it at 86.7% or more at int3 (84.1% at int4 on GPT-2; Appendix S), and group sizes from 64 to per-channel at 96.3–100.5% (Appendix M).

4.2 At the precision that is served

Serving toolchains ship int4 as W4A16: groups of 128 over every linear layer, quantized by GPTQ or AWQ (AutoGPTQ, AutoAWQ, llm-compressor). Table 1 runs MEMIT's channels and QAE

in that setup, reimplemented, on GPT-2-XL, Llama-3.2-3B, Llama-3.1-8B and Qwen2.5-3B, with zero-point groups, act-order GPTQ with static groups and AWQ with its scale and clipping searches (Appendix E); the edited matrix is re-quantized per edit by the same algorithm, and the layers above it keep the unedited model's calibration. **At 4 bits**, which cost the unedited model 1–7% in perplexity, a single edit loses 4–16% of its margin (0.5–2.5 nats), 3 of 800 edit–quantizer pairs lose efficacy (all on Qwen2.5-3B), and the conditioning share is 90–99%. **At 3 bits**, which cost the model 5–36% in perplexity, an edit loses 20–56% of its margin, 23 of 800 pairs lose efficacy, the share is 91–100%, and QAE returns 89–104% on the first three models and 77–86% on Qwen2.5-3B. At 4 bits QAE's shares are imprecise (intervals spanning 21–194%) and its efficacy gain unmeasurable.

At 3 bits, and at 4 bits on the Llamas, better quantizers of the model are only partly better quantizers of the edit. Against zero-point rounding at 3 bits on the three models it leaves usable (Qwen2.5-3B's perplexity: $714\times$), GPTQ and AWQ cut the model's in-domain perplexity excess by 48–75% but a single edit's loss by 8–36%; at 4 bits they change it by -14 to $+9\%$ on the Llamas, every interval covering zero (Appendix E). AWQ has the largest single-edit weight channel at 3 bits (≤ 0.35 nats), since re-searching its scale and clip on the edited matrix moves its grid, breaking assumption (i). Batches of 100 behave similarly (share 86.0–102.0%), but Qwen2.5-3B's overloads the matrix (weight channel up to 1.47 nats). Quantizing activations too does not move the damage into the edited weights: W8A8 with SmoothQuant (Xiao et al., 2023) costs a single edit 0.07–0.33 nats, and W4A8 9–14% of its margin, 97–104% of it through the conditioning, which now includes the rounded key (three models; Table 17).

4.3 Proposition 1 on real edits, and the bin-width criterion

With the periodic variance of Lemma 1, the measured readout error over the predicted $1/\mathrm{ESNR}$ (ratio of cell means) is 0.895–1.017 over all 32 MEMIT cells at $n \leq 100$ on the four main models, from $\bar{\delta}/\Delta = 0.033$ to 18.9, and 0.98–1.01 per edit on ZsRE at $n = 1$ (0.86–0.91 at $n = 100$; Appendix B). Those 32 are in sample: the periodic form was adopted on them (§4.4 gives the out-of-sample tests). Calibration degrades at $n = 1000$ (0.54–0.80) and on Qwen2.5-3B (0.89–0.91 at $n = 1$, 0.74–0.84 at $n = 100$). A batch raises most groups' steps, and on GPT-2-XL and Llama-3.2-3B a common lattice for W and $W + \Delta W$ restores 0.97–1.00, except on GPT-2-XL at $n = 1000$, where assumption (ii) fails (Table 9).

Single edits at int3, int4 and NF4 move the entries by 0.02–0.19 of a bin on average on the four main models (MEMIT 0.03–0.12; Appendix B); the bin-width criterion predicts that they are erased, and they keep 0.995–1.001 of their fp margin per cell when only the edited matrix is quantized (Table 7). Nor does it predict which edits fail. Every edit of a batch shares the batch's $\bar{\delta}/\Delta$, so it cannot rank them; over the six single-edit cells with enough failures its AUROC is 0.34–0.64, with three intervals excluding 0.5, two for it and one against; and pooled over editors and batch sizes within a quantizer it points the wrong way (0.12–0.40 on CounterFact except Qwen2.5-3B; Appendix L), because the configurations with the largest $\bar{\delta}/\Delta$ fail most for other reasons.

4.4 Two out-of-sample tests of the predictor

Two out-of-sample tests, the calibration predictions P2 (no numeric tolerance) and P10, were logged before the runs and passed; P1, P8 and P9, logged with them, failed (Appendix Z). **AWQ** with its scale searched once on the unedited matrix changes only the step map in (3); on the four main models and at every batch size its calibration ratio lands in its round-to-nearest partner's band (Figure 3; at $n = 1000$ it shares its partner's miscalibration). The test is mild: AWQ's step differs from rounding's by 9–14% on the most salient 1% of input channels and by 1–2% at the median. **FT-L** is outside the closed-form family: its gradient update is dense, less aligned with the key and 3.3–3.5$\times$ smaller, and at int3, $n = 1$, its calibration ratio is 0.993 and 0.990 on GPT-2-XL and Llama-3.2-3B; the logged prediction that its conditioning share would fall below 90% failed (95.0–100.6%; Appendix R). Under the served quantizers, zero-point rounding is calibrated as well (0.99–1.00 for single edits), while GPTQ and AWQ, which correlate the errors and move the grid, give 0.88–1.25 (Appendix E).

Table 2: **Editing at deployment precision.** Deployed rewrite margin, int3, single edits, 200 probes (a separate run from Figure 1); fp: MEMIT or QAE unquantized; +key (KAE), +value and +both (QAE) solve with k_q^*, v_q^* and both; recovered: share of MEMIT's deployed damage QAE returns (in sample; intervals in Table 22); ΔES: QAE minus MEMIT, paired 95% CI, bold if excluding zero.

model	fp	MEMIT	+key	+value	+both (QAE)	QAE fp	recovered	ΔES
GPT-2-XL	12.84	6.36	9.76	7.96	12.85	6.59	100%	**+0.075** [+0.040, +0.115]
Llama-3.2-3B	13.69	7.06	9.18	9.48	13.20	4.04	93%	**+0.030** [+0.010, +0.055]
GPT-J-6B	14.91	10.19	11.35	12.73	14.46	8.50	90%	+0.010 [−0.010, +0.030]
Llama-3.1-8B	14.80	6.65	9.47	8.76	14.17	3.13	92%	**+0.080** [+0.040, +0.120]
Qwen2.5-3B	14.56	−0.22	0.53	−0.16	6.95	−5.20	49%	**+0.470** [+0.400, +0.540]

4.5 Editing at deployment precision

Table 2 runs QAE and its halves at int3. **Each half returns part of the loss, and together they return most of it.** For single edits the key alone recovers 25–52% of MEMIT's deployed damage and the value alone 25–54%; together they return 90–100% on the four main models (78–91% at $n = 100$) and 108–109% on ZsRE, 60–98% for released editors at $n \leq 1000$ and 77–104% under served 3-bit GPTQ/AWQ (Appendices F, T, E). As their construction implies, each editor's margin is largest in the channel it was solved for (int3, $n \leq 100$: paired intervals exclude zero in 44 of 48 comparisons on the four main models and ZsRE; three of eight invert on Qwen2.5-3B; Figure 2, Appendix F).

Held-out prompts and facts. QAE optimizes the rewrite margin through the network it is evaluated on, so "recovered" is in sample for it. On paraphrases, which name the same subject but never enter its objective, it returns 72–95% of the single-edit loss on the four main models and 66–124% under served 3-bit GPTQ and AWQ on GPT-2-XL and the Llamas (every gain significant; Table 21); net of its neighbourhood cost, 66–92% on the four main models. On 200 CounterFact facts that no earlier single-edit run had scored, with every setting unchanged, it returns 97–103% of the rewrite loss and 83–101% of the paraphrase loss on GPT-2-XL and the two Llamas.

Where it fails. On Qwen2.5-3B, whose symmetric int3 model is itself broken (perplexity 11.0 → 21,654; Appendix V), QAE returns 49% (53% on 100 fresh facts, 11% at $n = 100$) and buys efficacy (0.49 → 0.96) by lowering the neighbourhood score (0.52 → 0.10); its 87% on paraphrases comes with a 4.59-nat neighbourhood loss (15% net). Under the served 3-bit GPTQ and AWQ it returns 77–86% of Qwen2.5-3B's rewrite loss without a significant locality cost, but only 36–47% of its paraphrase loss. At $n = 1000$ on GPT-2-XL QAE returns 50%: quantizing its own matrix on the quantized backbone costs it 2.10 nats, against MEMIT's 0.99 (full minus keys-only; Appendix F). QAE's edit is specific to its precision: unquantized its margin is 21–57% of MEMIT's (negative on Qwen2.5-3B), and a KAE key solved at NF4, int4 or int3 and deployed at another precision is worse than MEMIT on average (Appendix P). Elsewhere locality moves little: at int3, $n = 1$, the neighbourhood margin falls 0.09–0.24 nats but the neighbourhood score by at most 0.03, and perplexity moves by under 0.6% against MEMIT (one edited fact per model, two models; 0.4% and 5.0% at $n = 100$; Appendix V). In the symmetric-grid and GPTQ-backbone runs all 98 significant margin contrasts of QAE and its halves (of 105) survive Benjamini–Hochberg (Appendix L).

Unlearning by the same closed form. With a placeholder suppressing target, int3 costs MEMIT 10.33 and 12.91 of 15.39- and 18.94-nat margins on GPT-2-XL and Llama-3.2-3B; QAE returns 95–103% of the loss for single suppressions and 78–93% at $n = 100$, where it lowers the neighbourhood score by 0.10 and 0.34, and unquantized it suppresses only 0.4–3.9 nats (Appendix I).

Where rounding does erase the update. On 28 released TOFU (Maini et al., 2024) checkpoints of Llama-3.2-1B from eight unlearning methods (Appendix J), an update trained at rate 10^{-5} changes, in bf16, mainly near-zero weights, whose bin positions are not uniform, and int4 rounding delivers 5–29% of it on seven methods (not RMU); from 2×10^{-5} up it delivers 66–102%. A closed-form edit saved in bf16 keeps its margin (within 0.01 nats on two models; Appendix B).

It changes what the model says. A single MEMIT edit deployed at int3 lowers the rate of free continuations asserting the new object by 0.063–0.152 on the four main models (1,200 probes, first LLM judge; the second gives 0.107–0.250; blind annotators see a drop of the same sign), and KAE

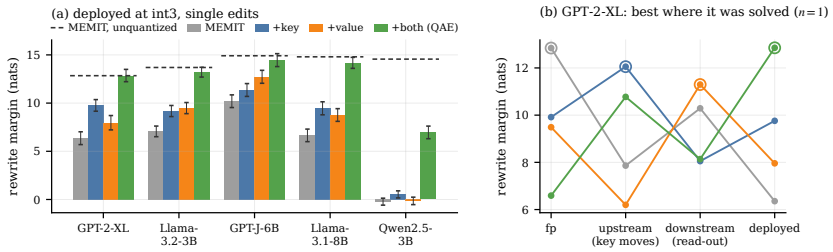


Figure 2: **Solving the edit where it will be read.** (a) Deployed rewrite margin at int3 for single edits, 95% CI; the dashed line is MEMIT on the unquantized model. (b) The four editors of Table 2 read in each channel on GPT-2-XL at $n = 1$, circles marking each one's maximum: each editor's margin is largest in the channel it was solved for (colours as in (a); Appendix F).

recovers 20–61% of it, significantly on GPT-2-XL and Llama-3.1-8B, allowing for an interim 400-probe look (Figure 5, Appendix W). In a 200-probe run QAE returns $+0.20$ on GPT-2-XL against a 0.175 cost (not shown equivalent to the unquantized edit) and nothing significant on Llama-3.2-3B.

What this means in practice. An edit's fp margin overstates what survives quantization, so edits and unlearning updates should be evaluated at the served precision, and closed-form ones solved in the quantized model where it can host them (shown at 3 bits; not Qwen2.5-3B's symmetric int3 model or an overloaded batch), re-solving if that precision changes. Enlarging an edit by post-hoc floors and hinges targets a channel carrying about a tenth or less of a single edit's damage; its gains on GPT-2-XL come by an unmeasured route, reverse on Llama-3.2-3B and cost locality.

5 Limitations

The damage is largest at 3 bits, where the models are themselves degraded (perplexity 1.05–$1.36\times$ under GPTQ and AWQ, 1.2–$2.7\times$ under the main grid's int3 on four models; Qwen2.5-3B's is broken). The conditioning share includes a generic part (33–69% of the keys-only loss after adjusting for margin, 91% on Qwen2.5-3B) and the read-out half, for which §3 gives no closed form. The main grid's unweighted single-layer solve overloads the matrix at $n = 1000$ on GPT-2-XL and GPT-J and in some batches of 100; fp margins reflect v^*'s lack of early stopping; our released-setting MEMIT and AlphaEdit are reimplementations on three models, and MEMIT's reaches only 0.74–0.86 fp efficacy on GPT-2-XL. QAE's gains are established at 3 bits; it is untested when deployed at another quantized precision (its key-aware half loses 0.04–7.2 nats against MEMIT on average there; Appendix P), keeps 21–57% of MEMIT's margin unquantized, fails on Qwen2.5-3B and at $n = 1000$, and costs neighbourhood margin. ESNR (variance form chosen in sample) passes the two logged out-of-sample tests and fails under the served GPTQ and AWQ, and neither it nor the bin-width criterion predicts which single edit will fail (Appendix X). Quantization is simulated (GPTQ, AWQ and SmoothQuant reimplemented; activations quantized on three models); checkpoints stored on the deployment grid, outside Lemma 1, are untested; Instruct models get no chat template; evaluation is single-hop recall on 12–200 probes per cell and LLM judges (binary $\kappa = 0.45$–0.56 against each blind annotator, three-way 0.20–0.44); the served-quantizer and fresh-fact runs omit GPT-J-6B; the gradient-unlearning boundary rests on one 1B model; models are $\leq$ 8B but one (13B), and the data are English.

6 Conclusion

Quantization does not weaken closed-form edits by rounding small updates away: on a fixed, evenly spaced grid whose weights are placed uniformly and independently of the update, rounding dithers it without bias. It changes the conditioning, the key the edit fires at and the read-out it was fitted against, which carries 90–101% of the loss of a single edit of one layer at 3 and 4 bits across the models, datasets and weight-only quantizers we tested (97–104% under W4A8), 33–91% of it generic at int3. Solving the edit in the quantized model returns most of that loss at 3 bits, on held-out paraphrases and fresh facts too, except on Qwen2.5-3B and in an overloaded batch. Edit the model you deploy.

AI USE STATEMENT

In this work, we used generative AI tools to assist in writing, editing, and debugging code, including code used to implement our methods and experiments. We did not use generative AI tools for any of the other tasks that require disclosure: in particular, we did not use them to generate synthetic data (the commercial LLMs that score model outputs as automated judges in Section 4.5 and Appendix W are part of the reported method, not an assistive use), to develop theoretical models or conceptual frameworks, to formulate mathematical claims or assist in proofs, to propose or refine hypotheses, to design the research methodology or experiments, to assist with translation, to clean or reformat datasets, or to support data analysis or the interpretation of results. Additionally, we used generative AI tools to edit the manuscript to improve readability and clarity, and to obtain simulated peer-review-style feedback on drafts, which informed revisions to the writing and presentation. All AI-assisted code was reviewed and tested for correctness by the authors, and all AI-suggested edits and feedback were reviewed and adopted at the authors' discretion; the scientific content, claims, and results are entirely the authors' own. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies how knowledge edits, and unlearning updates produced by the same closed form, survive deployment. The finding that quantization can weaken an unlearning update is already public (Zhang et al., 2025); our contribution is to show that the explanation given for it, the rounding away of sub-bin updates, does not carry over to closed-form edits and to provide a method that makes edits more durable through quantization. That cuts both ways: it helps a closed-form suppression update keep its margin at the served precision, and it would equally help a false or malicious edit survive, so it is a reason to evaluate both at the precision that will be served; at the same time, a QAE unlearning update solved for int3 barely suppresses the fact unquantized and is untested at other quantized precisions, so an audit must test every precision the weights may be served at, the more so since behaviour that differs by precision is what the attack of Egashira et al. (2024) exploits. Apart from two annotators who labelled 240 model continuations to audit the judges (Appendix W), no human participants were involved, and no personal or proprietary data were used; CounterFact (counterfactual statements about real public entities; we release no edited model), ZsRE and TOFU are public benchmarks, WikiText-103 is public text, and every checkpoint (GPT-2, GPT-J, Llama, Qwen, and the released unlearning checkpoints) is used under its licence. External API spend on the judges, including a dropped 342-call pilot of a third judge ($0.01), was $1.52.

REPRODUCIBILITY STATEMENT

Code, the dataset samples with their manifests, every result file (among them 62,280 grid rows, 51,600 ESNR re-prediction rows of Appendix B, 45,000 precision-transfer rows of Appendix P, the AWQ, FT-L, ZsRE, perplexity, GPTQ, multi-layer, released-editor, extra-batch, unlearning and 13B runs, the deployment-precision, served-quantizer, weight-activation, ten-batch, bf16-storage, lattice, $1/\rho$, upstream/downstream-split, key-geometry, magnitude-defence, group-size and gradient-unlearning runs of the appendices, all judge verdicts and the human audit labels), the API cost ledger, and the figure and analysis scripts, which recompute every table and figure from those files, are released in the supplementary material, whose README gives the commands that produce the result files, and will be public on acceptance. Regenerating the result files themselves needs the gated Llama checkpoints, WikiText-103, the checkpoints released by open-unlearning (Dorna et al., 2025), a 24–48 GB GPU and API keys for the two judges (the judge verdicts are reproducible from the released files, not by re-querying the models). Section 4 states the models, layers, quantizer settings, calibration budget, batch sizes and estimator; Appendices A and E give the remaining hyperparameters (among them the NF4 block and the served quantizers' calibration) and the hardware used; Appendix Z lists the predictions of the internal log with their verdicts. Every bootstrap interval in the paper is seeded by a hash of its data and is therefore reproducible from the result files.

REFERENCES

Advik Raj Basani and Anshuman Chhabra. Exposing the illusion of erasure in knowledge editing for LLMs. *arXiv preprint arXiv:2606.23276*, 2026.

Yoav Benjamini and Yosef Hochberg. Controlling the false discovery rate: A practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society: Series B (Methodological)*, 57(1):289–300, 1995.

Jacob Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46, 1960.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, 2022.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, 2023.

Yijiang River Dong, Hongzhou Lin, Mikhail Belkin, Ramon Huerta, and Ivan Vulić. UNDIAL: Self-distillation with adjusted logits for robust unlearning in large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 8827–8840, 2025.

Vineeth Dorna, Anmol Mekala, Wenlong Zhao, Andrew McCallum, J. Zico Kolter, Zachary C. Lipton, and Pratyush Maini. OpenUnlearning: Accelerating LLM unlearning via unified benchmarking of methods and metrics. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, 2025.

Evan Duan. Perplexity can miss SAE feature damage under quantization. *arXiv preprint arXiv:2606.03002*, 2026.

Bradley Efron. Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, 7(1): 1–26, 1979.

Kazuki Egashira, Mark Vero, Robin Staab, Jingxuan He, and Martin Vechev. Exploiting LLM quantization. In *Advances in Neural Information Processing Systems*, 2024.

Chongyu Fan, Jiancheng Liu, Licong Lin, Jinghan Jia, Ruiqi Zhang, Song Mei, and Sijia Liu. Simplicity prevails: Rethinking negative preference optimization for LLM unlearning. In *Advances in Neural Information Processing Systems*, 2025.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. AlphaEdit: Null-space constrained knowledge editing for language models. In *International Conference on Learning Representations*, 2025.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations*, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Robert M. Gray and David L. Neuhoff. Quantization. *IEEE Transactions on Information Theory*, 44 (6):2325–2383, 1998.

Akshat Gupta, Sidharth Baskaran, and Gopala Anumanchipalli. Rebuilding ROME: Resolving model collapse during sequential model editing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 21738–21744, 2024.

Suyog Gupta, Ankur Agrawal, Kailash Gopalakrishnan, and Pritish Narayanan. Deep learning with limited numerical precision. In *International Conference on Machine Learning*, pp. 1737–1746, 2015.

Ajay Jaiswal, Zhe Gan, Xianzhi Du, Bowen Zhang, Zhangyang Wang, and Yinfei Yang. Compressing LLMs: The truth is rarely pure and never simple. In *International Conference on Learning Representations*, 2024.

Abdullah Ahmad Khan and Ferdous Sohel. DurableUn: Quantization-induced recovery attacks in machine unlearning. *arXiv preprint arXiv:2605.02196*, 2026.

Arinbjörn Kolbeinsson, Kyle O'Brien, Tianjin Huang, Shanghua Gao, Shiwei Liu, Jonathan Richard Schwarz, Anurag Vaidya, Faisal Mahmood, Marinka Zitnik, Tianlong Chen, and Thomas Hartvigsen. Composable interventions for language models. In *International Conference on Learning Representations*, 2025.

Anubhav Lakra and Yue Feng. CACHE-UK: A stability-aware memory editor for sequentially updated quantized LLMs in finance. *arXiv preprint arXiv:2607.28292*, 2026.

Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-shot relation extraction via reading comprehension. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pp. 333–342, 2017.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, et al. The WMDP benchmark: Measuring and reducing malicious use with unlearning. In *International Conference on Machine Learning*, pp. 28525–28550, 2024.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, 2024.

Zhenyan Lu, Daliang Xu, Dongqi Cai, Zexi Li, Wei Liu, Fangming Liu, Shangguang Wang, and Mengwei Xu. MobiEdit: Resource-efficient knowledge editing for personalized on-device LLMs. In *International Conference on Learning Representations*, 2026.

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.

Anmol Mekala, Vineeth Dorna, Shreya Dubey, Abhishek Lalwani, David Koleczek, Mukund Rungta, Sadid Hasan, and Elita Lobo. Alternate preference optimization for unlearning factual knowledge in large language models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 3732–3752, 2025.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, 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.

Himanshu Mishra and Kanwal Mehreen. QUAIL: Quantization aware unlearning for mitigating misinformation in LLMs. *arXiv preprint arXiv:2601.15538*, 2026.

Satya Sai Srinath Namburi, Makesh Sreedhar, Srinath Srinivasan, and Frederic Sala. The cost of compression: Investigating the impact of compression on parametric knowledge in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

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. Technical report, OpenAI, 2019.

Saisab Sadhu, Pratinav Seth, and Vinay Kumar Sankarapu. Forgetting that sticks: Quantization-permanent unlearning via circuit attribution. *arXiv preprint arXiv:2605.15138*, 2026.

Leonard Schuchman. Dither signals and their effect on quantization noise. *IEEE Transactions on Communication Technology*, 12(4):162–165, 1964.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. Massive activations in large language models. In *Conference on Language Modeling*, 2024.

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

Ben Wang and Aran Komatsuzaki. GPT-J-6B: A 6 billion parameter autoregressive language model. GitHub repository, https://github.com/kingoflolz/mesh-transformer-jax, 2021.

Qianli Wang, Mingyang Wang, Nils Feldhus, Simon Ostermann, Yuan Cao, Hinrich Schütze, Sebastian Möller, and Vera Schmitt. Through a compressed lens: Investigating the impact of quantization on factual knowledge recall. In *Proceedings of the 6th Workshop on Trustworthy NLP (TrustNLP 2026)*, pp. 21–39, 2026.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pp. 38087–38099, 2023.

Jianhao Yan, Futing Wang, Yun Luo, Yafu Li, and Yue Zhang. Keys to robust edits: From theoretical insights to practical advances. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 22545–22560, 2025.

Wanli Yang, Fei Sun, Xinyu Ma, Xun Liu, Dawei Yin, and Xueqi Cheng. The butterfly effect of model editing: Few edits can trigger large language models collapse. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 5419–5437, 2024.

Mengxia Yu, De Wang, Qi Shan, Colorado Reed, and Alvin Wan. The super weight in large language models. *arXiv preprint arXiv:2411.07191*, 2024.

Pedram Zamirai, Jian Zhang, Christopher R. Aberger, and Christopher De Sa. Revisiting BFloat16 training. *arXiv preprint arXiv:2010.06192*, 2020.

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

Zhiwei Zhang, Fali Wang, Xiaomin Li, Zongyu Wu, Xianfeng Tang, Hui Liu, Qi He, Wenpeng Yin, and Suhang Wang. Catastrophic failure of LLM unlearning via quantization. In *International Conference on Learning Representations*, 2025.

Chenxi Zhou, Pengfei Cao, Jinyu Ye, Bohan Yu, Haida Yu, Jiang Li, Jun Zhao, and Kang Liu. Quantization degradation in large language models: A signal-noise perspective. *arXiv preprint arXiv:2608.08188*, 2026.

Chen Zhu, Ankit Singh Rawat, Manzil Zaheer, Srinadh Bhojanapalli, Daliang Li, Felix Yu, and Sanjiv Kumar. Modifying memories in transformer models. *arXiv preprint arXiv:2012.00363*, 2020.

GUIDE TO THE APPENDICES

Setup: A (details and the definition of every quantity). *Channel decomposition:* D (every int3, int4 and NF4 MEMIT main-grid cell, extra batches of 100, a 13B model), E (the quantizers that are served), G (upstream against downstream), H (edits against facts the model knows), M (group size), S (bands of layers), T (MEMIT and AlphaEdit with released settings). *The weight channel and Proposition 1:* B (Lemma 1's variance and assumptions, bf16 storage, the lattice on real matrices), C (Corollary 2), N (alignment and magnitude), Q (AWQ), R (FT-L), Y (GPTQ on the edited matrix and as a backbone), X (per-edit prediction). *The key channel:* U (Proposition 2, the $1/\rho$ rescaling). *The fix:* F (QAE on the main grid; served quantizers in E), O (neighbourhood), P (precision transfer), V (perplexity). *Unlearning:* I (closed form), J (gradient-based); K (magnitude defences, applied to edits). *Behaviour and statistics:* W (generation, judges, human audit), L (multiplicity, the bin-width criterion's AUROC), Z (predictions logged before the runs).

A EXPERIMENTAL DETAILS

Editing. The edited down-projection is the one ROME's released settings edit on GPT-2-medium (layer 8), GPT-2-XL (17) and GPT-J (5), the top layer of the MEMIT band on GPT-2-XL (13–17) and inside it on GPT-J (3–8); on GPT-2, Llama-3.2-3B, Qwen2.5-3B and Llama-3.1-8B it is an early-middle layer (5, 4, 5, 5), on Llama-3.1-8B inside the band (4–8) of MEMIT's and AlphaEdit's released settings (Appendix T). For every fact the target value v^* is obtained as in ROME (Meng et al., 2022), with fixed rather than sampled context prefixes: a change δ, starting at zero, is optimized so that $Wk^* + \delta$, substituted for the edited down-projection's output at the subject's last token (the last token whose character span overlaps the subject's first occurrence) of every prompt, maximizes the probability of the new object under the rewrite prompt, with 25 Adam steps (PyTorch's defaults apart from the learning rate, 0.5 and constant; δ kept in fp32 and cast to the backbone's format in each forward pass), four context prefixes (none, "The", "Recently," and "In fact,"), a KL "essence" term of weight 0.0625 and a norm clamp of $4\times$ the original value's norm. The KL term compares the next-token distribution after "{subject} is a" with the model's own before the edit; the objective is the new object's mean per-token negative log-probability (the object tokenized with a leading space), averaged over the four prefixes; unlike ROME's released code, there is no weight decay on δ and no early stopping, so the fp margins (12.84–14.91 nats for single edits on the five main models) reflect this choice and a loss stated as a share of the margin depends on it (the tables also give it in nats); and k^* is read once, at the subject's last token of the unprefixed rewrite prompt, not averaged over prefixes. The batch solve is $\Delta W = RK^\top (C + KK^\top + \lambda I)^{-1}$ with $\lambda = 10^{-3}\bar{\lambda}$ for MEMIT, KAE, KRE, KAE+KRE, QAE and value-aware editing and $\lambda = 3\bar{\lambda}$ for the alignment ridge, $\bar{\lambda}$ being the mean diagonal of C; for $n \le 64$ the inverse is applied through the Woodbury identity on a cached Cholesky factor of $C + \lambda I$, and at $n = 1$ it delivers $a\, s/(1+s)$ with $s = {k^*}^\top (C + \lambda I)^{-1} k^* \gg 1$, i.e. the closed form (1) up to a relative $1/(1+s)$, with $(C + \lambda I)^{-1}$ in place of C^{-1} (at $n = 1$ the fp margins of MEMIT, the ridge and the AlphaEdit-style variant agree to within 0.14 nats on the five main models). C is the unweighted second moment $\mathbb{E}[kk^\top]$; published MEMIT multiplies it by a covariance weight of order 10^4 and spreads each residual over a band of layers, so a single-layer edit here delivers its whole residual and a 1000-fact batch is a far larger perturbation of one matrix than published MEMIT ever makes ($\|\Delta W\|_F / \|W\|_F$ of 11.6 and 7.5 on GPT-2-XL and GPT-J, fp perplexity $5.4\times$ and $6.5\times$ the unedited value, against 2.2 and 1.7, and $1.27\times$ and $1.13\times$, on the two Llamas; Appendix V). Every $n = 1000$ statement about the main grid is a statement about that regime, in which editing itself degrades the model, the batch analogue of the collapse Yang et al. (2024) report for single edits; MEMIT and AlphaEdit as released stay out of it (Appendix T). We call a batch *overloaded* when quantizing the edited matrix alone, on the unquantized backbone, costs a visible share of its deployed loss (D_W / D_F): 42% and 37% of the int3 loss on GPT-2-XL and GPT-J at $n = 1000$, 12% on Qwen2.5-3B at $n = 100$ and 22–24% in batch 1 of GPT-2-XL and GPT-J (Table 12), against 4.4% and 1.7% on the Llamas at $n = 1000$ (GPT-2 at $n = 150$, at 10%, is borderline). The label names where the weight channel is large rather than predicting it, and the norm ratio alone does not decide it (Qwen2.5-3B's update is 2.1 times the norm); the weights-only damage itself, in nats (Table 11), needs no quantized backbone and can be measured before deployment, but its share of the deployed loss needs the quantized model, and near 10% the share is not stable between runs (Qwen2.5-3B's 12% is 9.8%, 1.49 of 15.21 nats, in the re-run of Table 10 on the same 100 probes). The AlphaEdit-style variant projects K onto the eigen-subspace of C with the lowest half of the spectrum and solves the projected system with an absolute damping $10^{-2} I$ added to the projected C; the published method projects onto the null space of the preserved keys and edits several layers, so it is a single-layer variant in the same spirit; the published method itself is run in Appendix T. KAE reads the keys through a backbone quantized with the deployment quantizer and solves the same system with K_q and residual $V^* - WK_q$ (V^* the stacked targets v^*); KRE augments K with four copies perturbed by isotropic Gaussian noise of relative radius 0.30 (per-entry scale $0.30\|k\|/\sqrt{d_{\text{in}}}$; a torch generator seeded with the fact's identifier for a single edit and with n for a batch), each paired with the same target; it is a control for isotropic key robustness, below the int3 key displacement (0.59–1.68) and inside the int4 and NF4 range (0.22–0.46), uncorrelated with it, and not the key-robust method of Yan et al. (2025), which we do not run. FT-L runs Adam (learning rate 5×10^{-4}) for 25 steps on the single edited matrix, minimizing the new object's mean per-token negative log-probability after the unprefixed rewrite prompt (no KL term; averaged over the facts of a batch), and after every step clamps the update entrywise to an L_∞ trust region of 5×10^{-4} times the matrix's largest $|W_{ij}|$. For a dense update, u is defined as the top right-singular vector of ΔW; on rank-one updates this reproduces MEMIT's u to $1 - |\cos| < 10^{-6}$ over 20 updates, and the

randomized estimate (`svd_lowrank`, rank 8, 4 power iterations) agrees with the exact top vector to $1 - |\cos| < 10^{-6}$ on every probe where both are computed. Because the sign of a singular vector is arbitrary, $|\cos(u, k^*)|$ is reported for FT-L; its effective rank is $(\sum_i \sigma_i)^2 / \sum_i \sigma_i^2$ over the update's singular values, averaged over the first ten probes.

Quantization. All quantizers of the main grid (round-to-nearest and NF4) and of the separate AWQ and edited-matrix GPTQ runs (Appendices Q and Y) act group-wise with 64 consecutive input entries of a row sharing a scale (bitsandbytes' block size; GPTQ and AWQ releases commonly use 128, and Appendix M sweeps the size). Round-to-nearest uses symmetric absmax scales with $2^{b-1} - 1$ levels on each side of zero ($\Delta = \text{absmax}/(2^{b-1} - 1)$; $2^b - 1$ levels in all, 7, 15 and 255 at 3, 4 and 8 bits, one fewer than the zero-point grid of Appendix E); NF4 normalizes each group by its absmax, kept in full precision (the absmax constants are not double-quantized as in QLoRA), onto the 16-level NormalFloat codebook (Dettmers et al., 2023), and its per-entry step is the width of the rounding cell of the edited entry's ($W_{ij} + \delta_{ij}$'s) nearest codebook value (half the gap between that value's two neighbours; for the end values ± 1, half the gap to their one neighbour) times the group absmax. GPTQ uses the second moment C already estimated at the edited layer's input as its Hessian, damping 1%, a lazy-update block of 128 columns (not to be confused with the 64-entry quantization group), the columns in their natural order (no act-order) with each group's absmax set when the loop reaches it (no static groups), and is applied to the edited matrix only (the GPTQ backbone of Appendix Y is described there). AWQ searches the scale exponent α over a 21-point grid $(0, 0.05, \ldots, 1$; the served AWQ of Appendix E uses AutoAWQ's 20) with the $\sqrt{\max \cdot \min}$ normalization of Lin et al. (2024) on 500,000 tokens of activation statistics, without the clipping search that ships with AutoAWQ, so it is a slightly weaker AWQ than a serving stack's; the search is run per linear matrix (not per group of linears sharing an input) and minimizes that matrix's output error $\mathbb{E}\|(\mathcal{Q}(W \operatorname{diag} s) \operatorname{diag}(s)^{-1} - W)x\|^2$ with $\mathbb{E}[xx^\top]$ replaced by its diagonal; $\alpha = 0$ reproduces round-to-nearest bit-exactly. The scale vector is searched once on the unedited matrix and reused for $W + \Delta W$ while the group absmax is re-fitted; re-running the full search on 40 edited matrices (two models, two bit widths) moved the chosen α on none of them. Quantization is simulated and weight-only: every linear layer of the transformer blocks is fake-quantized in place (in fp32 with fp32 scales, then cast back to the backbone's 16-bit format; GPT-2's Conv1D weights are transposed first, so that groups always run along the input dimension; embeddings and output head stay in 16 bits; activations are not quantized except in the W8A8 and W4A8 runs of Appendix E), the backbone is held in 16 bits on the GPU (fp16 for GPT-2-XL, GPT-J and Llama-2-13B; bf16 for the Llama-3 and Qwen2.5 checkpoints; GPT-2 and GPT-2-medium ran on CPU in fp32) and the edited matrix, upcast from the loaded weights, is held in fp32 in every channel, so that 16-bit mantissa rounding cannot masquerade as the weight channel. An edited checkpoint saved in bf16 before quantization, the storage path that Appendix J shows can correlate a small update with the grid, is simulated separately in Appendix B. The zero-point, group-128 GPTQ and AWQ of §4.2 are described in Appendix E.

Evaluation. The rewrite margin is the difference of length-normalized log-probabilities of the new and true objects after the rewrite prompt; paraphrase and neighbourhood margins use the first two paraphrase prompts and the first three neighbourhood prompts of each CounterFact record (in the dataset's order) and ZsRE's one rephrase and one locality question; ZsRE's rewrite prompt is its question with the subject in place (no question–answer template), and its locality margin is $\log P(\text{locality answer}) - \log P(o_{\text{new}})$ on the locality question (not ZsRE's usual locality metric, agreement with the unedited model's prediction). ZsRE's locality probe is an unrelated question with its own answer, so those records carry per-neighbour targets; that code path is inert on CounterFact and a test in the repository checks that it produces identical probe lists on all 1,600 records and bit-identical outputs on real forward passes. The ZsRE sample is drawn from the 19,086-record release used by ROME (`zsre_mend_eval.json`, MD5 `41a1e721755a4c02fdb7629658476c93`); 716 records are dropped (639 where the new and true answers coincide, 77 with an empty field) with the reasons counted in a manifest, the remaining 18,370 are shuffled with Python's `random.Random(0)` and the first 1,600 kept, and two generic generation prompts ("{subject} is known for" and "The story of {subject} begins") are synthesized per subject because ZsRE ships none. The CounterFact sample is a random draw of 1,600 of its 21,919 records (Python's `random.Random(20260821).sample` of the record indices; the manifest lists them), and a generation probe is one fact, continued from the first of its generation prompts; like the margin probes, a run's generation probes are the lowest identifiers among the facts whose key and v^* were computed (1,200 per model for single edits, 400 on ZsRE and 200 for QAE, so they include the margin probes, and the 300 lowest of the 1,000-fact batch for the batch panel). Free continuations are scored by an

LLM judge (`gpt-4o-mini`) asked whether the continuation asserts the new object, the true object, or neither (the new object is always candidate A; the order is not counterbalanced, so a position bias would raise every A-rate and cancels from a contrast only insofar as it is equal in both arms); the same prompt was re-run through a second judge from a different provider (`deepseek-chat`) on byte-identical text. The released cost ledger, which records the model alias requested and not the version that answered, also records 342 calls of a `gemini-2.5-flash` pilot judge that no reported number uses ($0.01 of the $1.52 spent); it was dropped when its API endpoint rate-limited it. The readout error is $\|(\mathcal{Q}(W+\Delta W)-\mathcal{Q}(W)-\Delta W)k^*\|/\|\Delta W k^*\|$, and ESNR is computed from the update, the key and the quantizer's step map with no forward pass; the readout error itself costs the same to compute, so ESNR's use is that it is a closed form in the update's displacements, which Corollary 2 reduces to the alignment, not that it is cheaper.

Implementation details. C is the uncentred second moment of the edited layer's MLP key, accumulated in float64 over every non-padding position of WikiText-103 *train* paragraphs (the `wikitext-103-raw-v1` configuration, as for calibration and perplexity) of at least 100 characters, taken in the dataset's order (no sampling), in batches of eight truncated to 128 tokens, until 500,000 tokens (100,000 for GPT-2 and GPT-2-medium, which ran on CPU); nothing is shrunk (neither shrinkage, a larger sample nor the quantized backbone's own C was tested), and KAE and QAE reuse the fp C. Except in the ZsRE deployment-precision runs (*Independent solves* below), the probes of a cell are the 200 smallest `case_id` values among the facts of the 1,600-record sample whose key and v^* were computed for that model (on the five main models' CounterFact grids, the sample's first 200 records, the sample file being stored in `case_id` order; on GPT-2 and GPT-2-medium, the lowest of a 220-record subset of the sample used in the pilot runs (the first 220 records of the same seeded draw), listed in `data/counterfact_1pct_manifest.json`), and at $n \in \{100, 1000\}$ one batch of the n lowest identifiers is solved, so every batch contains the probes it is scored on (the $n = 100$ cells score the first 100); at $n = 1$ each probe is edited alone. Taking the lowest identifiers does not skew the relation mix: the 200 probes cover 32 of the sample's 34 relations in proportions consistent with the other 1,400 records (χ^2 test, $p = 0.52$), and the 1,200 generation probes likewise ($p = 0.58$). The metrics are not CounterFact's original protocol (we use two paraphrase and three neighbourhood prompts and length-normalized log-probabilities), so absolute ES, PS and NS are not comparable with published tables. Prompts are fed to the Instruct checkpoints as raw text, without a chat template, and are tokenized with each tokenizer's default special tokens (a BOS token on the Llama models, none on GPT-2, GPT-J and Qwen2.5); a batch of prompts is right-padded with an attention mask. Perplexity is measured on non-overlapping 512-token windows of the WikiText-103 *test* split (the first 99,840 tokens of its non-empty lines, each line tokenized on its own and the tokens concatenated in order, 195 windows; every run takes its windows from the start of the split; the served-quantizer and released-editor runs use 20,000 and Appendix E's diagnostic 8,000, as stated there, rounded down to whole windows: 39 and 15 windows, 19,968 and 7,680 tokens), disjoint from the text C is estimated on. The judges are called at temperature 0 with at most four output tokens and the first non-space character of the reply, upper-cased, is read as A (new object), B (true object) or N (neither), any other character counting as not A (all 56,744 verdicts in the result files are A, B or N); the prompt gives the subject, the relation template, both candidates and the continuation, and asks which candidate the continuation asserts or implies (verbatim from `code/gen_judge.py`, line breaks omitted, between the quotes: "`A language model was asked to continue a prompt about **{subject}**. The relevant fact template is: "{template}" Candidate A: {new object} Candidate B: {true object} Continuation: """{continuation}""" Which candidate does the continuation assert or imply for that fact? Answer with exactly one token: A, B, or N (neither/unclear).`"; both judges are called by their API aliases, with no snapshot pinned, other parameters at the provider's defaults and up to six attempts per call, a call being repeated only when the API returned an error, with exponential back-off, and the first reply received kept as the verdict). Continuations are greedy, at most 36 new tokens (generation stops at the model's end-of-sequence token, and special tokens are stripped from the decoded text), generated one prompt at a time without padding. The bootstrap stream is seeded by an 8-byte BLAKE2b digest of the float64 data rounded to 12 decimals and drawn with NumPy's default generator. Table 3 fixes the vocabulary.

Statistics. Unless a caption says otherwise, every interval is a 95% percentile bootstrap (Efron, 1979; 20,000 resamples) of the mean of a paired per-probe difference, seeded by a hash of the data so that it is reproducible bit for bit from the result files. A contrast and its negation hash to different streams, so

Table 3: Quantities used throughout. All ratios are ratios of cell means unless stated.

term	definition
rewrite margin m	$\log P(o_{\text{new}} \mid p) - \log P(o_{\text{true}} \mid p)$, length-normalized, nats
damage (loss)	fp margin minus channel margin, paired per probe; positive when quantization hurts. Tables that print margin *changes* print its negative
channel margins	fp, weights-only (edited matrix quantized; its damage includes the rounding of W itself, $\mathcal{Q}(W) - W$, which the unedited model shares, on top of the rounding error on the update, $\tilde{E}$), keys-only (everything else quantized), full; keys-only splits into upstream (matrices that feed k^*) and downstream (matrices after the edited layer)
conditioning share	keys-only damage D_K over full damage D_F (printed K/F; D_W is the weights-only damage), for one editor against its own fp margin; exceeds 100% when $D_F < D_K$, i.e. when quantizing the edited matrix on top of the quantized backbone raises the margin. It is the share of the whole conditioning, key and read-out together
even split	$[D_K + (D_F - D_W)]/(2D_F)$: each channel receives half of the interaction (Table 11)
up / down share	upstream (downstream) damage over keys-only damage, same editor; the damage is sub-additive (upstream plus downstream exceeds keys-only) in every MEMIT cell of Tables 24–26 except GPT-J-6B at int3, $n = 1$ ($43\% + 56\%$, unrounded sum 99.0%) and Llama-3.1-8B at int4 (99.9% at $n = 1$, 92.1% at $n = 100$), so the shares usually sum past 100%
deployed damage vs fp MEMIT	MEMIT's fp margin minus an editor's full-channel margin (Tables 2, 19, 20)
recovered	$[D(\text{MEMIT}) - D(\text{editor})]/D(\text{MEMIT})$ with D the deployed damage vs fp MEMIT; above 100% when the editor's deployed margin exceeds MEMIT's fp margin (Table 22 gives both)
retention (Table 5)	mean per-row ratio of weights-only to fp margin, rows with fp margin above 0.5 nats
calibration ratio	measured readout error over $1/\text{ESNR}$: ratio of cell means (Table 4, §4.3) or mean per-edit ratio where so labelled
$\cos(u, k^*)$	for a batch, per fact with its own $u_i = (C + KK^\top + \lambda I)^{-1} k_i$, averaged over the probes of the cell
$\bar{\delta}/\Delta$	mean $\lvert\Delta W_{ij}\rvert$ over the mean step of the deployed quantizer on $W + \Delta W$, the step ESNR uses (Corollary 2, Table 4); Table 45 (multi-layer bands) uses the mean per-entry ratio; a row is *sub-bin* when it is below 1 (a criterion on the mean, not on every entry)
readout error	error of the edited layer's own output at its key, a weight-channel quantity (the *read-out*, hyphenated, is the layers above that turn that output into a prediction): $\lVert(\mathcal{Q}(W+\Delta W) - \mathcal{Q}(W) - \Delta W)k^*\rVert / \lVert\Delta W k^*\rVert$, the relative error that $1/\text{ESNR}$ predicts (Proposition 1); KSNR and ρ are defined in §3.2 and Appendix U
ES, PS, NS	efficacy $\mathbf{1}[m > 0]$; paraphrase success, the same on the mean margin of two paraphrases (one on ZsRE); neighbourhood success, $\mathbf{1}$[mean over three neighbouring subjects of $\log P(o_{\text{true}}) - \log P(o_{\text{new}}) > 0$; on ZsRE, $\log P(\text{locality answer}) - \log P(o_{\text{new}}) > 0$ on its one locality question]

where two tables print the same paired contrast with opposite signs (Tables 22 and 29) their bounds can differ by a few hundredths. Cells have 200 probes at $n \in \{1, 1000\}$ and 100 at $n = 100$ on the main models (100 at $n = 1$ in Appendices M and S), 40 on GPT-2 (100 in Appendix P) and 20 on GPT-2-medium (30 in its multi-layer run), the first 160 or 200 TOFU forget10 questions in Appendix J, 12–25 for the GPTQ sweep of Appendix Y, 1,200 for the single-edit generation runs, 300 for the batch ones, 400 for the ZsRE ones and 200 for the QAE generation run; percentile intervals on cells of 12–40 probes (GPT-2, GPT-2-medium, the GPTQ sweep) may under-cover and are approximate. Batch sizes are $n \in \{1, 100, 1000\}$ on the four main models, $\{1, 100\}$ on Qwen2.5-3B and on ZsRE, $\{1, 150\}$ on GPT-2 and $\{1, 30\}$ on GPT-2-medium; a batch cell is a single solve on one batch (the n lowest identifiers), so its interval covers variation over probe facts and not over batch composition; Appendix D adds further disjoint batches at $n = 100$ (ten on GPT-2-XL, Llama-3.2-3B and Llama-3.1-8B, three on GPT-J-6B). The 12,000 rows per main model are 6 editors $\times$ 4 quantizers $\times$ $(200 + 100 + 200)$ probe slots over the three batch sizes, drawn from 200 distinct facts, each row carrying all four channels; Qwen2.5-3B has $6 \times 4 \times (200 + 100) = 7{,}200$, each of the two ZsRE grids (GPT-2-XL, Llama-3.2-3B) 3 editors (MEMIT, KAE, the ridge) $\times$ 3 quantizers

(int3, int4, NF4) $\times$ $(200 + 100) = 2{,}700$, GPT-2 $6 \times 3 \times 2 \times 40 = 1{,}440$ and GPT-2-medium $3 \times 2 \times 2 \times 20 = 240$ (62,280 in all). The 51,600 ESNR re-prediction rows of Appendix B are the 12,000 of each of the four main models, Qwen2.5-3B's $4 \times (200 + 100) = 1{,}200$ MEMIT rows, GPT-2's $6 \times 4 \times 2 \times 40 = 1{,}920$ and GPT-2-medium's $3 \times 4 \times 2 \times 20 = 480$ (these include int8, which neither small model's channel grid runs, and on GPT-2-medium NF4, which its grid does not run either: a re-prediction row rebuilds the update and measures its readout error without a forward pass); the 45,000 precision-transfer rows of Appendix P are 5 solve precisions $\times$ 5 deployment precisions $\times$ 2 batch sizes $\times$ 200 probes on each of the four main models and $\times$ 100 on GPT-2. Because a fact recurs across editors and quantizers, the pooled sub-bin retention of Table 5 is also recomputed with a bootstrap that resamples *facts* rather than rows: the widest interval grows by $1.7\times$ and none grows enough to move a conclusion (`code/subbin_cluster.py`). The exceptions to the paired per-cell bootstrap are named where they occur: ratios (conditioning share, even split, "recovered") resample the probes jointly, the margin-matched control of Appendix H is a two-sample bootstrap, the human-audit contrast of Appendix W is unpaired, the AUROC intervals of Appendix X are bootstraps of the AUROC that resample failures and successes separately (2,000 resamples) and those of Appendix W resample probes (4,000 resamples, dropping any resample with a single class), and the equivalence and two-look intervals of Appendix W are at 90% and 97.5%. Rounded bounds and means keep their sign: -0.00 is below zero, $+0.00$ at or above it. An interval excludes zero (bold) when its unrounded lower bound is above zero or its upper bound below it, so a bound of exactly zero does not.

Independent solves across experiments. The channel grid, the upstream/downstream split and the deployment-precision experiment are separate runs that each re-estimate C from the same first 500,000 tokens of WikiText-103 and re-solve the editor; the solve is reproducible only up to floating-point summation order (GPU type and kernels, C accumulated on the GPU or the CPU, batched or sequential v^* optimization; `code/editing.py`), so the same cell can differ slightly between tables. On GPT-2-XL the three agree to 0.01 nats (fp margin $12.84/12.84/12.84$ at $n = 1$, $13.12/13.12/13.13$ at $n = 100$); on Llama-3.2-3B they differ by at most 0.21 nats ($13.73/13.72/13.69$ and $13.76/13.75/13.55$), about 1.5% of the margin and inside the interval of every quantity they are compared on (the full-channel loss at $n = 1$ is 6.75 $[6.23, 7.26]$ in the grid and 6.63 $[6.14, 7.13]$ in the deployment-precision run). The precision-transfer run (Appendix P), the AWQ run (Appendix Q), the group-size, multi-layer and GPTQ-backbone runs (Appendices M, S and Y; Llama-3.2-3B's fp margin on the first 100 single-edit probes is 13.85 there against 14.02 in the grid) and the two perplexity runs (Appendix V) are likewise separate solves, which is why the same nominal cell can differ between tables, usually by a few hundredths of a nat and by up to 0.2 nats in the fp margin (0.3 where a key/value cache, of the facts' k^* and v^* (not the attention cache), was rebuilt, and up to 1.05 between the first 100 single-edit probes of Table 15 and the 200 of Table 11, a difference of probe sets: on the same 100 probes the two runs agree to within 0.27 nats). Damages can differ by more than fp margins. Qwen2.5-3B varies most. Between its grid and the bf16-storage run of Table 10, the residual norms agree to 0.9% on the same 100 probes and the cell-mean fp margins to 0.17 nats. Even so, the per-probe fp margins differ by a median of 1.0–1.2 nats (Pearson 0.92–0.93), and the int3 weights-only damage is 0.11 against 0.21 nats at $n = 1$ and 1.81 against 1.49 at $n = 100$, a difference we have not traced. On Llama-3.2-3B the same two runs agree to 0.03 nats in every such cell. On ZsRE the deployment-precision runs (Tables 18, 20, 21 and 22) draw their probes from the first 200 records of the shuffled sample, whereas the channel grid scores the 200 smallest identifiers among the records whose key and v^* it computed; the two probe sets share 60 facts at $n = 1$ and 29 at $n = 100$, on which MEMIT's fp margins agree to 0.00 nats at $n = 1$ and to within 0.16 at $n = 100$. No paired statistic is computed across two experiments; where the text sets two runs side by side (the upstream share of Appendix G next to the recoveries of Appendix F, say) the comparison is qualitative.

Compute. All GPU experiments ran on a single GPU, a 24 GB or a 48 GB consumer GPU, shared with unrelated jobs; GPT-2 and GPT-2-medium ran on CPU. GPU-hours were not logged per experiment, and no deterministic-algorithm flags were set. The production-quantizer runs, the extra batches and the checks of Appendices E, D (ten batches), B (bf16 storage, real-matrix lattice) and U ($1/\rho$) rebuilt the key/value caches on the 24 GB card with the same code and data. External API spend for judging was $1.52. The checkpoints are `gpt2`, `gpt2-medium`, `gpt2-xl` (Radford et al., 2019), `EleutherAI/gpt-j-6B` (Wang & Komatsuzaki, 2021), `meta-llama/Llama-3.2-3B-Instruct`, `meta-llama/Llama-3.1-8B-Instruct` (Grattafiori et al., 2024), `Qwen/Qwen2.5-3B-Instruct` (Qwen Team, 2024) and, for Ap-

pendix D, the base model `meta-llama/Llama-2-13b-hf` (Touvron et al., 2023); Appendix J uses the TOFU target model and 29 unlearned checkpoints (the 28 of Table 32 and GradDiff at 10^{-5} with $\alpha = 5$, Table 30) released by open-unlearning (Dorna et al., 2025); for the runs of Appendices T, D and Y the SHA-256 of every weight file is released (`results/model_revisions.txt`; GPT-2-XL and GPT-J match the Hugging Face commits `15ea56de` and `47e16930`, and the two Llama-3 models' shards have the SHA-256 of the public mirrors `unsloth/Llama-3.2-3B-Instruct` and `unsloth/Meta-Llama-3.1-8B-Instruct` of the official release, whose gated repositories do not expose theirs), with PyTorch 2.14 and Transformers 5.17 (versions were not recorded for the earlier runs). In those runs several jobs shared the card as MPS clients and the per-fact value optimizations of one model ran as workers that map a single copy of its weights, which is bit-identical to a single process on the same facts; on GPT-J and Llama-3.1-8B the published-editor runs pass the four prefixed prompts and the KL prompt of each optimization step through one padded batch rather than five sequential passes, the same objective, whose optimum on the norm clamp is reached at a slightly different point (cosine 0.96 on average to the value change $v^* - Wk^*$ found with sequential passes, over eight Llama-3.1-8B facts, equal norm and NLL; `code/vstar_batched_check.py`), used for MEMIT and QAE alike.

Further limitations, collected from the appendices. Every loss quoted as a share of the margin depends on the fp margin that the value optimization reaches (no weight decay or early stopping, above). The independence half of assumption (ii) is checked only through the real edits' readout error and retention (Appendix B). The judges' candidate order is not counterbalanced and their model versions are not pinned. Separate runs of the same cell can differ by up to 0.32 nats in a damage, on Qwen2.5-3B (above). QAE needs the full-precision weights and C, the exact deployment quantizer and a fake-quantized backbone to optimize v_q^* through. It keeps the residual $v_q^* - Wk_q^*$ with the unquantized W and the full-precision C, and neither $\mathcal{Q}(W)$ nor a quantized-backbone C was tried. Each $n = 1000$ cell is a single batch. KRE's noise is isotropic and smaller than the int3 key displacement, and neither the key-robust method of Yan et al. (2025) nor MobiEdit (Lu et al., 2026) is run as a baseline; no GPTQ or AWQ calibration set includes the edit prompts. Floating-point (FP8, FP4), vector and lattice formats are outside assumption (iii) of Lemma 1 and untested (Appendix B).

B THE PERIODIC VARIANCE OF LEMMA 1

The sub-bin variance $\Delta|\delta| - \delta^2$ is the $m = 0$ case of the periodic form and becomes negative past one bin; used as a predictor (negative terms clamped at zero) it under-predicts the readout error by $2.1\times$ at int8 and $n = 1$ and $4.2\times$ at int8 and $n = 100$ on GPT-2-XL. Table 4 shows the prediction with the periodic variance of Lemma 1 against the same measured outcomes. Every $n \le 100$ cell of these two models (GPT-2-XL and Llama-3.2-3B) lands at 0.895–0.992 (the int4 and NF4 cells at $n = 100$ are in Table 6), including int8 where $\bar{\delta}/\Delta$ reaches 18.9 and 95% of entries move past a full bin; at int3 with $n \le 100$ and at int4 and NF4 with $n = 1$, where at most 10.3% of entries move a full bin, the two forms differ by up to 20% (NF4, $n = 1$). The $n = 1000$ rows sit at 0.54–0.75 in the table (0.54–0.78 on these two models over int3, int4, NF4 and int8, 0.66–0.75 on GPT-J-6B and 0.77–0.80 on Llama-3.1-8B, so 0.54–0.80 over the four main models; ratio of cell means; every cell in Table 6). Assumption (i) is the cause at $n = 100$ and on Llama-3.2-3B at $n = 1000$ (on GPT-2-XL at $n = 1000$ (ii) fails as well; below): a batch raises the group absmax, so $\mathcal{Q}(W)$ and $\mathcal{Q}(W + \Delta W)$ sit on different lattices, and Δ in (3) and in $\bar{\delta}/\Delta$ is the step of the edited matrix. On a synthetic int4 matrix the edited matrix's step exceeds the unedited one's by a factor 1.00, 1.03, 1.14, 1.31 on average at $\bar{\delta}/\Delta = 0.05, 0.4, 1.0, 1.6$ against the unedited step (the inflation depends on $\bar{\delta}$ relative to the group absmax, $(2^{b-1} - 1)\Delta$, so these factors hold for int4 only), and forcing both onto one lattice restores accuracy to 0.96–0.99 at every displacement tested there. On the real matrices (Table 9, below) a batch of 100 raises the step somewhat more than this synthetic update at a similar int4 displacement: by 1.27 and 1.24 on average (median 1.18 and 1.19) at $\bar{\delta}/\Delta = 1.13$ and 1.26 against the unedited int4 step, where the synthetic factors are 1.14 at 1.0 and 1.31 at 1.6. Monte Carlo on the lemma itself, a check of the implementation rather than of the assumptions (an unclipped grid; 4×10^5 independent uniform phases at each of ten displacements from 0.05Δ to 18Δ, `code/synth_lemma1.py`) gives $|\mathbb{E}[\tilde{E}]| < 10^{-3}\Delta$ and a variance within $4 \times 10^{-4}\Delta^2$ of $\Delta r - r^2$, every deviation within 1.2 Monte Carlo standard errors. On Qwen2.5-3B ($\bar{\delta}/\Delta = 3.0$ at int8, $n = 1$) the periodic form improves on the sub-bin form, sits at the band's lower edge at $n = 1$ and misses it at $n = 100$: the sub-bin form is off by $2.4\times$ and $2.1\times$ at int8 ($n = 1, 100$), and the periodic form lands at 0.89–0.91 at $n = 1$ and

0.74–0.84 at $n = 100$ over its four quantizers (ratio of cell means; at int8 the mean per-edit ratios are 0.927 and 0.738). The over-prediction grows with n on every model (every one of the 16 $n = 100$ cells of Table 6 is below 1, the $n = 1$ cells sit at 0.970–1.017), which the lattice measurement below accounts for in the cells it covers. ZsRE's per-edit ratios are in the caption of Table 5.

Table 4: Measured readout error divided by predicted $1/\mathrm{ESNR}$ (ratio of the cell means), MEMIT, with the sub-bin variance $\Delta|\delta| - \delta^2$ (negative terms clamped at zero) and with the periodic variance of Lemma 1; "> 1 bin" is the fraction of entries displaced by more than one step. Left GPT-2-XL, right Llama-3.2-3B.

		GPT-2-XL				Llama-3.2-3B			
quant	n	$\bar{\delta}/\Delta$	> 1 bin	sub-bin	periodic	$\bar{\delta}/\Delta$	> 1 bin	sub-bin	periodic
int3	1	0.036	0.001	1.016	0.980	0.041	0.000	1.054	0.970
int4	1	0.084	0.006	1.118	0.983	0.096	0.003	1.096	0.972
NF4	1	0.110	0.011	1.189	0.992	0.123	0.008	1.166	0.987
int8	1	1.516	0.425	2.114	0.979	1.733	0.492	2.165	0.973
int3	100	0.391	0.080	0.969	0.905	0.446	0.103	0.984	0.908
int8	100	16.542	0.954	4.173	0.895	18.866	0.961	4.613	0.899
int3	1000	0.442	0.201	0.553	0.540	0.742	0.281	0.869	0.752
int8	1000	18.723	0.960	1.140	0.702	31.412	0.978	4.850	0.751

Table 5: Sub-bin edits are not erased, and the closed form predicts what happens to them: fraction of grid rows (all editors; at $n \geq 100$ the rows of one editor share one update) whose mean per-parameter $|\Delta W_{ij}|$ is below the bin width (a criterion on the mean: individual entries can exceed a bin), how far below (median), their weight-channel margin retention (mean over rows of the per-row ratio of weights-only to fp margin, rows with an fp margin above 0.5 nats; 95% CI; the bin-width criterion predicts that the edit is rounded away, i.e. a ratio equal to the unedited model's margin over the edited model's, which is negative), the mean per-edit measured/predicted ratio at $n = 1$ for MEMIT at int3 and int8 (Table 6 reports the ratio of cell means), and r_s, Spearman$(1/\mathrm{ESNR}$, readout error$)$ over the pooled rows (all 12,000 of a main model: every editor, quantizer and batch size, so it largely ranks cells; within a cell its median is $+0.904$, Appendix X). Retention intervals are row-level (a fact-level bootstrap widens them by up to $1.7\times$). Qwen pools its 4,800 $n = 1$ rows; ZsRE pools 2,700 rows per model (three editors, int3, int4, NF4, $n \leq 100$) and has no int8 run: its MEMIT per-edit ratio at $n = 1$ is 0.990, 0.994 and 1.010 on GPT-2-XL (int3, int4, NF4) and 0.983, 0.979 and 1.003 on Llama-3.2-3B, 0.857–0.906 at $n = 100$, and 0.965 and 0.952 over all 2,700 rows. Every ratio uses the periodic variance of Lemma 1: the re-prediction of this appendix (rows counted in Appendix A) on the four main models, the grids' own values on Qwen2.5-3B and ZsRE, which were run after the correction. "Sub-bin" is measured against the step of the deployed quantizer on $W + \Delta W$ (Table 3).

				per-edit ratio, $n = 1$		
model	sub-bin	below Δ	retention	int3	int8	r_s
GPT-2-XL	48%	8.2×	0.953 [0.945, 0.960]	0.983	0.983	0.985
Llama-3.2-3B	46%	7.7×	0.991 [0.987, 0.994]	0.974	0.975	0.957
GPT-J-6B	52%	9.5×	0.966 [0.960, 0.971]	0.992	0.986	0.999
Llama-3.1-8B	51%	8.7×	0.997 [0.994, 0.999]	0.987	0.985	0.994
Qwen2.5-3B ($n = 1$)	79%	6.1×	0.999 [0.997, 1.002]	0.932	0.927	0.970
GPT-2-XL, ZsRE	89%	12.6×	0.997 [0.996, 0.998]	0.990	—	0.994
Llama-3.2-3B, ZsRE	78%	10.6×	0.999 [0.997, 1.000]	0.983	—	0.992

All 32 cells, and what is out of sample. Table 6 lists, and Figure 3 plots, every MEMIT cell at $n \leq 100$ on the four main models: the ratio of cell means is 0.895–1.017 and the mean per-edit ratio 0.895–1.016, every value above 1 being an NF4 cell, outside (iii) and so outside the bound of Proposition 1. Their $\bar{\delta}/\Delta$ runs from 0.033 (GPT-J-6B and Llama-3.1-8B, int3, $n = 1$) to 18.9 (Llama-3.2-3B, int8, $n = 100$). For MEMIT's single edits at int3, int4 and NF4 it is 0.036, 0.084, 0.110 on GPT-2-XL, 0.041, 0.096, 0.123 on Llama-3.2-3B, 0.033, 0.076, 0.099 on GPT-J-6B, 0.033, 0.077, 0.098 on Llama-3.1-8B and 0.071, 0.166, 0.229 on Qwen2.5-3B; over all six editors on the four main models it is 0.02–0.19. The periodic variance was adopted after the sub-bin form failed on these same cells, so this table is a fit of the corrected form, not an out-of-sample test of it; the

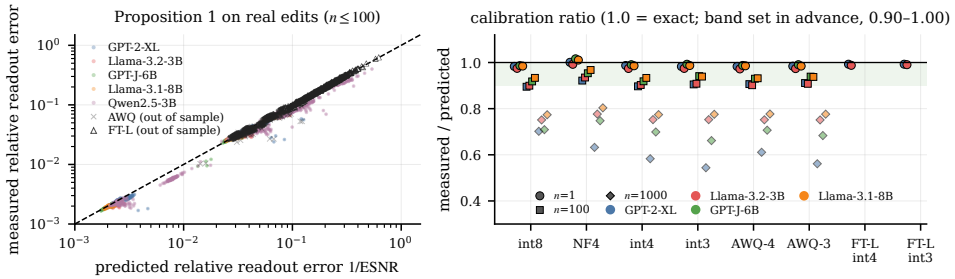


Figure 3: **Proposition 1 is calibrated at $n \le 100$ (in sample for rounding and NF4), not merely monotone.** Left: predicted $1/\mathrm{ESNR}$ against the measured weight-channel readout error for every MEMIT edit at $n \le 100$ on five models, with the out-of-sample AWQ ($\times$) and FT-L ($\triangle$) points; dashed: measured = predicted. Right: mean per-edit measured/predicted per quantizer and batch size on the four main models (FT-L: GPT-2-XL and Llama-3.2-3B, $n = 1$), in sample for round-to-nearest and NF4 and out of sample for AWQ and FT-L (shaded: the 0.90–1.00 band logged before the last cells were run, on the cells run by then). Every $n \le 100$ point lies in 0.89–1.02, five just outside the band, including int8 where 95% of entries move past a full bin (Qwen2.5-3B's overloaded batches of 100, plotted only on the left, sit at 0.74–0.85); AWQ and FT-L land where their round-to-nearest partners do; the $n = 1000$ cells sit at 0.54–0.80, in part because the batch update inflates the group scales (Table 9).

out-of-sample tests are AWQ and FT-L (§4.4), which were logged before they were run. The ZsRE and Qwen2.5-3B grids, run after the correction but not logged as tests, are out of sample for it as well: ZsRE's single edits land at 0.979–1.010 (mean per-edit ratio) and its batches of 100 at 0.857–0.906, and Qwen2.5-3B, which violates (ii) at int3, at 0.89–0.91 ($n = 1$) and 0.74–0.84 ($n = 100$) as ratios of cell means. Proposition 1 equates $1/\mathrm{ESNR}$ with the root-mean-square readout error and bounds the mean error by it. The tables report means, but for an error spread over many of the $d_{\mathrm{out}} = 1600$–4096 output rows the mean falls short of the root mean square by a relative $1/(4d_{\mathrm{eff}})$ to leading order and by slightly less in fact (Gaussian row errors, $d_{\mathrm{eff}} = (\sum_i \sigma_i^2)^2 / \sum_i \sigma_i^4$ the effective number of rows; 0.0246 against 0.025 at $d_{\mathrm{eff}} = 10$ with equal variances), so the proposition predicts a ratio just below one for the mean as well: ratios near 0.9 at $n = 100$ measure the failure of assumption (i) (Table 9), not the slack of the bound.

What the pooled sub-bin fraction counts. Table 7 breaks the sub-bin fraction and the retention down by quantizer and batch size. At int8, rows are rarely below the bin width (0.7–16.8% at $n = 1$) and int3, int4 and NF4 single edits almost always are, so the pooled 46–52% is set by the design, and at $n \ge 100$ a cell is six updates rather than hundreds of edits. The statement that matters is the per-cell one: single edits at int3, int4 and NF4 are at least 99.2% sub-bin in every cell and keep 0.995–1.001 of their fp margin when only the edited matrix is quantized (the average matters only in the third decimal: the ratio of cell means gives 0.994–1.000, the median per-row ratio 0.996–1.000, and moving the 0.5-nat filter to 0, 1 or 2 nats 0.992–1.002). The filter drops 80 of the 14,384 sub-bin single-edit rows (67 of them KAE or KAE+KRE edits). Per edit, pooled over the four models, six editors and three quantizers, the 1st and 5th percentiles of retention are 0.943 and 0.972; 43 rows (0.3%) keep less than 0.9, the lowest 0.686. The weights-only margin also contains the rounding of W itself, $(\mathcal{Q}(W) - W)k^*$, which Lemma 1 does not model; the readout error of Tables 4 and 6 isolates $\tilde{E}$. MEMIT's single edits move the entries by 0.033–0.123 of a bin on the four main models and 0.071–0.229 on Qwen2.5-3B (int3, int4, NF4; listed above), and the single edits of all six editors span 0.021–0.187 on the four main models. These are unweighted means; the readout weights entry (i, j) by k_j^{*2}, and alignment makes the entries under large $|k_j^*|$ the large ones, so the entries that carry the readout move further than $\bar{\delta}/\Delta$ says (the int4-to-int3 growth of the predicted single-edit error in Table 6, 2.0–2.1$\times$ against $\sqrt{7/3} \approx 1.53$ for entries far below a bin, reflects this). The pooled retention of Table 5 is lower on GPT-2-XL and GPT-J because it includes the overloaded int3, $n = 1000$ updates, which are sub-bin by the mean only against the step they themselves inflate (6.3 bins against the unedited step on GPT-2-XL at int3, Table 9) and keep only 0.734 and 0.808.

Assumption (ii) on the real matrices. For a given checkpoint W is fixed, so the expectations of Lemma 1 are over a model in which the phases are uniform draws independent of δ; the checks

below treat the entries of a matrix as such draws. Table 8 reads the edited matrix of every checkpoint and measures what (ii) is for: the fraction of a sub-bin displacement of t bins that rounding delivers in expectation (`code/phase_uniformity.py`, `code/phase_delivered.py`; the gated Llama weights are read from public mirrors whose shards have the SHA-256 of the files our GPU runs loaded). Read off the phases as boundary mass, it is 0.992–1.008 on the fp32 and fp16 checkpoints and 0.94 upward against 1.06 downward on the two bf16 Llamas at int3 and int4. That statistic scores an entry lying exactly on a rounding boundary as moving down, which is how $\lfloor x + \frac{1}{2} \rfloor$ rounds; the quantizer rounds half to even, and 0.35–0.38% of a bf16 Llama matrix's entries sit exactly on a boundary at int3 and int4 (bf16 keeps eight significant bits, and $|W| = \mathrm{absmax}/2$ is a boundary at every bit width). Measured with the quantizer's own rounding, the delivered fraction is 0.995–1.005 at int3 and int4, in both directions and at $t = 0.03$ and 0.1, on all six checkpoints other than Qwen2.5-3B, so (ii) holds there in the sense the lemma uses it, as far as a displacement common to all entries can show: the test does not condition on δ_{ij}, so the independence half of (ii) is checked only through the real edits' readout error (Table 6) and retention (Table 7). At int8 the ties make the three bf16 checkpoints over-deliver a 0.03-bin displacement in both directions (1.024–1.055; 1.002–1.007 at $t = 0.1$); int8 edits move entries by 1.5–31 bins on average, but about half the entries of an int8 single edit move by less than one, so the ties can reach them. Qwen2.5-3B violates (ii) at int3 whichever way it is measured: rounding delivers 0.820 of a small displacement in either direction (0.964 at int4), because its int3 step is coarse relative to the bulk of its weights and the boundary is under-populated. That is an entry-level statement; the readout weights entries by the key, and the measured readout error bounds whatever bias reaches it: Qwen's int3 readout error at $n = 1$, bias included, is 0.083 of the edit's own readout (Table 5 gives its ratio to the variance-only prediction), well below the 0.18 an entry loses, so the entry-level shortfall must sit mostly in entries the key down-weights. Absmax entries, which sit on the grid by construction, are 1.56–1.59% of every matrix in Table 8. Two cases are outside the lemma outright: a checkpoint already on the deployment grid (quantization-aware trained, or quantized and dequantized with the same groups), where every $\varphi_{ij} = \frac{1}{2}$ and any update below half a step is erased deterministically, as the sub-bin explanation says; and a non-uniform grid such as NF4, which is outside (iii): the phase is not defined across cells of unequal width, and a displacement's mean effect depends on the widths of the cells it crosses. Vector and lattice codebooks and floating-point formats (FP8, FP4) are outside (iii) in the same way, and none is tested here.

The lattice on the real matrices. ESNR and $\bar{\delta}/\Delta$ use the step of the edited matrix $W + \Delta W$ (written Δ' in this paragraph and Table 9, where Δ is the step of W), whereas Lemma 1 assumes one grid for W and $W + \Delta W$ (assumption (i)). Table 9 measures the inflation Δ'/Δ group by group on the real edited down-projections and repeats on them the synthetic common-lattice test above (`code/lattice_real.py`). A batch of 100 raises the step of 84–86% of the groups, by a median factor of 1.18–1.19, and the predictor with the post-edit step then over-predicts the measured error by about 10% (ratio of means 0.90–0.91) where the pre-edit step under-predicts it by a similar amount (ratio of means 1.10–1.13; the table's "pre-edit" column, a median per-probe ratio, agrees to within 0.002); with W and $W + \Delta W$ both rounded on the edited matrix's lattice the ratio is 0.99–1.00. In these four $n = 100$ cells the residual is the lattice mismatch. A batch of 1000 raises 99.9–100% of the group steps, 2.6-fold at the median (2.7 on average) on Llama-3.2-3B, where the common lattice again removes the whole over-prediction (0.97–1.00), and 9.7-fold at the median (14.7 on average) on GPT-2-XL, where it does not (0.55–0.63). There, at int3, a group's unedited weights, at most three old steps from zero, lie within a third of a new step of it in half the groups (on Llama-3.2-3B within 1.1 new steps at the median), so their phases are no longer spread over the bin: assumption (ii) fails, and the update is rounded almost without dither (a plain-rounding variance $\Delta'^2/12$ does not fit either: measured over predicted 0.48–0.65, median over probes). The within-row uncorrelatedness that Proposition 1 adds holds in every cell of the table ($\sqrt{I} = 0.995$–1.003). The $\bar{\delta}/\Delta$ of the $n = 1000$ cells is therefore measured against a lattice the update has itself coarsened: against the unedited one it is 2.0 (Llama-3.2-3B) and 6.3 (GPT-2-XL) at int3.

Edit, save in bf16, quantize. The grid holds the edited matrix in fp32. Editing a bf16 checkpoint and saving it before quantizing rounds $W + \Delta W$ to bf16 first, about eight significant bits per entry, which is the mechanism that confines small-learning-rate unlearning to near-zero weights (Appendix J). Table 10 runs that path on Llama-3.2-3B and Qwen2.5-3B, two of the three bf16 checkpoints, with the fp32 path in the same run. Rounding the edited matrix to bf16 before anything else leaves the edit where it was: on Llama-3.2-3B it moves the margin by $+0.01$ $[-0.01, +0.03]$ nats for

single edits and -0.00 $[-0.02, +0.02]$ for a batch of 100, bf16 rounds away 9.7% and 0.3% of the update's entries entirely, and the readout error it adds at the key is 0.1% of the edit's (on Qwen2.5-3B -0.01 $[-0.06, +0.03]$ and -0.00 $[-0.04, +0.03]$ nats, 10.0% and 0.3% of the entries rounded away); stored in bf16 and then quantized, Llama-3.2-3B's single-edit weights-only retention is 0.9962 against 0.9958 in fp32 at int3 and 0.9993 against 0.9993 at int4. The fp32 path of this run moves Qwen2.5-3B's margin by -0.21 $[-0.41, +0.01]$ nats at int3 on its first 100 probes, more than the grid's -0.11 on the same probes (-0.09 over its 200, Table 11) and 1.4% of the full channel's loss, so the "at most 0.09 nats" of §4.1 is the grid's figure, not a bound (Appendix A compares the two runs). Since the weights of a bf16 checkpoint already sit on the bf16 grid, this step does not dither: it rounds the update itself, on a grid whose spacing is 2^{-8} to 2^{-7} of each weight's magnitude, which is far below a closed-form update on the entries that carry the key's mass and above a small-learning-rate unlearning update on typical weights. That is why the same storage step erases the unlearning update and not the edit.

Table 6: Every calibration cell at $n \leq 100$ on the four main models (MEMIT; below the rule, the $n = 1000$ cells and Qwen2.5-3B, which are not among the 32 of the text): mean measured weight-channel readout error, mean predicted $1/\mathrm{ESNR}$ (periodic variance), their ratio, and the mean of the per-edit ratios.

model	quant	n	measured	predicted	ratio of means	per-edit ratio
GPT-2-XL	int8	1	0.00243	0.00248	0.979	0.983
GPT-2-XL	int8	100	0.00279	0.00312	0.895	0.895
GPT-2-XL	int4	1	0.03918	0.03984	0.983	0.987
GPT-2-XL	int4	100	0.05050	0.05632	0.897	0.897
GPT-2-XL	NF4	1	0.03273	0.03299	0.992	1.001
GPT-2-XL	NF4	100	0.04102	0.04451	0.922	0.922
GPT-2-XL	int3	1	0.08150	0.08315	0.980	0.983
GPT-2-XL	int3	100	0.11604	0.12821	0.905	0.905
Llama-3.2-3B	int8	1	0.00200	0.00205	0.973	0.975
Llama-3.2-3B	int8	100	0.00224	0.00249	0.899	0.900
Llama-3.2-3B	int4	1	0.03220	0.03313	0.972	0.974
Llama-3.2-3B	int4	100	0.04065	0.04506	0.902	0.903
Llama-3.2-3B	NF4	1	0.02800	0.02836	0.987	0.992
Llama-3.2-3B	NF4	100	0.03459	0.03705	0.934	0.935
Llama-3.2-3B	int3	1	0.06396	0.06592	0.970	0.974
Llama-3.2-3B	int3	100	0.09388	0.10343	0.908	0.908
GPT-J-6B	int8	1	0.00198	0.00201	0.988	0.986
GPT-J-6B	int8	100	0.00222	0.00242	0.918	0.918
GPT-J-6B	int4	1	0.03025	0.03050	0.992	0.991
GPT-J-6B	int4	100	0.04019	0.04375	0.919	0.919
GPT-J-6B	NF4	1	0.02576	0.02532	1.017	1.016
GPT-J-6B	NF4	100	0.03302	0.03462	0.954	0.953
GPT-J-6B	int3	1	0.06079	0.06121	0.993	0.992
GPT-J-6B	int3	100	0.09244	0.09822	0.941	0.941
Llama-3.1-8B	int8	1	0.00197	0.00200	0.984	0.985
Llama-3.1-8B	int8	100	0.00212	0.00227	0.933	0.933
Llama-3.1-8B	int4	1	0.03087	0.03133	0.986	0.986
Llama-3.1-8B	int4	100	0.03831	0.04106	0.933	0.933
Llama-3.1-8B	NF4	1	0.02682	0.02656	1.010	1.011
Llama-3.1-8B	NF4	100	0.03274	0.03384	0.967	0.968
Llama-3.1-8B	int3	1	0.06042	0.06126	0.986	0.987
Llama-3.1-8B	int3	100	0.08785	0.09357	0.939	0.939
GPT-2-XL	int8	1000	0.04359	0.06208	0.702	0.702
GPT-2-XL	int4	1000	0.48635	0.83921	0.580	0.583
GPT-2-XL	NF4	1000	0.36654	0.58391	0.628	0.632
GPT-2-XL	int3	1000	0.85627	1.58432	0.540	0.544
Llama-3.2-3B	int8	1000	0.00402	0.00536	0.751	0.751
Llama-3.2-3B	int4	1000	0.07316	0.09722	0.752	0.752
Llama-3.2-3B	NF4	1000	0.05910	0.07611	0.777	0.777
Llama-3.2-3B	int3	1000	0.17030	0.22635	0.752	0.752
GPT-J-6B	int8	1000	0.01737	0.02448	0.709	0.709
GPT-J-6B	int4	1000	0.30383	0.43454	0.699	0.699
GPT-J-6B	NF4	1000	0.21253	0.28374	0.749	0.748
GPT-J-6B	int3	1000	0.61893	0.93571	0.661	0.662
Llama-3.1-8B	int8	1000	0.00329	0.00425	0.774	0.774
Llama-3.1-8B	int4	1000	0.05964	0.07705	0.774	0.774
Llama-3.1-8B	NF4	1000	0.04906	0.06107	0.803	0.804
Llama-3.1-8B	int3	1000	0.13903	0.17929	0.775	0.775
Qwen2.5-3B	int8	1	0.00237	0.00262	0.907	0.927
Qwen2.5-3B	int8	100	0.00586	0.00794	0.737	0.738
Qwen2.5-3B	int4	1	0.03829	0.04210	0.909	0.931
Qwen2.5-3B	int4	100	0.10477	0.13451	0.779	0.781
Qwen2.5-3B	NF4	1	0.03297	0.03704	0.890	0.927
Qwen2.5-3B	NF4	100	0.06967	0.09044	0.770	0.772
Qwen2.5-3B	int3	1	0.08271	0.09090	0.910	0.932
Qwen2.5-3B	int3	100	0.23453	0.27877	0.841	0.848

Table 7: Fraction of grid rows whose mean $|\Delta W_{ij}|$ is below the bin width, by quantizer and batch size (six editors pooled; at $n \geq 100$ all rows of one editor share one update, so each such cell is six updates, not hundreds of edits), and the weight-channel margin retention of those sub-bin rows in each cell (as in Table 5); "—" where no row is sub-bin.

model	quant	sub-bin, $n = 1$	$n = 100$	$n = 1000$	retention, $n = 1$	$n = 100$	$n = 1000$
GPT-2-XL	int8	5.8%	0.0%	0.0%	1.000	—	—
GPT-2-XL	int4	99.8%	50.0%	0.0%	0.999	0.999	—
GPT-2-XL	NF4	99.8%	0.0%	0.0%	0.995	—	—
GPT-2-XL	int3	100.0%	100.0%	100.0%	1.001	0.996	0.734
Llama-3.2-3B	int8	0.7%	0.0%	0.0%	1.000	—	—
Llama-3.2-3B	int4	100.0%	16.7%	0.0%	0.999	1.000	—
Llama-3.2-3B	NF4	100.0%	0.0%	0.0%	0.999	—	—
Llama-3.2-3B	int3	100.0%	100.0%	100.0%	0.998	0.995	0.962
GPT-J-6B	int8	16.8%	0.0%	0.0%	1.000	—	—
GPT-J-6B	int4	99.8%	83.3%	0.0%	1.000	1.000	—
GPT-J-6B	NF4	99.2%	16.7%	0.0%	0.998	0.999	—
GPT-J-6B	int3	100.0%	100.0%	100.0%	0.999	0.998	0.808
Llama-3.1-8B	int8	10.0%	0.0%	0.0%	1.000	—	—
Llama-3.1-8B	int4	100.0%	83.3%	0.0%	1.001	0.999	—
Llama-3.1-8B	NF4	100.0%	16.7%	0.0%	0.999	0.998	—
Llama-3.1-8B	int3	100.0%	100.0%	100.0%	0.997	0.995	0.988

Table 8: Assumption (ii) of Lemma 1 on the matrix each checkpoint edits, before editing and in its stored format (group 64, absmax entries excluded; GPT-2-XL's grid runs load it in fp16 first, which this table does not measure). KS is the Kolmogorov–Smirnov distance of the fractional grid positions φ from uniform; "on a boundary" is the fraction of entries whose φ is exactly 0, i.e. that sit on a rounding boundary. Each up/down pair is the expected fraction of a sub-bin displacement of t bins that rounding delivers upward and downward, 1 under (ii): "boundary mass" reads it off the phases as $P(\varphi \geq 1 - t)/t$ and $P(\varphi < t)/t$, which scores an entry on a boundary as moving down; "quantizer" applies the round-to-nearest operator itself, which rounds half to even (`code/phase_delivered.py`). s.e. is the binomial standard error under (ii) at $t = 0.03$.

model	stored	grid	KS	on a boundary	boundary mass, t=0.03	quantizer, t=0.03	quantizer, t=0.1	s.e.
GPT-2	fp32	int3	0.0005	0.00%	0.999 / 0.995	0.999 / 0.995	0.999 / 0.996	0.004
GPT-2	fp32	int4	0.0005	0.00%	0.997 / 1.002	0.997 / 1.002	1.001 / 1.000	0.004
GPT-2	fp32	int8	0.0005	0.00%	1.006 / 1.002	1.006 / 1.002	1.003 / 0.999	0.004
GPT-2-medium	fp32	int3	0.0004	0.00%	0.998 / 1.003	0.998 / 1.003	0.999 / 1.000	0.003
GPT-2-medium	fp32	int4	0.0003	0.00%	1.002 / 0.997	1.002 / 0.997	1.000 / 1.001	0.003
GPT-2-medium	fp32	int8	0.0004	0.00%	0.997 / 0.995	0.997 / 0.994	1.000 / 0.999	0.003
GPT-2-XL	fp32	int3	0.0007	0.00%	1.004 / 1.002	1.004 / 1.002	1.003 / 1.002	0.002
GPT-2-XL	fp32	int4	0.0004	0.00%	0.999 / 1.003	0.999 / 1.003	1.000 / 1.003	0.002
GPT-2-XL	fp32	int8	0.0003	0.00%	0.998 / 1.000	0.998 / 1.000	1.001 / 1.001	0.002
GPT-J-6B	fp16	int3	0.0008	0.04%	0.994 / 1.008	1.001 / 1.002	1.001 / 1.002	0.001
GPT-J-6B	fp16	int4	0.0005	0.04%	0.995 / 1.007	1.001 / 1.000	1.000 / 1.000	0.001
GPT-J-6B	fp16	int8	0.0004	0.04%	0.992 / 1.005	0.998 / 0.999	1.000 / 1.000	0.001
Llama-3.2-3B	bf16	int3	0.0035	0.35%	0.944 / 1.058	1.000 / 1.000	1.002 / 1.003	0.001
Llama-3.2-3B	bf16	int4	0.0037	0.36%	0.942 / 1.059	1.000 / 1.001	1.000 / 1.001	0.001
Llama-3.2-3B	bf16	int8	0.0028	0.24%	0.985 / 1.064	1.024 / 1.024	1.006 / 1.007	0.001
Llama-3.1-8B	bf16	int3	0.0038	0.38%	0.940 / 1.063	1.001 / 1.001	1.002 / 1.002	0.001
Llama-3.1-8B	bf16	int4	0.0039	0.37%	0.945 / 1.063	1.005 / 1.003	1.000 / 1.001	0.001
Llama-3.1-8B	bf16	int8	0.0052	0.48%	0.976 / 1.134	1.055 / 1.055	1.005 / 1.004	0.001
Qwen2.5-3B	bf16	int3	0.0385	0.24%	0.781 / 0.859	0.820 / 0.820	0.828 / 0.828	0.001
Qwen2.5-3B	bf16	int4	0.0095	0.23%	0.928 / 1.000	0.964 / 0.964	0.960 / 0.959	0.001
Qwen2.5-3B	bf16	int8	0.0028	0.25%	0.986 / 1.072	1.028 / 1.030	1.002 / 1.003	0.001

Table 9: The lattice on the real edited matrices (MEMIT, symmetric groups of 64 as in the main grid). $\bar{\delta}/\Delta'$ with the post-edit step (pre-edit in parentheses); share of (row, group) steps the update raised; Δ'/Δ per group: mean, median, 95th percentile and mean weighted by the group's k^{*2} mass; measured over predicted readout error (ratio of means over the batch's first 100 probes) with the post-edit step, as the grid computes it; pre-edit: the same (median over probes) with the pre-edit step; $\sqrt{I} = \|\tilde{E}k^*\|/(\sum_{ij}\tilde{E}_{ij}^2 k_j^{*2})^{1/2}$ (median), which is close to 1 when the rounding errors of a row are uncorrelated; common lattice: the same ratio of means when W and $W + \Delta W$ are both rounded on the post-edit lattice.

model	quant	n	$\bar{\delta}/\Delta'$	raised	mean	median	p95	k^{*2}-w.	meas./pred.	pre-edit	$\sqrt{I}$	common
GPT-2-XL	int3	100	0.391 (0.484)	84.0%	1.266	1.179	1.887	1.269	0.905	1.122	0.998	0.986
GPT-2-XL	int4	100	0.912 (1.130)	84.0%	1.266	1.179	1.887	1.269	0.897	1.129	1.001	0.992
GPT-2-XL	int3	1000	0.442 (6.289)	100.0%	14.697	9.656	36.066	14.650	0.538	8.206	0.995	0.553
GPT-2-XL	int4	1000	1.032 (14.675)	100.0%	14.697	9.656	36.066	14.650	0.582	10.678	0.997	0.631
Llama-3.2-3B	int3	100	0.446 (0.542)	86.0%	1.239	1.192	1.713	1.244	0.908	1.099	0.999	0.988
Llama-3.2-3B	int4	100	1.040 (1.264)	86.0%	1.239	1.192	1.713	1.244	0.902	1.104	1.003	0.995
Llama-3.2-3B	int3	1000	0.742 (1.960)	99.9%	2.739	2.624	4.109	2.740	0.752	1.980	1.003	0.973
Llama-3.2-3B	int4	1000	1.731 (4.573)	99.9%	2.739	2.624	4.109	2.740	0.753	1.986	1.003	1.000

Table 10: Edit, save in bf16, quantize. The edited matrix $W + \Delta W$ is rounded to bf16 before quantization ("bf16 only": nothing else is done to it); the int3 and int4 rows without the prefix hold it in fp32 as the main grid does, in the same run (NF4 only after bf16 storage). Weights-only: margin change when only the edited matrix is stored/quantized; retention: weights-only over fp margin; full: margin change when every matrix gets the row's treatment (a no-op on the bf16 backbone in the "bf16 only" rows); ES fp/dep.: efficacy unquantized / in the full channel; lost: share of the update's nonzero entries that bf16 rounds away entirely; storage error: $\|(\mathrm{bf16}(W+\Delta W) - W - \Delta W)k^*\|/\|\Delta W k^*\|$. MEMIT, 100 probes.

model	n	storage / quant	fp	weights-only	retention	full	ES fp/dep.	lost	storage error
Llama-3.2-3B	1	bf16 only	14.03	$+0.01$ $[-0.01, +0.03]$	1.0009	$+0.01$ $[-0.01, +0.03]$	1.00/1.00	9.7%	0.001
Llama-3.2-3B	1	int3	14.03	-0.06 $[-0.12, +0.01]$	0.9958	-7.03 $[-7.72, -6.34]$	1.00/0.97	–	–
Llama-3.2-3B	1	bf16→int3	14.03	-0.05 $[-0.11, +0.01]$	0.9962	-7.02 $[-7.70, -6.34]$	1.00/0.97	9.7%	0.001
Llama-3.2-3B	1	int4	14.03	-0.01 $[-0.04, +0.02]$	0.9993	-1.50 $[-1.87, -1.14]$	1.00/1.00	–	–
Llama-3.2-3B	1	bf16→int4	14.03	-0.01 $[-0.04, +0.02]$	0.9993	-1.52 $[-1.89, -1.16]$	1.00/1.00	9.7%	0.001
Llama-3.2-3B	1	bf16→NF4	14.03	-0.01 $[-0.03, +0.02]$	0.9994	-1.18 $[-1.52, -0.87]$	1.00/0.99	9.7%	0.001
Llama-3.2-3B	100	bf16 only	13.76	-0.00 $[-0.02, +0.02]$	0.9998	-0.00 $[-0.02, +0.02]$	1.00/1.00	0.3%	0.001
Llama-3.2-3B	100	int3	13.76	-0.05 $[-0.11, +0.02]$	0.9967	-6.98 $[-7.66, -6.31]$	1.00/0.94	–	–
Llama-3.2-3B	100	bf16→int3	13.76	-0.05 $[-0.12, +0.01]$	0.9962	-6.99 $[-7.68, -6.32]$	1.00/0.94	0.3%	0.001
Llama-3.2-3B	100	int4	13.76	-0.00 $[-0.04, +0.03]$	0.9999	-1.47 $[-1.84, -1.12]$	1.00/1.00	–	–
Llama-3.2-3B	100	bf16→int4	13.76	-0.01 $[-0.05, +0.03]$	0.9992	-1.48 $[-1.85, -1.12]$	1.00/1.00	0.3%	0.001
Llama-3.2-3B	100	bf16→NF4	13.76	-0.02 $[-0.05, +0.01]$	0.9987	-1.13 $[-1.41, -0.85]$	1.00/1.00	0.3%	0.001
Qwen2.5-3B	1	bf16 only	15.61	-0.01 $[-0.06, +0.03]$	0.9991	-0.01 $[-0.06, +0.03]$	1.00/1.00	10.0%	0.001
Qwen2.5-3B	1	int3	15.61	-0.21 $[-0.41, +0.01]$	0.9863	-15.51 $[-16.76, -14.30]$	1.00/0.54	–	–
Qwen2.5-3B	1	bf16→int3	15.61	-0.18 $[-0.40, +0.08]$	0.9884	-15.51 $[-16.77, -14.31]$	1.00/0.56	10.0%	0.001
Qwen2.5-3B	1	int4	15.61	$+0.00$ $[-0.11, +0.12]$	1.0000	-3.98 $[-4.99, -3.06]$	1.00/0.95	–	–
Qwen2.5-3B	1	bf16→int4	15.61	-0.05 $[-0.15, +0.06]$	0.9970	-3.95 $[-4.96, -3.02]$	1.00/0.94	10.0%	0.001
Qwen2.5-3B	1	bf16→NF4	15.61	-0.05 $[-0.14, +0.03]$	0.9969	-2.17 $[-2.67, -1.65]$	1.00/0.98	10.0%	0.001
Qwen2.5-3B	100	bf16 only	15.32	-0.00 $[-0.04, +0.03]$	0.9998	-0.00 $[-0.04, +0.03]$	0.99/0.99	0.3%	0.001
Qwen2.5-3B	100	int3	15.32	-1.49 $[-2.01, -0.99]$	0.9024	-15.21 $[-16.52, -13.91]$	0.99/0.55	–	–
Qwen2.5-3B	100	bf16→int3	15.32	-1.48 $[-1.99, -0.97]$	0.9037	-15.21 $[-16.53, -13.93]$	0.99/0.55	0.3%	0.001
Qwen2.5-3B	100	int4	15.32	-0.36 $[-0.71, -0.08]$	0.9765	-4.42 $[-5.52, -3.41]$	0.99/0.91	–	–
Qwen2.5-3B	100	bf16→int4	15.32	-0.35 $[-0.70, -0.08]$	0.9770	-4.40 $[-5.49, -3.40]$	0.99/0.91	0.3%	0.001
Qwen2.5-3B	100	bf16→NF4	15.32	-0.07 $[-0.19, +0.04]$	0.9953	-2.39 $[-2.87, -1.90]$	0.99/0.97	0.3%	0.001

C COROLLARY 2 IN ISOLATION

The main grids confound d_{in} with architecture, data and layer, so Corollary 2 is also tested on synthetic W (Gaussian entries, group-64 RTN-int4, each threshold a median over draws: alignment sweeps over $\cos = 0.02$–0.95 at $d_{\text{in}} = 1024$ with $d_{\text{out}} = 256$ and 60 draws per point, and at $d_{\text{in}} = 3072$, 6400 and 8192 with $d_{\text{out}} = 128$ and 15, 12 and 10 draws; random-direction runs at $d_{\text{in}} = 256$, 1024 and 4096 with $d_{\text{out}} = 256$ and 25 draws, and at 3072, 6400 and 8192 with the draws of their sweeps; a threshold run at $d_{\text{in}} = 256$, 1024 and 3072 with $d_{\text{out}} = 128$ and 12 draws) with keys drawn post-GELU-like (the absolute value of a Gaussian plus 0.1, so all positive), the residual a Gaussian, u constructed at a prescribed $\cos(u, k^*)$ as a mix of k^* and a random orthogonal direction, and the empirical survival threshold r^*_{emp} located by bisection on the scale of a against the realized readout error rather than by the closed form it is meant to check (`code/synth_alignment.py`, `code/synth_threshold.py`; seeds are in the scripts). At fixed $\cos(u, k^*) = 0.1$, $r^*_{\text{emp}} = 3.635 \times 10^{-2}, 1.336 \times 10^{-2}, 5.598 \times 10^{-3}, 4.527 \times 10^{-3}$ at $d_{\text{in}} = 1024, 3072, 6400, 8192$, so $d_{\text{in}} \cdot r^*_{\text{emp}} = 37.2, 41.0, 35.8, 37.1$: within 9% of their mean while r^* moves 8-fold. With a direction at a random direction's mean $d_{\text{in}} \cos^2 = 1$ ($\cos = 1/\sqrt{d_{\text{in}}}$; a uniformly random u has median $d_{\text{in}} \cos^2 \approx 0.45$ and so, by (4), a threshold near 0.28), $r^*_{\text{emp}} = 0.211, 0.215, 0.212, 0.224, 0.215, 0.219$ at $d_{\text{in}} = 256, 1024, 3072, 4096, 6400, 8192$: flat across a 32-fold change in width, which may be why the bin-width criterion has looked like a law.

Derivation of (4). Write $\delta_{ij} = a_i u_j / s$ with $s = u^\top k^*$, so that $\|\Delta W k^*\| = \|a\|$ and $\bar{\delta} = \overline{|a|}\,\overline{|u|}/|s|$, bars denoting means over entries. With every entry below one bin and one step Δ, the denominator of (3) is $\Delta \|a\|_1 \sum_j |u_j| k_j^{*2}/|s| - \|a\|^2 \sum_j u_j^2 k_j^{*2}/s^2$. If $|u_j|$ and u_j^2 are uncorrelated with k_j^{*2}, the two sums are $\overline{|u|}\|k^*\|^2$ and $\overline{u^2}\|k^*\|^2$; substituting $\|a\|^2 = d_{\text{out}} \kappa_a \overline{|a|}^2$, $\|u\|^2 = d_{\text{in}} \kappa_u \overline{|u|}^2$ and $s^2 = \cos^2(u, k^*)\|u\|^2\|k^*\|^2$ gives $\text{ESNR}^2 = \kappa d_{\text{in}} \cos^2 x/(1 - \kappa x)$ with $x = \bar{\delta}/\Delta$, and setting it to 1 gives r^*. When $|u_j|$ is associated with k_j^* the sums acquire factors γ_1, γ_2 (≥ 1 under positive association) and $r^* = \gamma_1/[\kappa(d_{\text{in}} \cos^2 + \gamma_2)]$, which is why the law under-predicts at high alignment. At $\cos = 1$, where $|u_j| \propto |k_j^*|$, $\gamma_1 = \mathbb{E}|k|^3/(\mathbb{E}|k|\,\mathbb{E}k^2)$ and $\gamma_2 = \mathbb{E}k^4/(\mathbb{E}k^2)^2$, which are 2 and 3 for Gaussian key entries (1.83 and 2.71 for the post-GELU-like keys of the synthetic runs above), so at fixed κ an aligned update's threshold is $(d_{\text{in}} + 3)/4$ times below a random direction's; $(d_{\text{in}} + 1)/2$ is reached only for keys of constant magnitude. Dropping the r^2 term gives the linearization $\text{ESNR}^2 \approx \kappa d_{\text{in}} \cos^2 x$, whose random-direction threshold is $1/\kappa \approx 0.4$. For Gaussian entries $\kappa_x = \pi/2$, so $\kappa = \pi^2/4$; other entry shapes move $1/(2\kappa)$, to 0.125 for Laplace entries (lower for heavier tails) and at most 0.5 (constant magnitude). Formula (4) has a pole at $\bar{\delta}/\Delta = 1/\kappa$, which the corollary excludes by requiring $\kappa\bar{\delta}/\Delta < 1$ (automatic when every entry is below one bin, since $\kappa\bar{\delta}/\Delta = \overline{\delta^2}/(\Delta\bar{\delta}) \leq \max|\delta_{ij}|/\Delta < 1$, but not when only almost every entry is: a few entries far past a bin can raise $\overline{\delta^2}$ arbitrarily), and is meant for $\kappa\bar{\delta}/\Delta < \frac{1}{2}$, where raising $\cos(u, k^*)$ at a fixed residual a also raises ESNR (at fixed a and entry shapes $\bar{\delta} \propto 1/|\cos(u, k^*)|$, and $\partial \text{ESNR}^2/\partial|\cos|$ has the sign of $1 - 2\kappa\bar{\delta}/\Delta$); with Gaussian κ, MEMIT's int3, int4 and NF4 single edits sit at $\kappa\bar{\delta}/\Delta \leq 0.31$ on the four main models (cell means; two of the 200 edits on GPT-2-XL and on GPT-J-6B exceed $\frac{1}{2}$) and only Qwen2.5-3B's NF4 edits ($\bar{\delta}/\Delta = 0.23$) pass $\frac{1}{2}$ on average (int8 single edits move entries past a bin and are outside it; κ itself is not measured on real edits). MEMIT's single edits on the four main models have $\cos(u, k^*) = 0.63$–0.75, where $|u_j|$ and k_j^* correlate, and there (4) with Gaussian κ, at each cell's mean $\cos(u, k^*)$ and $\bar{\delta}/\Delta$, under-predicts MEMIT's measured single-edit readout error by 1.45–1.56× at int3 and 1.15–1.30× at int4 on the four main models, and its growth from int4 to int3 (1.62–1.66× predicted, 1.96–2.08× measured): on real edits it gives the direction of the alignment effect, not its size. The size of the miss is not decomposed. Besides the correlation, a real κ below the Gaussian value would also make the predicted error too low (it is not measured), and (4) takes one step Δ where the real step varies by group (the mean step is used), which matters if the step correlates with $|u_j| k_j^{*2}$.

Against the synthetic thresholds. Over the four sweeps the r^* of Corollary 2 with Gaussian $\kappa = \pi^2/4$ divided by r^*_{emp} is 0.95–1.11 for $\cos \leq 0.2$ (except $\cos = 0.02$ at $d_{\text{in}} = 1024$, where $d_{\text{in}} \cos^2 < 1$, i.e. $\kappa r^* > \frac{1}{2}$, near the pole: 0.78), 0.88–0.92 at $\cos = 0.4$ and 0.49–0.73 at $\cos \geq 0.7$; the linearized form with the constant 2 in place of κ, $1/(2d_{\text{in}} \cos^2)$, divided by r^*_{emp}, spans 0.61–3.31 (for a random direction, $d_{\text{in}} \cos^2 = 1$, this form is the half-step rule $r^* = \frac{1}{2}$). For a random direction ($d_{\text{in}} \cos^2 = 1$) the law gives $1/(2\kappa) = 0.203$ against 0.211–0.224 measured; the remaining few

percent are entries past one bin, which the sub-bin variance under-charges. Proposition 1 with the full periodic variance, solved for $\mathrm{ESNR} = 1$ by bisection, predicts 0.216–0.220 at $d_{\mathrm{in}} = 256$–3072 against 0.217–0.221 measured in that run (a second, independent draw; the sweep above gives 0.211–0.215 at the same widths), while the linearization with the sample's own $\kappa_a \kappa_u$ predicts 0.40–0.42 ($= 1/(\kappa_a \kappa_u)$). The full and linearized predictors are therefore distinguishable. The predictor r^*_{pred} is the first-order threshold of (3) with the sample's own Δ_{ij} and k^* (noise $\sum \Delta_{ij} |\delta_{ij}| k_j^{*2}$): over the four sweeps with $\cos \geq 0.05$ its ratio to r^*_{emp} has median 1.04 and range $[0.93, 1.38]$ (the 1.38 at $d_{\mathrm{in}} = 1024$, $\cos = 0.05$; $[0.93, 1.19]$ at $d_{\mathrm{in}} \geq 3072$), and it degrades at $\cos = 0.02$ (1.34–2.71), the near-random regime where the r^2 term matters. A naive extension of (4) to a batch, treating the n residuals as independent noise sources, would predict that ESNR^2 falls as $1/n$; on real MEMIT edits it falls far less: from $n = 1$ to $n = 100$ the measured weight-channel readout error grows only 1.22–1.52× on the four main models at int3, int4 and NF4 (Table 6), so ESNR^2 falls 1.5–2.3×, not 100×. $\|\Delta W\|_F$ does grow roughly as $\sqrt{n}$ (8.6–10.1× from $n = 1$ to $n = 100$ on these models), which in the sub-bin linear regime would by itself give $\mathrm{ESNR}^2 \propto n^{-1/2}$, a 10-fold fall. The signal cannot make up the difference: with $G = K^\top (C + \lambda I)^{-1} K$ the solve gives $\Delta W K = RG(I+G)^{-1}$, whose eigenvalues lie in $[0, 1)$, so $\sum_i \|\Delta W k_i\|^2 \leq \sum_i \|r_i\|^2$, no more than the residuals themselves. The noise variance at each fact's key must therefore grow more slowly than $\|\Delta W\|_F$, as it would in the sub-bin linear regime; one reason is that a batch leaves the linear regime ($|\delta| \ll \Delta$; mean $\bar{\delta}/\Delta$ of 0.36–0.45 at int3 and 0.83–1.04 at int4 at $n = 100$), where the variance of Lemma 1 grows less than linearly in the displacement, and we did not separate other causes; the corollary is stated for rank one only.

D FOUR-CHANNEL DECOMPOSITION WITH CONFIDENCE INTERVALS

Table 11 gives every MEMIT cell behind Figure 1 and the text, with intervals: int3, int4 and NF4 at every batch size on all nine grids (64 cells; GPT-2-medium was run at int3 and int4 only). The channels interact: quantizing the edited matrix on top of the moved key can recover margin, and where it does the conditioning share exceeds 100%; the interaction $D_K + D_W - D_F$ (damages as in §4) is small for single edits and reaches 1.4–1.8 nats (12–26% of D_F) in the three int3 cells where the weight channel costs 1.8–2.6 nats (on GPT-2 at $n = 150$, where the weight channel costs 0.89 nats, the interaction is 0.86 nats, 10% of D_F). Because K/F credits the whole interaction to the conditioning, the table also gives the even split, which halves it. Equivalently $D_K/D_F = 1 - (D_W - D_I)/D_F$ with $D_I = D_K + D_W - D_F$ the interaction, so for single edits, where $D_W \approx 0$, a share near 100% adds only that the interaction is small. The attribution does not rest on measuring in nats: in the 15 MEMIT single-edit cells of the five main models at int3, int4 and NF4, quantizing the edited matrix alone flips the efficacy of none of the 3,000 edits and moves the paraphrase score by at most 0.01, while the keys-only channel lowers efficacy by as much as the full channel to within 0.01 (0.015–0.505 at int3). The share also saturates when the margin reaches a floor: where the keys-only channel already takes the margin to the unedited quantized model's, as on Qwen2.5-3B at int3 (deployed -0.22 against the unedited model's -0.25), the edit is already fully erased and the share is 100% whatever the weight channel would cost on its own (the $n = 100$ cell, whose weight channel alone costs 1.81 nats, has an even split of 94.1%), and no conditioning-share range quoted in §4.1 or the conclusion takes an endpoint from these two cells; Appendix H separates the part of the conditioning loss that is specific to edits. For single edits K/F and the even split agree (91.5–100.5% and 91.7–100.3% over the 26 cells; AWQ, Table 42, 97.9–101.0%); where the weight channel is not negligible they do not: at int3, $n = 1000$ the conditioning share is 83.6–97.6% on the four main models and the even split 70.6–98.0%. Separately, at $n = 1000$ the weight channel is about half of the damage by both measures in two int4/NF4 cells (GPT-2-XL NF4: conditioning share 46.5%, even split 46.1% [32.1, 56.2]; GPT-J int4: 54.4% and 49.4% [31.7, 62.5]).

Independent batches. Every batch cell above is one solve on one batch; the $n = 1000$ cells have no replicate, since the 1,600-fact sample holds only one disjoint block of 1,000. Table 12 adds the next two disjoint blocks of 100 facts of the same sample on the four main models, with MEMIT, KAE and the alignment ridge. Ten of the twelve batches stay where batch 0 is, $\|\Delta W\|_F / \|W\|_F = 0.50$–$0.66$, and in them the conditioning share is 98.9–100.7% at int3, 96.5–101.1% at int4 and 91.8–99.8% at NF4. Batch 1 on GPT-2-XL and on GPT-J drives the update to 4.1 and 3.3 times the matrix's norm (we have not isolated which facts make a batch ill-conditioned): there the weight channel costs 1.8 and 1.3 nats at int3 and the conditioning share falls to 87.0% and 85.9% (74.7–77.6% at int4, 65.9% and

Table 11: Every MEMIT cell at int3, int4 and NF4 on the nine grids (int8 is omitted: the full damage is near zero and the share unstable). fp = unquantized rewrite margin (nats); the next three columns are the change of that margin when only the edited matrix (weights-only), everything except it (keys-only), or everything (full) is quantized, negative = loss, paired 95% CI over the probes (200 at $n \in \{1, 1000\}$ and 100 at $n = 100$ on the main models and ZsRE, 40 on GPT-2 and 20 on GPT-2-medium). K/F is the conditioning share of the main text; "even split" gives each channel half of their interaction, $[D_K + (D_F - D_W)]/(2D_F)$ with D the damages (minus the columns), with probes resampled jointly.

model	quant	n	fp	weights-only	keys-only	full	K/F	even split
GPT-2-XL	int4	1	12.84	−0.01 [−0.02, +0.00]	−1.13 [−1.33, −0.95]	−1.14 [−1.33, −0.96]	99.5%	99.5% [98.6, 100.3]
GPT-2-XL	int3	1	12.84	+0.00 [−0.02, +0.03]	−6.52 [−7.06, −6.00]	−6.49 [−7.03, −5.97]	100.5%	100.3% [99.9, 100.6]
GPT-2-XL	NF4	1	12.84	−0.04 [−0.06, −0.03]	−0.51 [−0.63, −0.39]	−0.55 [−0.68, −0.43]	91.5%	91.7% [89.0, 93.8]
GPT-2-XL	int4	100	13.12	−0.03 [−0.05, −0.01]	−1.11 [−1.34, −0.89]	−1.13 [−1.37, −0.91]	98.0%	97.7% [96.0, 99.3]
GPT-2-XL	int3	100	13.12	−0.04 [−0.08, +0.01]	−6.37 [−7.14, −5.65]	−6.44 [−7.19, −5.71]	99.0%	99.2% [98.6, 99.8]
GPT-2-XL	NF4	100	13.12	−0.04 [−0.06, −0.03]	−0.47 [−0.64, −0.31]	−0.51 [−0.68, −0.35]	91.8%	91.7% [86.6, 95.3]
GPT-2-XL	int4	1000	7.25	−0.81 [−1.03, −0.61]	−1.27 [−1.59, −0.95]	−1.91 [−2.27, −1.55]	66.5%	61.9% [53.0, 70.1]
GPT-2-XL	int3	1000	7.25	−2.56 [−2.92, −2.21]	−5.05 [−5.65, −4.46]	−6.04 [−6.64, −5.45]	83.6%	70.6% [67.7, 73.5]
GPT-2-XL	NF4	1000	7.25	−0.62 [−0.76, −0.48]	−0.53 [−0.78, −0.28]	−1.14 [−1.44, −0.85]	46.5%	46.1% [32.1, 56.2]
Llama-3.2-3B	int4	1	13.73	−0.01 [−0.03, +0.01]	−1.54 [−1.79, −1.29]	−1.55 [−1.80, −1.31]	99.6%	99.5% [98.0, 101.0]
Llama-3.2-3B	int3	1	13.73	−0.05 [−0.09, −0.00]	−6.72 [−7.23, −6.22]	−6.75 [−7.26, −6.23]	99.6%	99.5% [98.8, 100.2]
Llama-3.2-3B	NF4	1	13.73	−0.00 [−0.02, +0.02]	−0.94 [−1.14, −0.74]	−0.94 [−1.15, −0.75]	99.2%	99.5% [97.7, 101.5]
Llama-3.2-3B	int4	100	13.76	−0.02 [−0.05, +0.02]	−1.50 [−1.86, −1.15]	−1.50 [−1.87, −1.14]	100.4%	99.6% [97.5, 102.1]
Llama-3.2-3B	int3	100	13.76	−0.04 [−0.10, +0.02]	−6.94 [−7.61, −6.27]	−7.00 [−7.68, −6.31]	99.1%	99.3% [98.2, 100.4]
Llama-3.2-3B	NF4	100	13.76	−0.03 [−0.06, +0.00]	−1.12 [−1.40, −0.84]	−1.14 [−1.43, −0.87]	97.9%	97.7% [95.1, 100.1]
Llama-3.2-3B	int4	1000	11.50	−0.07 [−0.13, −0.01]	−1.68 [−2.00, −1.37]	−1.77 [−2.09, −1.44]	95.1%	95.6% [92.7, 98.5]
Llama-3.2-3B	int3	1000	11.50	−0.30 [−0.46, −0.14]	−6.55 [−7.15, −5.95]	−6.81 [−7.43, −6.17]	96.3%	95.9% [93.9, 97.9]
Llama-3.2-3B	NF4	1000	11.50	−0.01 [−0.06, +0.03]	−0.91 [−1.15, −0.66]	−0.94 [−1.18, −0.69]	97.1%	97.9% [93.0, 102.6]
GPT-J-6B	int4	1	14.91	−0.00 [−0.02, +0.01]	−0.65 [−0.81, −0.50]	−0.66 [−0.83, −0.51]	98.8%	99.1% [97.3, 101.0]
GPT-J-6B	int3	1	14.91	−0.02 [−0.04, +0.01]	−4.67 [−5.09, −4.25]	−4.70 [−5.13, −4.28]	99.3%	99.5% [98.9, 100.1]
GPT-J-6B	NF4	1	14.91	−0.03 [−0.04, −0.02]	−0.42 [−0.54, −0.31]	−0.45 [−0.57, −0.33]	94.3%	93.9% [90.9, 96.4]
GPT-J-6B	int4	100	15.01	−0.03 [−0.05, −0.01]	−0.66 [−0.86, −0.46]	−0.68 [−0.88, −0.47]	97.7%	96.5% [93.5, 99.3]
GPT-J-6B	int3	100	15.01	−0.04 [−0.09, +0.01]	−4.51 [−5.06, −3.98]	−4.56 [−5.12, −4.02]	99.0%	99.1% [98.1, 100.1]
GPT-J-6B	NF4	100	15.01	−0.02 [−0.04, −0.01]	−0.37 [−0.55, −0.21]	−0.40 [−0.58, −0.23]	94.0%	93.9% [87.1, 98.0]
GPT-J-6B	int4	1000	9.04	−0.62 [−0.80, −0.45]	−0.61 [−0.92, −0.31]	−1.12 [−1.44, −0.81]	54.4%	49.4% [31.7, 62.5]
GPT-J-6B	int3	1000	9.04	−2.41 [−2.77, −2.06]	−5.48 [−6.06, −4.93]	−6.47 [−7.06, −5.89]	84.7%	73.7% [70.5, 76.9]
GPT-J-6B	NF4	1000	9.04	−0.31 [−0.40, −0.21]	−0.74 [−1.02, −0.47]	−1.05 [−1.33, −0.77]	70.8%	70.8% [58.9, 79.7]
Llama-3.1-8B	int4	1	14.80	+0.00 [−0.03, +0.03]	−1.88 [−2.19, −1.57]	−1.89 [−2.20, −1.58]	99.3%	99.7% [98.3, 101.1]
Llama-3.1-8B	int3	1	14.80	−0.09 [−0.14, −0.03]	−8.15 [−8.70, −7.60]	−8.14 [−8.69, −7.59]	100.1%	99.5% [99.0, 100.1]
Llama-3.1-8B	NF4	1	14.80	−0.01 [−0.03, +0.01]	−0.99 [−1.23, −0.74]	−1.02 [−1.27, −0.77]	97.0%	98.0% [96.0, 99.9]
Llama-3.1-8B	int4	100	14.91	−0.01 [−0.05, +0.03]	−1.88 [−2.28, −1.47]	−1.90 [−2.30, −1.49]	99.1%	99.4% [97.7, 101.2]
Llama-3.1-8B	int3	100	14.91	−0.04 [−0.15, +0.05]	−8.05 [−8.85, −7.24]	−8.02 [−8.84, −7.23]	100.4%	99.9% [98.7, 101.1]
Llama-3.1-8B	NF4	100	14.91	−0.03 [−0.07, +0.02]	−1.19 [−1.59, −0.81]	−1.24 [−1.65, −0.86]	95.4%	96.7% [92.9, 99.7]
Llama-3.1-8B	int4	1000	13.34	−0.02 [−0.08, +0.04]	−1.74 [−2.07, −1.42]	−1.75 [−2.07, −1.43]	99.8%	99.4% [96.4, 102.4]
Llama-3.1-8B	int3	1000	13.34	−0.14 [−0.28, −0.01]	−8.09 [−8.73, −7.48]	−8.29 [−8.92, −7.67]	97.6%	98.0% [96.6, 99.3]
Llama-3.1-8B	NF4	1000	13.34	−0.04 [−0.08, +0.01]	−1.06 [−1.38, −0.75]	−1.06 [−1.38, −0.75]	99.9%	98.2% [94.0, 102.3]
Qwen2.5-3B	int4	1	14.56	+0.01 [−0.05, +0.07]	−4.22 [−4.96, −3.51]	−4.23 [−4.97, −3.50]	99.7%	99.9% [98.9, 101.0]
Qwen2.5-3B	int3	1	14.56	−0.09 [−0.22, +0.06]	−14.80 [−15.67, −13.93]	−14.78 [−15.65, −13.91]	100.1%	99.8% [99.3, 100.3]
Qwen2.5-3B	NF4	1	14.56	−0.00 [−0.05, +0.04]	−2.53 [−3.03, −2.03]	−2.56 [−3.07, −2.06]	98.6%	99.2% [97.8, 100.7]
Qwen2.5-3B	int4	100	15.40	−0.61 [−0.89, −0.36]	−4.34 [−5.44, −3.29]	−4.89 [−6.08, −3.75]	88.8%	88.1% [83.8, 91.9]
Qwen2.5-3B	int3	100	15.40	−1.81 [−2.37, −1.28]	−15.28 [−16.53, −14.02]	−15.29 [−16.56, −14.00]	100.0%	94.1% [92.1, 95.9]
Qwen2.5-3B	NF4	100	15.40	−0.02 [−0.16, +0.12]	−2.44 [−3.05, −1.83]	−2.44 [−3.01, −1.86]	100.2%	99.6% [93.5, 105.9]
GPT-2-XL/ZsRE	int4	1	8.63	−0.00 [−0.01, +0.01]	−1.36 [−1.55, −1.17]	−1.36 [−1.56, −1.17]	100.0%	99.9% [99.1, 100.7]
GPT-2-XL/ZsRE	int3	1	8.63	−0.02 [−0.04, +0.00]	−5.53 [−5.95, −5.13]	−5.52 [−5.94, −5.10]	100.1%	99.9% [99.6, 100.2]
GPT-2-XL/ZsRE	NF4	1	8.63	−0.03 [−0.04, −0.02]	−0.60 [−0.75, −0.47]	−0.64 [−0.79, −0.50]	94.6%	95.1% [93.6, 96.4]
GPT-2-XL/ZsRE	int4	100	8.49	−0.00 [−0.02, +0.02]	−1.23 [−1.51, −0.98]	−1.24 [−1.50, −0.98]	99.9%	99.9% [98.4, 101.3]
GPT-2-XL/ZsRE	int3	100	8.49	−0.04 [−0.08, +0.00]	−5.02 [−5.57, −4.48]	−5.07 [−5.62, −4.53]	99.0%	99.1% [98.5, 99.8]
GPT-2-XL/ZsRE	NF4	100	8.49	−0.03 [−0.05, −0.02]	−0.53 [−0.69, −0.38]	−0.57 [−0.73, −0.41]	94.0%	94.2% [91.1, 96.7]
Llama-3.2-3B/ZsRE	int4	1	8.17	−0.01 [−0.02, +0.00]	−0.76 [−0.94, −0.57]	−0.77 [−0.96, −0.58]	98.8%	98.9% [97.2, 100.4]
Llama-3.2-3B/ZsRE	int3	1	8.17	−0.04 [−0.07, −0.01]	−5.02 [−5.48, −4.57]	−5.01 [−5.47, −4.55]	100.2%	99.7% [99.1, 100.3]
Llama-3.2-3B/ZsRE	NF4	1	8.17	−0.00 [−0.01, +0.01]	−0.46 [−0.58, −0.34]	−0.47 [−0.60, −0.35]	97.2%	98.5% [96.4, 100.7]
Llama-3.2-3B/ZsRE	int4	100	8.06	−0.01 [−0.03, +0.02]	−0.77 [−1.00, −0.54]	−0.75 [−0.99, −0.52]	101.6%	100.4% [97.4, 104.2]
Llama-3.2-3B/ZsRE	int3	100	8.06	−0.06 [−0.12, −0.00]	−4.85 [−5.51, −4.21]	−4.96 [−5.64, −4.31]	97.6%	98.2% [97.1, 99.4]
Llama-3.2-3B/ZsRE	NF4	100	8.06	−0.00 [−0.03, +0.02]	−0.42 [−0.56, −0.28]	−0.43 [−0.58, −0.28]	97.3%	98.4% [93.3, 104.5]
GPT-2	int4	1	11.69	−0.03 [−0.06, +0.01]	−1.81 [−2.16, −1.48]	−1.82 [−2.16, −1.49]	99.4%	99.0% [97.3, 100.7]
GPT-2	int3	1	11.69	−0.07 [−0.20, +0.06]	−8.27 [−9.39, −7.21]	−8.44 [−9.59, −7.36]	98.0%	98.6% [97.3, 99.9]
GPT-2	NF4	1	11.69	−0.05 [−0.08, −0.02]	−1.42 [−1.80, −1.07]	−1.46 [−1.84, −1.11]	97.0%	96.6% [94.2, 98.7]
GPT-2	int4	150	10.46	−0.17 [−0.28, −0.06]	−1.91 [−2.35, −1.49]	−1.97 [−2.41, −1.55]	96.6%	94.1% [88.8, 98.8]
GPT-2	int3	150	10.46	−0.89 [−1.20, −0.60]	−8.51 [−9.80, −7.26]	−8.55 [−9.83, −7.33]	99.6%	94.6% [92.2, 97.0]
GPT-2	NF4	150	10.46	−0.05 [−0.14, +0.03]	−1.21 [−1.69, −0.70]	−1.23 [−1.71, −0.72]	98.0%	96.8% [89.0, 103.5]
GPT-2-medium	int4	1	12.53	−0.05 [−0.10, −0.00]	−1.39 [−2.06, −0.74]	−1.44 [−2.13, −0.78]	96.6%	96.6% [93.5, 99.4]
GPT-2-medium	int3	1	12.53	−0.12 [−0.28, +0.03]	−7.68 [−9.18, −6.22]	−7.70 [−9.28, −6.18]	99.7%	99.1% [98.0, 100.3]
GPT-2-medium	int4	30	12.43	−0.01 [−0.06, +0.03]	−1.37 [−2.06, −0.70]	−1.38 [−2.07, −0.69]	99.4%	99.2% [96.6, 102.4]
GPT-2-medium	int3	30	12.43	+0.00 [−0.09, +0.10]	−7.53 [−9.02, −6.14]	−7.48 [−8.96, −6.09]	100.7%	100.4% [99.0, 102.0]

88.7% at NF4). It is the $n = 1000$ exception of the main grid again, reached by one ill-conditioned batch of 100 instead of by batch size, and published MEMIT, with its covariance weight and layer band, shows no such batch (its update, read from the result files, is 0.09–0.13 of the band's weights in Frobenius norm in all nine batches of 100, and its int3 conditioning share 91.7–96.1%, Table 48). KAE's deployed margin exceeds MEMIT's in all 36 cells, with a paired interval above zero in 35 (GPT-J's overloaded batch at NF4 is the exception). Re-running batch 0 from caches rebuilt on the

24 GB card reproduces Table 11's cell on GPT-2-XL, GPT-J and Llama-3.1-8B (fp margin in Table 11 against re-run $13.12/13.13$, $15.01/15.02$ and $14.91/14.89$, deployed $6.69/6.69$, $10.45/10.46$ and $6.89/6.88$, per-probe Pearson 0.999–1.000) and on Llama-3.2-3B to the spread already seen between independent solves (fp 13.76 against 13.55, per-probe Pearson 0.97; Appendix A).

Table 12: The single-layer grid at $n = 100$ over three disjoint batches of the 1,600-fact sample (batch 0 is the main grid's batch). MEMIT's fp and deployed (full) margin, its conditioning share and even split, and the deployed margins of KAE and of the alignment ridge ("ridge"); 100 probes per cell; $\|\Delta W\|_F/\|W\|_F$: MEMIT's update norm relative to the edited matrix; weights-only: MEMIT's margin change with only the edited matrix quantized; KAE−MEMIT: paired difference of deployed margins (nats, 95% CI).

model	quant	batch	$\|\Delta W\|_F/\|W\|_F$	MEMIT fp	weights-only	MEMIT full	K/F	even split	KAE full	KAE−MEMIT	ridge full
GPT-2-XL	int3	0	0.63	13.12	−0.04 [−0.08, +0.01]	6.69	99.0%	99.2%	9.76	+3.07 [+2.61, +3.55]	6.88
GPT-2-XL	int3	1	4.12	11.49	−1.82 [−2.23, −1.43]	3.81	87.0%	81.6%	7.28	+3.47 [+2.97, +4.01]	5.22
GPT-2-XL	int3	2	0.65	12.26	−0.04 [−0.10, +0.01]	6.23	98.9%	99.1%	9.07	+2.84 [+2.46, +3.24]	6.38
GPT-2-XL	int4	0	0.63	13.12	−0.03 [−0.05, −0.01]	11.99	98.0%	97.7%	12.52	+0.53 [+0.41, +0.65]	11.83
GPT-2-XL	int4	1	4.12	11.49	−0.55 [−0.76, −0.37]	9.47	77.6%	75.2%	10.41	+0.94 [+0.60, +1.31]	9.96
GPT-2-XL	int4	2	0.65	12.26	−0.00 [−0.03, +0.02]	11.07	99.6%	99.8%	11.60	+0.53 [+0.36, +0.73]	10.96
GPT-2-XL	NF4	0	0.63	13.12	−0.04 [−0.06, −0.03]	12.61	91.8%	91.7%	12.91	+0.30 [+0.23, +0.38]	12.44
GPT-2-XL	NF4	1	4.12	11.49	−0.36 [−0.48, −0.26]	10.40	65.9%	66.2%	11.21	+0.80 [+0.59, +1.02]	10.72
GPT-2-XL	NF4	2	0.65	12.26	−0.05 [−0.07, −0.03]	11.63	92.6%	92.6%	12.00	+0.37 [+0.26, +0.50]	11.51
Llama-3.2-3B	int3	0	0.64	13.76	−0.04 [−0.10, +0.02]	6.77	99.1%	99.3%	8.83	+2.06 [+1.51, +2.63]	7.06
Llama-3.2-3B	int3	1	0.64	13.33	−0.00 [−0.09, +0.08]	6.91	100.7%	100.3%	8.89	+1.98 [+1.46, +2.49]	7.23
Llama-3.2-3B	int3	2	0.66	12.74	−0.09 [−0.19, +0.00]	5.95	99.1%	98.9%	8.10	+2.15 [+1.60, +2.67]	6.24
Llama-3.2-3B	int4	0	0.64	13.76	−0.02 [−0.05, +0.02]	12.27	100.4%	99.6%	12.62	+0.35 [+0.14, +0.57]	12.27
Llama-3.2-3B	int4	1	0.64	13.33	−0.02 [−0.05, +0.02]	11.71	100.7%	99.7%	12.26	+0.56 [+0.36, +0.75]	11.72
Llama-3.2-3B	int4	2	0.66	12.74	+0.00 [−0.03, +0.04]	11.20	100.1%	100.2%	11.78	+0.59 [+0.39, +0.78]	11.11
Llama-3.2-3B	NF4	0	0.64	13.76	−0.03 [−0.06, +0.00]	12.62	97.9%	97.7%	12.94	+0.32 [+0.16, +0.49]	12.53
Llama-3.2-3B	NF4	1	0.64	13.33	−0.01 [−0.04, +0.02]	12.50	94.3%	96.7%	13.00	+0.51 [+0.36, +0.65]	12.49
Llama-3.2-3B	NF4	2	0.66	12.74	−0.01 [−0.05, +0.02]	11.66	99.8%	99.3%	12.16	+0.50 [+0.33, +0.69]	11.72
GPT-J-6B	int3	0	0.53	15.01	−0.04 [−0.09, +0.01]	10.45	99.0%	99.1%	11.16	+0.71 [+0.42, +1.03]	10.47
GPT-J-6B	int3	1	3.33	13.11	−1.29 [−1.54, −1.04]	7.13	85.9%	82.2%	7.64	+0.51 [+0.01, +1.04]	8.68
GPT-J-6B	int3	2	0.55	13.73	−0.04 [−0.09, +0.02]	9.37	100.4%	99.8%	9.91	+0.54 [+0.28, +0.80]	9.36
GPT-J-6B	int4	0	0.53	15.01	−0.03 [−0.05, −0.01]	14.33	97.7%	96.5%	14.55	+0.21 [+0.13, +0.31]	14.22
GPT-J-6B	int4	1	3.33	13.11	−0.22 [−0.37, −0.08]	12.31	74.7%	73.9%	12.62	+0.31 [+0.10, +0.53]	12.73
GPT-J-6B	int4	2	0.55	13.73	−0.02 [−0.04, −0.00]	12.98	96.5%	96.7%	13.19	+0.20 [+0.11, +0.30]	12.83
GPT-J-6B	NF4	0	0.53	15.01	−0.02 [−0.04, −0.01]	14.61	94.0%	93.9%	14.70	+0.09 [+0.02, +0.15]	14.50
GPT-J-6B	NF4	1	3.33	13.11	−0.08 [−0.15, −0.02]	12.25	88.7%	89.6%	12.34	+0.09 [−0.26, +0.39]	12.71
GPT-J-6B	NF4	2	0.55	13.73	−0.02 [−0.04, −0.00]	13.11	96.9%	96.9%	13.26	+0.14 [+0.08, +0.21]	12.95
Llama-3.1-8B	int3	0	0.50	14.91	−0.04 [−0.15, +0.05]	6.89	100.4%	99.9%	9.61	+2.72 [+2.03, +3.43]	7.04
Llama-3.1-8B	int3	1	0.50	14.24	−0.03 [−0.12, +0.07]	6.61	100.0%	99.8%	8.82	+2.21 [+1.51, +2.93]	6.73
Llama-3.1-8B	int3	2	0.51	13.62	−0.06 [−0.16, +0.03]	6.43	99.9%	99.6%	8.64	+2.21 [+1.46, +2.95]	6.70
Llama-3.1-8B	int4	0	0.50	14.91	−0.01 [−0.05, +0.03]	13.01	99.1%	99.4%	13.71	+0.70 [+0.43, +0.96]	12.87
Llama-3.1-8B	int4	1	0.50	14.24	−0.00 [−0.06, +0.05]	12.45	97.4%	98.6%	12.80	+0.35 [+0.11, +0.61]	12.37
Llama-3.1-8B	int4	2	0.51	13.62	−0.03 [−0.07, +0.02]	12.37	101.1%	99.5%	12.89	+0.52 [+0.21, +0.88]	12.31
Llama-3.1-8B	NF4	0	0.50	14.91	−0.03 [−0.07, +0.02]	13.66	95.4%	96.7%	14.21	+0.55 [+0.35, +0.75]	13.62
Llama-3.1-8B	NF4	1	0.50	14.24	−0.03 [−0.07, +0.01]	13.39	97.5%	96.9%	13.79	+0.41 [+0.14, +0.67]	13.23
Llama-3.1-8B	NF4	2	0.51	13.62	−0.02 [−0.06, +0.02]	12.65	98.8%	98.4%	13.14	+0.49 [+0.21, +0.77]	12.68

Ten independent batches. Table 13 extends this to ten disjoint batches of 100 per model on GPT-2-XL, Llama-3.2-3B and Llama-3.1-8B (batches 3–9 re-run with a rebuilt key/value cache and the same code, MEMIT and KAE; on Llama-3.1-8B at int3 and int4 only), and pools each model's ten with a two-stage bootstrap that resamples batches and then probes within them. Composition matters where the table above said it does. On Llama-3.2-3B every batch keeps the update below the matrix's own norm ($\|\Delta W\|_F/\|W\|_F = 0.64$–$0.66$) and MEMIT's conditioning share is 96.5–100.7% in every batch at int3 (pooled 99.1% $[98.2, 100.0]$), and 97.4–102.9% at int4; on Llama-3.1-8B the update stays at 0.50–0.51 of the norm and the share is 98.4–100.7% at int3 (pooled 99.5% $[98.8, 100.2]$) and 97.4–101.1% at int4 (pooled 99.3%). On GPT-2-XL four of the ten batches exceed it (1.01, 1.12, 3.85 and 4.12 times the matrix's norm) and there the conditioning share falls to 85.2–95.2% at int3 and 75.7–97.7% at int4, against 98.7–99.8% and 97.9–100.7% in the other six; across the ten batches it falls as the norm ratio rises (Spearman -0.75 at int3). The damage itself varies between batches more than a one-batch interval admits: the two-stage interval of GPT-2-XL's pooled int3 damage is $[5.74, 6.82]$ against $[6.01, 6.48]$ from the probes alone. Resampling probes inside resampled batches counts the within-batch variance twice, so where batches differ little the two-stage intervals are conservative (up to $\sqrt{2}$ too wide when they do not differ at all): on the two Llamas the ten batch means vary no more than probe sampling predicts, and a t interval on them (9 degrees of freedom) is as narrow as the probe-only one ($[6.49, 6.92]$ against $[6.48, 6.93]$ at int3 on Llama-3.2-3B), while on GPT-2-XL, whose batches do differ, it is $[5.64, 6.85]$, slightly wider than the two-stage $[5.74, 6.82]$. KAE returns 45.3% $[43.0, 47.4]$ of MEMIT's int3 damage on GPT-2-XL, 31.4% $[28.6, 34.4]$ on Llama-3.2-3B and 31.3% $[28.1, 34.5]$ on Llama-3.1-8B, pooled over the batches. The batch-level claim is therefore conditional on the norm ratio rather than on n (a criterion drawn after seeing these batches): where an unweighted single-layer solve keeps $\|\Delta W\|_F$ below $\|W\|_F$ the conditioning

carries 96.5–102.9% of the loss of a batch of 100 at int3 and int4 (87.1–99.8% at NF4), and where it does not the weight channel can take a visible share, as at $n = 1000$ on GPT-2-XL and GPT-J (not on the Llamas, whose $n = 1000$ updates at 2.2 and 1.7 times the norm keep it under 5%, while Qwen2.5-3B's $n = 100$ update at 2.1 times the norm gives it 5.9% at int3 and 11.9% at int4 by the even split): above the norm the ratio flags a batch for measurement rather than predicting the share.

Table 13: Ten disjoint batches of 100 facts per model (batch 0 is the main grid's), MEMIT, single layer. Damage: fp minus deployed margin, mean over batches with the range; two-stage CI resamples batches and then probes within them, probe-only CI treats the pooled 1,000 probes as independent. K/F (range over batches) and even split are pooled over the batches (two-stage CI); $\|\Delta W\|_F/\|W\|_F$ range over batches; KAE: share of MEMIT's deployed damage it returns on the same batches, pooled, two-stage CI.

model	quant	batches	damage (range)	two-stage CI	probe-only CI	K/F [CI]	K/F range	even split [CI]	$\|\Delta W\|_F/\|W\|_F$	KAE rec.
GPT-2-XL	int3	10	6.25 (5.32–7.77)	[5.74, 6.82]	[6.01, 6.48]	95.2% [91.4, 98.6]	85.2–99.8%	93.7% [88.5, 98.5]	0.63–4.12	45% [43, 47]
GPT-2-XL	int4	10	1.23 (0.78–2.02)	[1.00, 1.50]	[1.14, 1.32]	91.3% [84.3, 98.8]	75.7–100.7%	91.1% [83.7, 98.9]	0.63–4.12	42% [38, 47]
GPT-2-XL	NF4	10	0.63 (0.38–1.08)	[0.48, 0.80]	[0.56, 0.70]	84.5% [76.5, 92.8]	65.9–96.1%	84.7% [76.8, 92.9]	0.63–4.12	58% [46, 70]
Llama-3.2-3B	int3	10	6.70 (6.04–7.08)	[6.41, 6.98]	[6.48, 6.93]	99.1% [98.2, 100.0]	96.5–100.7%	99.1% [98.4, 99.7]	0.64–0.66	31% [29, 34]
Llama-3.2-3B	int4	10	1.50 (1.21–1.70)	[1.36, 1.64]	[1.39, 1.62]	99.9% [98.8, 101.1]	97.4–102.9%	99.5% [98.6, 100.5]	0.64–0.66	38% [32, 44]
Llama-3.2-3B	NF4	10	0.93 (0.75–1.14)	[0.82, 1.04]	[0.84, 1.01]	96.5% [94.7, 98.1]	94.2–99.8%	97.3% [96.1, 98.5]	0.64–0.66	46% [38, 53]
Llama-3.1-8B	int3	10	7.92 (7.18–8.48)	[7.57, 8.26]	[7.65, 8.19]	99.5% [98.8, 100.2]	98.4–100.7%	99.3% [98.8, 99.7]	0.50–0.51	31% [28, 35]
Llama-3.1-8B	int4	10	1.72 (1.25–1.99)	[1.52, 1.93]	[1.57, 1.88]	99.3% [98.3, 100.4]	97.4–101.1%	99.2% [98.4, 100.0]	0.50–0.51	36% [29, 44]

A 13B model. Every model of the main grid has at most 8B parameters. Table 14 runs the main grid's protocol on the base model Llama-2-13B (40 layers, $d_{\text{in}} = 13{,}824$), editing layer 6 (depth fraction 0.15, between Llama-3.2-3B's $4/28$ and Llama-3.1-8B's $5/32$), with the unweighted C and an fp16 backbone. The attribution is the one found up to 8B: over the six MEMIT cells the conditioning share is 93.4–102.5% (even split 93.5–102.1%) and the weights-only channel costs at most 0.04 nats. At int3 a single edit loses 4.11 of its 9.70-nat margin, all of it through the conditioning, while the key at the subject moves by 0.74 of its norm (0.36 at int4, 0.29 at NF4). At $n = 100$ the update is 0.48 of the matrix in Frobenius norm, just below the range of the batches above that are not overloaded (0.50–0.66), and the split is the same (99.6% at int3). KAE recovers 25–61% of MEMIT's deployed damage, 25–26% at int3, against 25–52% for the key-aware half on the four main models (§4.5). Solved inside the int3 model on 100 of the facts, QAE recovers 106% [93, 119] of MEMIT's deployed damage for single edits (deployed margin 9.81 against MEMIT's 5.33 and unquantized 9.56; its own unquantized margin is 4.00), against 90–100% on the four main models. MEMIT's fp margin on this model is lower than on the five main models at $n \leq 100$ (9.5–9.7 nats against 12.84–15.40 in Table 11, the base models GPT-2-XL and GPT-J-6B included), for a reason we have not isolated, and one model at this size checks the attribution; it is not a scaling law.

Table 14: Llama-2-13B, the single-layer grid of the main text at layer 6 (unweighted C, fp16 backbone; 200 probes at $n = 1$, the first batch of 100 at $n = 100$). Margin changes in nats relative to the fp margin (negative = margin lost), with paired 95% intervals; $\|\delta k\|/\|k^*\|$: key displacement; KAE recovered: share of MEMIT's deployed damage KAE removes on the same facts. In a separate deployment-precision run on 100 of the facts ($n = 1$, int3), QAE recovers 106% [93, 119] of MEMIT's deployed damage.

n	quant	fp	weights-only	keys-only	full	K/F	even split	$\|\delta k\|/\|k^*\|$	KAE recovered
1	int3	9.70	-0.04 [$-0.07, -0.01$]	-4.11 [$-4.51, -3.72$]	-4.11 [$-4.50, -3.73$]	100.0%	99.5% [99.0, 100.1]	0.74	26% [21, 30]
1	int4	9.70	-0.01 [$-0.02, -0.00$]	-0.84 [$-1.02, -0.66$]	-0.84 [$-1.02, -0.67$]	99.6%	99.2% [98.1, 100.3]	0.36	45% [36, 57]
1	NF4	9.70	-0.03 [$-0.04, -0.02$]	-0.42 [$-0.56, -0.30$]	-0.45 [$-0.59, -0.32$]	93.4%	93.5% [90.6, 95.5]	0.29	61% [47, 83]
100	int3	9.45	-0.02 [$-0.05, +0.02$]	-4.22 [$-4.81, -3.65$]	-4.23 [$-4.82, -3.67$]	99.6%	99.6% [98.8, 100.3]	0.75	25% [17, 32]
100	int4	9.45	$+0.01$ [$-0.00, +0.03$]	-0.69 [$-0.91, -0.48$]	-0.67 [$-0.89, -0.46$]	102.5%	102.1% [99.9, 104.8]	0.36	53% [37, 77]
100	NF4	9.45	-0.03 [$-0.04, -0.01$]	-0.51 [$-0.72, -0.33$]	-0.54 [$-0.75, -0.35$]	95.6%	95.4% [92.2, 97.6]	0.29	44% [30, 66]

Key displacement. Mean $\|\delta k\|/\|k^*\|$ under a quantized backbone (MEMIT, $n = 1$): int8 0.015, 0.038, 0.015, 0.042, 0.025; NF4 0.218, 0.347, 0.220, 0.378, 0.250; int4 0.266, 0.421, 0.272, 0.457, 0.434; int3 0.593, 0.832, 0.643, 0.911, 1.684 for GPT-2-XL, Llama-3.2-3B, GPT-J-6B, Llama-3.1-8B and Qwen2.5-3B (so 1.5–4.2% of the norm at int8); GPT-2 and GPT-2-medium sit at 0.674 and 0.670 at int3.

Within one cell, magnitude orders the weight channel across editors solved at the same key, and not across editors solved at different keys. On Qwen2.5-3B at $n = 100$/int3 the weight-channel cost of the four editors solved at the fp key tracks $\|\Delta W\|_F/\|W\|_F$ (ratio and weights-only margin

change: alignment ridge 0.498, -0.25; MEMIT 2.133, -1.81; KRE 2.217, -1.55; AlphaEdit 4.091, -2.51 nats; Spearman of the ratio against the loss $+0.800$ over four points, descriptive only), whereas KAE and KAE+KRE at 8.090 and 8.413 have a weight channel that is *positive* and significant ($+0.42$, $+0.60$): the weights-only channel serves a KAE edit on an unquantized backbone, i.e. at the key it was deliberately not solved for, and quantizing its own matrix moves the delivered update back toward what it was fitted for. Over all six editors, ranking the signed cost (a gain counts as negative cost), the rank correlation is -0.486: across keys the ordering inverts.

E THE QUANTIZERS THAT ARE SERVED: ZERO-POINT GROUPS OF 128, GPTQ AND AWQ

The main grid quantizes with symmetric absmax groups of 64. Serving stacks ship int4 differently: every linear layer of every block, a zero point per group, groups of 128, and GPTQ or AWQ rather than rounding (AutoGPTQ, AutoAWQ and llm-compressor's W4A16 schemes). We use zero-point groups, AutoAWQ's default and a common GPTQ setting (AutoGPTQ and llm-compressor default to symmetric groups). The zero point is an integer, so the zero-point grid contains 0 and the phase of Lemma 1 is $\varphi_{ij} = (W_{ij}/\Delta_{ij} + \frac{1}{2}) \bmod 1$ with the zero-point step; an update that moves a group's minimum or maximum breaks (i). `code/prodquant.py` implements the three at 3 and 4 bits and plugs them into the same four-channel grid and the same QAE runs.

- **RTN-b**: scale $(\max - \min)/(2^b - 1)$ and zero point $\mathrm{round}(-\min/\mathrm{scale})$ per group of 128 input entries (where the input width is not a multiple of 128, as GPT-2-XL's 1600, the last group is the remaining 64 entries with its own scale and zero point, in all three quantizers), levels $0, \ldots, 2^b - 1$ (AutoAWQ's pseudo-quantizer; 8 levels at 3 bits and 16 at 4, against the symmetric grid's 7 and 15).
- **GPTQ-b**: AutoGPTQ's procedure with `desc_act=True, static_groups=True`: blocks in order, each calibrated on the output of the already-quantized blocks below it; true-sequential inside a block (q, k, v | o | gate, up | down; on GPT-2-XL `c_attn` | `attn.c_proj` | `c_fc` | `mlp.c_proj`); damping 1% of the mean Hessian diagonal (the damped inverse Hessian's Cholesky factor computed in fp64 and stored in fp32); lazy updates in blocks of 128 columns; columns in act-order; each group's scale and zero point from the unquantized weights, with 0 kept inside the range. Without act-order, GPTQ breaks Llama-3.2-3B: in block 1's down projection, whose input carries Llama's massive activations, the error feedback rewrites the "super weight" (Yu et al., 2024), and quantizing that one matrix alone raises Llama-3.2-3B's perplexity on 8,000 test tokens from 23.1 to 62.7 at 4 bits (23.4 with act-order and static groups, 24.0 with rounding).
- **AWQ-b**: AutoAWQ's procedure: activation statistics from the unquantized model; per group of layers sharing an input (q, k, v; o only where v and o have the same shape, which excludes the grouped-query attention of the Llamas and Qwen2.5-3B; gate and up; down; on GPT-2-XL each of `c_attn`, `attn.c_proj`, `c_fc` and `mlp.c_proj` alone), a 20-point grid over the exponent of the duo scaling $s = \bar{x}^r / \bar{w}^{1-r}$, normalized by $\sqrt{\max s \min s}$; then the clipping search (ten shrink ratios, 1.00 to 0.55 in steps of 0.05, scored with the second moment of every calibration token where AutoAWQ subsamples 512; not on q, k or GPT-2's fused qkv; a ratio below 1.00 clips a group's extreme entries in W and $W + \Delta W$ alike, outside assumption (iii)); then zero-point rounding. The deployed matrix in the model's coordinates is $\mathcal{Q}(\mathrm{clip}(W \,\mathrm{diag}\, s))\,\mathrm{diag}(s)^{-1}$ (s is folded into the preceding operation in a released checkpoint, which leaves the network's function unchanged: folding into a preceding linear scales its rows, to which zero-point rounding in row groups is equivariant, and where the preceding operation is an activation AutoAWQ keeps the scale as a separate floating-point operation, as we do). The scale search scores a candidate by the output error of the group's own linears under the full calibration second moment, where AutoAWQ uses the enclosing attention or MLP block; for the down projection, the edited matrix, the two coincide.

Calibration is 128 sequences of 512 tokens of WikiText-103 train (the first 128 consecutive 512-token chunks of its paragraphs of at least 200 characters, in the dataset's order), and perplexity is measured on WikiText-103 test, which flatters GPTQ and AWQ, calibrated on the same corpus, more than rounding, which uses no data. The formulas were checked against the published ones before any run (zero-point rounding equals AutoAWQ's bit for bit; GPTQ with a diagonal Hessian equals zero-point rounding; on a correlated synthetic input GPTQ halves and AWQ cuts by a third the output error of rounding; `code/test_prodquant.py`); no checkpoint written by AutoGPTQ, AutoAWQ or

llm-compressor themselves, and no integer kernel, was evaluated, so "served" results throughout are those of these reimplementations. The edited matrix $W + \Delta W$ is re-quantized per edit by the same algorithm with the same calibration statistics: GPTQ with the Hessian of that matrix's input as the sequential pass saw it, AWQ with its scale and clip searched again. That is what re-running the toolchain on the edited model does to the edited matrix; the blocks above it are calibrated on the unedited model, which a single edit barely changes on generic text (it moves perplexity by at most 0.70% at the symmetric grid's int8, int4 and int3, Appendix V); we keep that for batches of 100 as well (which move int3 perplexity 1.01–$1.06\times$ on the four main models, Table 53; not measured on Qwen2.5-3B) and do not re-calibrate the upper blocks on the edited model, so the served batch cells of Qwen2.5-3B, the one served model whose batch of 100 is overloaded, rest on this untested simplification.

Results. Table 15 gives every cell (MEMIT, $n = 1$ and the first batch of 100, 100 probes each) and Table 16 the deployment-precision editors on the same probes. The main grid's attribution holds under every production quantizer on all four models run here (GPT-2-XL, the two Llamas and Qwen2.5-3B). For single edits the weight channel is at most 0.35 nats and the conditioning share 90.2–100.9% over the 24 cells (even split 90.3–100.5%, every 95% interval within 86.1–102.7%). In the batches of 100 the share is 88.5–102.0% and the weight channel at most 0.26 nats on GPT-2-XL and the two Llamas; on Qwen2.5-3B the share is 86.0–99.0% and the weight channel reaches 1.17 (RTN-3) and 0.61 and 1.47 nats (GPTQ-4, GPTQ-3), as in its main-grid batches (MEMIT's is 1.81 nats at $n = 100$/int3). The quantizers differ in how much they damage and much less in where. Zero-point rounding loses 1.0–1.7 nats at 4 bits and 4.2–6.9 at 3 on GPT-2-XL and the Llamas, and 4.3 and 15.8 on Qwen2.5-3B, whose perplexity RTN-3 raises 714-fold; for single edits its weight channel is at most 0.10 nats and its ESNR calibration (ratio of cell means) 0.99–1.00 on all four models, Qwen2.5-3B included (0.987 and 0.991 at 4 and 3 bits), whose symmetric-grid single edits sit at 0.89–0.91 (Appendix B). GPTQ and AWQ cut the model's perplexity excess over rounding (on WikiText-103, their calibration corpus) by a factor of 1.9–4.0 at 3 bits and 1.5–3.4 at 4 bits (on Qwen2.5-3B at 3 bits, from 714-fold to 1.23- and 1.34-fold), but cut the edit's loss by only 8–36% at 3 bits on GPT-2-XL and the Llamas and change it by -14 to $+9\%$ at 4 bits on the Llamas, every paired interval covering zero (loss against rounding: GPTQ $+0.05$ $[-0.25, +0.36]$ and $+0.14$ $[-0.29, +0.58]$ nats, AWQ -0.18 $[-0.46, +0.10]$ and $+0.04$ $[-0.44, +0.56]$, on Llama-3.2-3B and Llama-3.1-8B; the intervals admit cuts of up to 18–36% of rounding's loss, each below the cut the same quantizer makes in that model's perplexity excess, 35–65%): the calibration objective of both is the output error on generic text, and an edit's key is not generic text. At 4 bits on GPT-2-XL and Qwen2.5-3B they cut the edit's loss by 44–54% and 41–52%, against perplexity-excess cuts of 46–71% and 48–68% (at 3 bits Qwen2.5-3B's rounding reference has a 714-fold perplexity, so no relative cut is quoted). GPTQ's single-edit weight channel stays small (≤ 0.21 nats) while its calibration ratio rises to 1.25 (ratio of means 1.003, 1.099, 1.157 and 1.178 at 4 bits and 1.025, 1.157, 1.247 and 1.190 at 3 bits on GPT-2-XL, Llama-3.2-3B, Llama-3.1-8B and Qwen2.5-3B), because its error feedback moves later columns to compensate each rounding error, so $\mathcal{Q}(W + \Delta W) - \mathcal{Q}(W)$ is no longer entrywise rounding of ΔW; AWQ's is larger (0.02–0.35 nats, calibration 1.009, 0.983, 1.006 and 0.878 at 4 bits and 1.078, 1.111, 1.156 and 0.909 at 3 bits, same order), because re-searching the scales and the clip on $W + \Delta W$ moves the grid itself. Of the conditioning, the layers below the edit carry more on GPT-2-XL (except under GPTQ-4; the two are equal under RTN-4 for single edits) and those above it on the two Llamas and Qwen2.5-3B. KAE, which re-solves at the key only, returns 21–64% of single-edit damage on GPT-2-XL and the Llamas and 3–9% on Qwen2.5-3B; QAE returns 81–128% and 62–93%, with intervals that are wide at 4 bits, where the loss is under 2.6 nats.

Weight-activation quantization. Serving stacks also quantize activations. We implement W8A8 (`quantize_backbone_wa` in `code/quant.py`) with SmoothQuant's smoothing (Xiao et al., 2023) and the per-channel weight, per-token activation scales of common int8 kernels: per-input-channel activation maxima from 32 sequences of 512 WikiText-103 train tokens; the smoothing scale $s_j = \max|x_j|^{\alpha} / \max|W_{:,j}|^{1-\alpha}$ with $\alpha = 0.5$ for every model (SmoothQuant tunes it per model and also quantizes the attention matmuls, which we do not; the weight maximum taken over the linears that share the input) folded into the linears that read a LayerNorm output (q, k, v and gate, up; GPT-2's `c_attn` and `c_fc`); int8 weights with one symmetric absmax scale per output channel; and int8 activations with one symmetric absmax scale per token, without clipping, at the input of every linear. The output and down projections are not smoothed, so the edited matrix's input, the key, is rounded per token as it is. W4A8 keeps the per-token int8 activations, without smoothing, and takes the zero-point group-128 RTN-4 weights above; it is a plain combination for this test, not a

published W4A8 recipe. The channels are defined as before with the activation rounding assigned to the conditioning: keys-only quantizes every weight but the edited matrix's and every activation, the key included; weights-only quantizes the edited matrix's weights alone. Table 17 gives MEMIT on GPT-2-XL, Llama-3.2-3B and Llama-3.1-8B. W8A8 costs the unedited model at most 2% in perplexity and an edit 0.07–0.34 nats (at most 2.2% of its margin; no efficacy lost), too little for the conditioning share to be estimated precisely (point estimates 73–141%). W4A8 costs an edit 1.20–2.22 nats, 0.15–0.58 more than the same weights with unquantized activations (Table 15), with no efficacy lost, at most 4 paraphrases, and 97–108% in the conditioning (97–104% for single edits): the weight channel is at most 0.01 nats, and the rounding of the key is one more displacement of the key, not a rounding of the update.

Table 15: Production quantizers: zero-point groups of 128 over every linear layer of the blocks, round-to-nearest (RTN), and our GPTQ and AWQ following AutoGPTQ and AutoAWQ (Appendix E); single-layer MEMIT, the first 100 probes ($n = 1$) or the whole first batch ($n = 100$), from key/value caches rebuilt for these runs; fp margins differ from Table 11's by up to 0.3 nats at $n = 100$ and by up to 1.05 at $n = 1$, where that table averages 200 probes, whose first 100 have a mean fp margin 0.26–1.22 nats above the 200-probe mean. ppl: WikiText-103 test perplexity (20,000 tokens) of the quantized unedited model (unquantized: 21.07, 18.29, 13.47 and 10.46 on GPT-2-XL, Llama-3.2-3B, Llama-3.1-8B and Qwen2.5-3B). Channel columns: margin change against the unquantized edit (nats, paired 95% CI). ES and PS: unquantized/deployed success rate, in parentheses the number of edits that succeed unquantized and fail deployed. KAE and QAE: share of MEMIT's deployed damage they return (same probes; –: not run, Table 16).

model	quant	n	ppl	fp	weights-only	keys-only	full	K/F	even split	ES	PS	KAE	QAE
GPT-2-XL	RTN-4	1	21.51	13.21	-0.00 $[-0.02, +0.01]$	-1.04 $[-1.23, -0.86]$	-1.04 $[-1.23, -0.86]$	99.7%	99.7%	1.00/1.00 (0)	0.93/0.90 (3)	35%	105% [76, 136]
GPT-2-XL	GPTQ-4	1	21.20	13.21	-0.02 $[-0.03, -0.00]$	-0.46 $[-0.62, -0.30]$	-0.48 $[-0.65, -0.32]$	96.2%	96.2%	1.00/1.00 (0)	0.93/0.91 (2)	39%	128% [78, 194]
GPT-2-XL	AWQ-4	1	21.31	13.21	-0.06 $[-0.07, -0.04]$	-0.53 $[-0.71, -0.35]$	-0.58 $[-0.78, -0.40]$	90.2%	90.3%	1.00/1.00 (0)	0.93/0.90 (3)	64%	81% [31, 136]
GPT-2-XL	RTN-3	1	23.56	13.21	-0.02 $[-0.05, +0.00]$	-4.17 $[-4.71, -3.66]$	-4.19 $[-4.73, -3.67]$	99.6%	99.5%	1.00/1.00 (0)	0.93/0.83 (11)	48%	98% [89, 109]
GPT-2-XL	GPTQ-3	1	22.37	13.21	-0.06 $[-0.09, -0.04]$	-2.62 $[-3.13, -2.14]$	-2.68 $[-3.18, -2.20]$	97.9%	97.8%	1.00/1.00 (0)	0.93/0.84 (9)	44%	104% [90, 119]
GPT-2-XL	AWQ-3	1	22.20	13.21	-0.24 $[-0.30, -0.19]$	-2.69 $[-3.13, -2.29]$	-2.98 $[-3.44, -2.54]$	90.5%	91.1%	1.00/1.00 (0)	0.93/0.81 (13)	51%	93% [80, 106]
GPT-2-XL	RTN-4	100	21.51	13.13	-0.01 $[-0.03, +0.01]$	-1.10 $[-1.30, -0.91]$	-1.11 $[-1.31, -0.92]$	99.0%	99.0%	1.00/1.00 (0)	0.91/0.87 (4)	36%	96% [67, 125]
GPT-2-XL	GPTQ-4	100	21.20	13.13	-0.02 $[-0.03, -0.00]$	-0.49 $[-0.65, -0.33]$	-0.50 $[-0.67, -0.34]$	98.0%	97.3%	1.00/1.00 (0)	0.91/0.87 (4)	37%	117% [69, 180]
GPT-2-XL	AWQ-4	100	21.31	13.13	-0.07 $[-0.09, -0.05]$	-0.53 $[-0.72, -0.35]$	-0.60 $[-0.78, -0.42]$	88.5%	88.4%	1.00/1.00 (0)	0.91/0.87 (5)	60%	84% [38, 135]
GPT-2-XL	RTN-3	100	23.56	13.13	-0.04 $[-0.08, +0.01]$	-4.18 $[-4.73, -3.66]$	-4.20 $[-4.75, -3.67]$	99.6%	99.4%	1.00/0.97 (3)	0.91/0.82 (11)	41%	88% [79, 98]
GPT-2-XL	GPTQ-3	100	22.37	13.13	-0.09 $[-0.13, -0.05]$	-2.72 $[-3.24, -2.23]$	-2.83 $[-3.37, -2.33]$	96.2%	96.5%	1.00/0.99 (1)	0.91/0.83 (8)	44%	98% [84, 113]
GPT-2-XL	AWQ-3	100	22.20	13.13	-0.26 $[-0.32, -0.21]$	-2.68 $[-3.13, -2.26]$	-2.96 $[-3.44, -2.51]$	90.6%	90.9%	1.00/1.00 (0)	0.91/0.80 (12)	44%	87% [73, 102]
Llama-3.2-3B	RTN-4	1	20.07	14.03	-0.01 $[-0.05, +0.03]$	-1.23 $[-1.56, -0.90]$	-1.27 $[-1.61, -0.95]$	96.6%	97.8%	1.00/1.00 (0)	0.96/0.96 (1)	50%	104% [67, 143]
Llama-3.2-3B	GPTQ-4	1	19.07	14.03	-0.01 $[-0.04, +0.02]$	-1.31 $[-1.57, -1.04]$	-1.32 $[-1.60, -1.06]$	98.7%	99.0%	1.00/1.00 (0)	0.96/0.95 (1)	25%	84% [50, 116]
Llama-3.2-3B	AWQ-4	1	19.21	14.03	-0.04 $[-0.07, -0.00]$	-1.05 $[-1.30, -0.80]$	-1.09 $[-1.35, -0.84]$	96.4%	96.6%	1.00/1.00 (0)	0.96/0.94 (2)	21%	90% [51, 130]
Llama-3.2-3B	RTN-3	1	28.93	14.03	-0.05 $[-0.10, +0.00]$	-5.55 $[-6.18, -4.95]$	-5.58 $[-6.20, -4.97]$	99.5%	99.3%	1.00/0.97 (3)	0.96/0.91 (7)	39%	90% [80, 99]
Llama-3.2-3B	GPTQ-3	1	22.45	14.03	-0.06 $[-0.11, -0.01]$	-5.03 $[-5.60, -4.47]$	-5.02 $[-5.60, -4.48]$	100.1%	99.5%	1.00/0.97 (3)	0.96/0.89 (8)	29%	99% [89, 109]
Llama-3.2-3B	AWQ-3	1	23.44	14.03	-0.13 $[-0.18, -0.07]$	-3.66 $[-4.13, -3.21]$	-3.83 $[-4.31, -3.36]$	95.6%	96.2%	1.00/1.00 (0)	0.96/0.95 (3)	31%	89% [78, 100]
Llama-3.2-3B	RTN-4	100	20.07	13.76	$+0.01$ $[-0.03, +0.06]$	-1.19 $[-1.54, -0.84]$	-1.17 $[-1.52, -0.83]$	101.0%	101.1%	1.00/1.00 (0)	0.96/0.95 (1)	40%	94% [55, 136]
Llama-3.2-3B	GPTQ-4	100	19.07	13.76	$+0.01$ $[-0.03, +0.06]$	-1.22 $[-1.49, -0.96]$	-1.20 $[-1.46, -0.94]$	101.9%	101.5%	1.00/1.00 (0)	0.96/0.96 (0)	26%	97% [59, 136]
Llama-3.2-3B	AWQ-4	100	19.21	13.76	-0.02 $[-0.06, +0.01]$	-1.01 $[-1.27, -0.76]$	-1.01 $[-1.27, -0.77]$	99.5%	98.5%	1.00/1.00 (0)	0.96/0.95 (1)	16%	84% [40, 132]
Llama-3.2-3B	RTN-3	100	28.93	13.76	-0.03 $[-0.08, +0.02]$	-5.15 $[-5.81, -4.52]$	-5.15 $[-5.80, -4.50]$	100.1%	99.7%	1.00/0.96 (4)	0.96/0.92 (6)	30%	80% [69, 91]
Llama-3.2-3B	GPTQ-3	100	22.45	13.76	-0.06 $[-0.13, -0.00]$	-4.92 $[-5.50, -4.36]$	-4.93 $[-5.50, -4.38]$	99.7%	99.2%	1.00/0.96 (4)	0.96/0.92 (5)	29%	97% [86, 109]
Llama-3.2-3B	AWQ-3	100	23.44	13.76	-0.09 $[-0.15, -0.02]$	-3.56 $[-4.04, -3.11]$	-3.66 $[-4.13, -3.20]$	97.3%	97.4%	1.00/1.00 (0)	0.96/0.94 (4)	28%	85% [73, 97]
Llama-3.1-8B	RTN-4	1	14.86	15.33	$+0.00$ $[-0.04, +0.04]$	-1.58 $[-2.03, -1.14]$	-1.57 $[-2.01, -1.12]$	100.9%	100.5%	1.00/1.00 (0)	0.95/0.95 (0)	–	–
Llama-3.1-8B	GPTQ-4	1	13.95	15.33	-0.01 $[-0.04, +0.03]$	-1.69 $[-2.02, -1.37]$	-1.71 $[-2.04, -1.39]$	99.1%	99.3%	1.00/1.00 (0)	0.95/0.95 (0)	24%	83% [52, 118]
Llama-3.1-8B	AWQ-4	1	14.37	15.33	-0.02 $[-0.05, +0.02]$	-1.58 $[-1.98, -1.19]$	-1.60 $[-2.02, -1.21]$	98.5%	98.7%	1.00/1.00 (0)	0.95/0.95 (0)	27%	92% [52, 134]
Llama-3.1-8B	RTN-3	1	25.46	15.33	-0.10 $[-0.17, -0.03]$	-6.83 $[-7.60, -6.09]$	-6.85 $[-7.62, -6.10]$	99.7%	99.1%	1.00/0.98 (2)	0.95/0.91 (6)	–	–
Llama-3.1-8B	GPTQ-3	1	16.49	15.33	-0.05 $[-0.13, +0.03]$	-6.31 $[-6.99, -5.67]$	-6.33 $[-6.98, -5.68]$	99.8%	99.5%	1.00/0.98 (2)	0.95/0.94 (3)	31%	98% [88, 109]
Llama-3.1-8B	AWQ-3	1	18.30	15.33	-0.14 $[-0.20, -0.07]$	-4.93 $[-5.64, -4.24]$	-5.16 $[-5.86, -4.45]$	95.6%	96.5%	1.00/1.00 (0)	0.95/0.96 (0)	34%	90% [78, 102]
Llama-3.1-8B	RTN-4	100	14.86	15.19	$+0.01$ $[-0.02, +0.05]$	-1.72 $[-2.19, -1.26]$	-1.69 $[-2.16, -1.23]$	102.0%	101.4%	1.00/1.00 (0)	0.95/0.95 (0)	–	–
Llama-3.1-8B	GPTQ-4	100	13.95	15.19	-0.01 $[-0.06, +0.03]$	-1.68 $[-2.02, -1.35]$	-1.68 $[-2.02, -1.36]$	99.7%	99.4%	1.00/1.00 (0)	0.95/0.95 (0)	22%	81% [48, 116]
Llama-3.1-8B	AWQ-4	100	14.37	15.19	-0.01 $[-0.05, +0.03]$	-1.51 $[-1.91, -1.13]$	-1.55 $[-1.96, -1.17]$	97.0%	98.2%	1.00/1.00 (0)	0.95/0.95 (0)	17%	86% [44, 130]
Llama-3.1-8B	RTN-3	100	25.46	15.19	-0.03 $[-0.10, +0.04]$	-6.75 $[-7.54, -5.99]$	-6.86 $[-7.67, -6.10]$	98.4%	99.0%	1.00/0.95 (5)	0.95/0.89 (8)	–	–
Llama-3.1-8B	GPTQ-3	100	16.49	15.19	-0.01 $[-0.09, +0.08]$	-6.25 $[-6.91, -5.61]$	-6.33 $[-6.98, -5.68]$	98.7%	99.3%	1.00/0.98 (2)	0.95/0.92 (5)	33%	92% [82, 103]
Llama-3.1-8B	AWQ-3	100	18.30	15.19	-0.16 $[-0.25, -0.09]$	-4.80 $[-5.50, -4.11]$	-5.08 $[-5.79, -4.37]$	94.5%	95.6%	1.00/0.99 (1)	0.95/0.95 (1)	27%	82% [69, 96]
Qwen2.5-3B	RTN-4	1	11.96	15.61	-0.09 $[-0.17, -0.01]$	-4.24 $[-5.08, -3.44]$	-4.25 $[-5.10, -3.45]$	99.7%	98.8%	1.00/0.95 (5)	0.91/0.89 (5)	–	–
Qwen2.5-3B	GPTQ-4	1	10.94	15.61	-0.08 $[-0.17, +0.00]$	-2.41 $[-3.00, -1.85]$	-2.51 $[-3.12, -1.92]$	96.1%	96.4%	1.00/0.99 (1)	0.91/0.89 (3)	3%	93% [55, 132]
Qwen2.5-3B	AWQ-4	1	11.24	15.61	-0.04 $[-0.13, +0.06]$	-1.98 $[-2.61, -1.37]$	-2.05 $[-2.71, -1.43]$	96.8%	97.5%	1.00/0.98 (2)	0.91/0.90 (2)	9%	62% [21, 103]
Qwen2.5-3B	RTN-3	1	7466.93	15.61	-0.08 $[-0.24, +0.07]$	-15.76 $[-17.04, -14.50]$	-15.78 $[-17.05, -14.50]$	99.9%	99.7%	1.00/0.51 (49)	0.91/0.47 (48)	–	–
Qwen2.5-3B	GPTQ-3	1	12.89	15.61	-0.21 $[-0.38, -0.03]$	-8.62 $[-9.64, -7.62]$	-8.73 $[-9.74, -7.72]$	98.7%	98.2%	1.00/0.89 (11)	0.91/0.82 (9)	4%	77% [65, 88]
Qwen2.5-3B	AWQ-3	1	13.97	15.61	-0.35 $[-0.55, -0.18]$	-6.55 $[-7.42, -5.72]$	-6.73 $[-7.58, -5.88]$	97.4%	96.1%	1.00/0.93 (7)	0.91/0.87 (6)	3%	86% [72, 100]
Qwen2.5-3B	RTN-4	100	11.96	15.32	-0.17 $[-0.34, +0.01]$	-3.91 $[-4.76, -3.08]$	-4.30 $[-5.15, -3.46]$	91.0%	93.5%	0.99/0.93 (6)	0.91/0.87 (5)	–	–
Qwen2.5-3B	GPTQ-4	100	10.94	15.32	-0.61 $[-0.83, -0.39]$	-2.26 $[-2.85, -1.67]$	-2.63 $[-3.25, -2.03]$	86.0%	81.5%	0.99/0.98 (1)	0.91/0.89 (3)	2%	70% [32, 107]
Qwen2.5-3B	AWQ-4	100	11.24	15.32	-0.15 $[-0.24, -0.08]$	-1.83 $[-2.42, -1.27]$	-1.91 $[-2.49, -1.36]$	95.9%	93.9%	0.99/0.98 (1)	0.91/0.91 (3)	4%	57% [9, 105]
Qwen2.5-3B	RTN-3	100	7466.93	15.32	-1.17 $[-1.56, -0.80]$	-15.45 $[-16.80, -14.11]$	-15.61 $[-16.95, -14.27]$	99.0%	95.8%	0.99/0.44 (55)	0.91/0.45 (49)	–	–
Qwen2.5-3B	GPTQ-3	100	12.89	15.32	-1.47 $[-1.93, -1.01]$	-8.25 $[-9.29, -7.25]$	-8.94 $[-10.01, -7.91]$	92.3%	87.9%	0.99/0.83 (16)	0.91/0.81 (11)	6%	55% [43, 65]
Qwen2.5-3B	AWQ-3	100	13.97	15.32	-0.29 $[-0.46, -0.13]$	-6.42 $[-7.31, -5.56]$	-6.59 $[-7.50, -5.71]$	97.4%	96.5%	0.99/0.93 (6)	0.91/0.86 (7)	5%	82% [67, 98]

Table 16: Editing at deployment precision under the production quantizers (same models, probes and batches as Table 15). MEMIT's fp and deployed margin; upstream and downstream: MEMIT's margin change with only the layers below, or only those above, the edited matrix quantized; KAE and value: share of MEMIT's deployed damage that KAE and value-aware editing return; QAE: its deployed margin (nats); QAE rec.: the share it returns (in sample; held-out paraphrases, 3 bits only: Table 21); ΔES, ΔPS, ΔNS: QAE against MEMIT, deployed, paired 95% CI, bold if it excludes zero. KAE, value-aware editing and QAE were not run on the zero-point rounding cells of Llama-3.1-8B and Qwen2.5-3B (MEMIT's are in Table 15).

model	quant	n	fp	MEMIT	upstream	downstream	KAE	value	QAE	QAE rec.	ΔES	ΔPS	ΔNS
GPT-2-XL	RTN-4	1	13.21	12.17	−0.53 [−0.67, −0.40]	−0.53 [−0.67, −0.39]	35%	66%	13.27	105% [76, 136]	+0.000 [+0.000, +0.000]	+0.000 [+0.000, +0.000]	+0.000 [+0.000, +0.000]
GPT-2-XL	GPTQ-4	1	13.21	12.74	−0.19 [−0.32, −0.08]	−0.28 [−0.40, −0.15]	39%	79%	13.35	128% [78, 194]	+0.000 [+0.000, +0.000]	+0.000 [−0.040, +0.040]	+0.000 [+0.000, +0.000]
GPT-2-XL	AWQ-4	1	13.21	12.63	−0.42 [−0.56, −0.31]	−0.11 [−0.24, +0.01]	64%	17%	13.11	81% [31, 136]	+0.000 [+0.000, +0.000]	+0.020 [+0.000, +0.050]	−0.010 [−0.030, +0.000]
GPT-2-XL	RTN-3	1	13.21	9.02	−2.77 [−3.24, −2.33]	−1.76 [−2.04, −1.49]	48%	36%	13.15	98% [89, 109]	+0.000 [+0.000, +0.000]	**+0.100** [+0.050, +0.160]	−0.010 [−0.030, +0.000]
GPT-2-XL	GPTQ-3	1	13.21	10.53	−1.77 [−2.15, −1.41]	−1.15 [−1.43, −0.87]	44%	51%	13.33	104% [90, 119]	+0.000 [+0.000, +0.000]	**+0.060** [+0.010, +0.120]	+0.000 [+0.000, +0.000]
GPT-2-XL	AWQ-3	1	13.21	10.24	−1.78 [−2.13, −1.46]	−1.03 [−1.30, −0.77]	51%	37%	13.00	93% [80, 106]	+0.000 [+0.000, +0.000]	**+0.090** [+0.030, +0.160]	+0.000 [+0.000, +0.000]
GPT-2-XL	RTN-4	100	13.13	12.01	−0.57 [−0.73, −0.42]	−0.55 [−0.69, −0.41]	36%	53%	13.08	96% [67, 125]	+0.000 [+0.000, +0.000]	+0.020 [+0.000, +0.050]	−0.030 [−0.070, +0.000]
GPT-2-XL	GPTQ-4	100	13.13	12.63	−0.20 [−0.32, −0.08]	−0.30 [−0.42, −0.18]	37%	72%	13.21	117% [69, 180]	+0.000 [+0.000, +0.000]	+0.020 [−0.030, +0.070]	+0.000 [+0.000, +0.000]
GPT-2-XL	AWQ-4	100	13.13	12.53	−0.41 [−0.55, −0.30]	−0.12 [−0.25, +0.00]	60%	18%	13.03	84% [38, 135]	+0.000 [+0.000, +0.000]	+0.030 [+0.000, +0.070]	+0.020 [+0.000, +0.050]
GPT-2-XL	RTN-3	100	13.13	8.93	−2.79 [−3.30, −2.32]	−1.76 [−2.05, −1.47]	41%	34%	12.62	88% [79, 98]	+0.030 [+0.000, +0.070]	**+0.070** [+0.030, +0.120]	+0.030 [−0.010, +0.070]
GPT-2-XL	GPTQ-3	100	13.13	10.29	−1.83 [−2.22, −1.46]	−1.13 [−1.42, −0.85]	44%	46%	13.08	98% [84, 113]	+0.010 [+0.000, +0.030]	+0.040 [+0.000, +0.090]	+0.000 [+0.000, +0.000]
GPT-2-XL	AWQ-3	100	13.13	10.16	−1.79 [−2.14, −1.46]	−1.08 [−1.33, −0.82]	44%	37%	12.75	87% [73, 102]	+0.000 [+0.000, +0.000]	**+0.090** [+0.030, +0.160]	+0.020 [−0.020, +0.060]
Llama-3.2-3B	RTN-4	1	14.03	12.76	−0.47 [−0.67, −0.27]	−0.76 [−1.01, −0.50]	50%	48%	14.09	104% [67, 143]	+0.000 [+0.000, +0.000]	−0.010 [−0.040, +0.020]	−0.020 [−0.060, +0.020]
Llama-3.2-3B	GPTQ-4	1	14.03	12.71	−0.43 [−0.57, −0.29]	−0.92 [−1.13, −0.71]	25%	49%	13.81	84% [50, 116]	+0.000 [+0.000, +0.000]	+0.020 [+0.000, +0.050]	+0.000 [−0.030, +0.030]
Llama-3.2-3B	AWQ-4	1	14.03	12.94	−0.24 [−0.38, −0.10]	−0.79 [−1.01, −0.58]	21%	58%	13.92	90% [51, 130]	+0.000 [+0.000, +0.000]	+0.010 [−0.030, +0.050]	+0.000 [+0.000, +0.000]
Llama-3.2-3B	RTN-3	1	14.03	8.45	−3.34 [−3.94, −2.76]	−3.56 [−4.05, −3.09]	39%	29%	13.45	90% [80, 99]	+0.030 [+0.000, +0.070]	+0.050 [+0.000, +0.100]	−0.010 [−0.040, +0.020]
Llama-3.2-3B	GPTQ-3	1	14.03	9.01	−2.40 [−2.88, −1.95]	−3.78 [−4.30, −3.28]	29%	49%	13.97	99% [89, 109]	+0.030 [+0.000, +0.070]	+0.060 [+0.000, +0.120]	+0.010 [−0.020, +0.050]
Llama-3.2-3B	AWQ-3	1	14.03	10.20	−1.72 [−2.07, −1.40]	−2.39 [−2.81, −1.99]	31%	40%	13.59	89% [78, 100]	+0.000 [+0.000, +0.000]	+0.000 [−0.040, +0.040]	+0.010 [+0.000, +0.030]
Llama-3.2-3B	RTN-4	100	13.76	12.58	−0.42 [−0.66, −0.19]	−0.75 [−1.02, −0.49]	40%	46%	13.68	94% [55, 136]	+0.000 [+0.000, +0.000]	+0.000 [−0.030, +0.030]	−0.040 [−0.090, +0.000]
Llama-3.2-3B	GPTQ-4	100	13.76	12.56	−0.42 [−0.58, −0.27]	−0.86 [−1.07, −0.66]	26%	57%	13.72	97% [59, 136]	+0.000 [+0.000, +0.000]	+0.010 [+0.000, +0.030]	−0.020 [−0.070, +0.030]
Llama-3.2-3B	AWQ-4	100	13.76	12.75	−0.17 [−0.33, −0.01]	−0.79 [−1.02, −0.58]	16%	56%	13.60	84% [40, 132]	+0.000 [+0.000, +0.000]	+0.000 [−0.040, +0.040]	**−0.050** [−0.100, −0.010]
Llama-3.2-3B	RTN-3	100	13.76	8.61	−3.01 [−3.66, −2.40]	−3.51 [−4.01, −3.02]	30%	30%	12.74	80% [69, 91]	**+0.040** [+0.010, +0.080]	+0.030 [−0.010, +0.080]	−0.050 [−0.110, +0.000]
Llama-3.2-3B	GPTQ-3	100	13.76	8.82	−2.27 [−2.76, −1.81]	−3.79 [−4.31, −3.29]	29%	51%	13.62	97% [86, 109]	**+0.040** [+0.010, +0.080]	+0.050 [+0.000, +0.100]	+0.000 [−0.050, +0.050]
Llama-3.2-3B	AWQ-3	100	13.76	10.09	−1.47 [−1.81, −1.14]	−2.42 [−2.87, −2.00]	28%	39%	13.21	85% [73, 97]	+0.000 [+0.000, +0.000]	+0.030 [−0.020, +0.080]	−0.020 [−0.060, +0.020]
Llama-3.1-8B	GPTQ-4	1	15.33	13.63	−0.38 [−0.61, −0.15]	−1.29 [−1.57, −1.03]	24%	66%	15.05	83% [52, 118]	+0.000 [+0.000, +0.000]	+0.000 [−0.050, +0.050]	−0.010 [−0.050, +0.030]
Llama-3.1-8B	AWQ-4	1	15.33	13.73	−0.54 [−0.79, −0.30]	−1.14 [−1.49, −0.81]	27%	59%	15.21	92% [52, 134]	+0.000 [+0.000, +0.000]	+0.010 [−0.030, +0.050]	−0.020 [−0.050, +0.000]
Llama-3.1-8B	GPTQ-3	1	15.33	9.01	−3.81 [−4.46, −3.15]	−4.44 [−4.98, −3.91]	31%	41%	15.22	98% [88, 109]	+0.020 [+0.000, +0.050]	+0.020 [−0.030, +0.080]	+0.010 [+0.000, +0.030]
Llama-3.1-8B	AWQ-3	1	15.33	10.18	−2.93 [−3.53, −2.35]	−3.78 [−4.42, −3.17]	34%	31%	14.81	90% [78, 102]	+0.000 [+0.000, +0.000]	+0.010 [−0.040, +0.060]	−0.010 [−0.040, +0.020]
Llama-3.1-8B	GPTQ-4	100	15.19	13.51	−0.39 [−0.65, −0.15]	−1.29 [−1.55, −1.03]	22%	63%	14.88	81% [48, 116]	+0.000 [+0.000, +0.000]	+0.020 [−0.020, +0.060]	−0.020 [−0.070, +0.030]
Llama-3.1-8B	AWQ-4	100	15.19	13.64	−0.39 [−0.64, −0.17]	−1.12 [−1.48, −0.78]	17%	56%	14.98	86% [44, 130]	+0.000 [+0.000, +0.000]	+0.020 [−0.020, +0.060]	+0.010 [−0.030, +0.050]
Llama-3.1-8B	GPTQ-3	100	15.19	8.86	−3.73 [−4.38, −3.09]	−4.30 [−4.85, −3.78]	33%	41%	14.70	92% [82, 103]	+0.020 [+0.000, +0.050]	+0.050 [−0.010, +0.110]	+0.000 [−0.030, +0.030]
Llama-3.1-8B	AWQ-3	100	15.19	10.12	−2.78 [−3.37, −2.21]	−3.68 [−4.30, −3.08]	27%	35%	14.30	82% [69, 96]	+0.010 [+0.000, +0.030]	+0.030 [−0.020, +0.080]	+0.010 [−0.050, +0.070]
Qwen2.5-3B	GPTQ-4	1	15.61	13.10	−0.14 [−0.31, +0.03]	−2.45 [−3.05, −1.87]	3%	89%	15.45	93% [55, 132]	+0.010 [+0.000, +0.030]	**+0.080** [+0.020, +0.140]	+0.000 [−0.030, +0.030]
Qwen2.5-3B	AWQ-4	1	15.61	13.57	−0.32 [−0.53, −0.12]	−1.65 [−2.21, −1.11]	9%	58%	14.83	62% [21, 103]	+0.010 [−0.020, +0.040]	−0.010 [−0.090, +0.070]	+0.010 [+0.000, +0.030]
Qwen2.5-3B	GPTQ-3	1	15.61	6.89	−1.63 [−2.34, −0.99]	−8.02 [−8.92, −7.15]	4%	65%	13.60	77% [65, 88]	**+0.080** [+0.020, +0.150]	+0.070 [+0.000, +0.150]	−0.020 [−0.050, +0.000]
Qwen2.5-3B	AWQ-3	1	15.61	8.89	−1.83 [−2.54, −1.14]	−5.79 [−6.61, −4.96]	3%	68%	14.64	86% [72, 100]	**+0.070** [+0.020, +0.120]	+0.070 [+0.000, +0.150]	+0.010 [−0.020, +0.050]
Qwen2.5-3B	GPTQ-4	100	15.32	12.69	−0.13 [−0.35, +0.08]	−2.32 [−2.89, −1.76]	2%	70%	14.53	70% [32, 107]	+0.020 [+0.000, +0.050]	+0.040 [−0.020, +0.100]	+0.010 [−0.030, +0.050]
Qwen2.5-3B	AWQ-4	100	15.32	13.41	−0.38 [−0.65, −0.11]	−1.51 [−2.04, −1.02]	4%	52%	14.51	57% [9, 105]	+0.010 [−0.020, +0.040]	−0.010 [−0.080, +0.060]	**+0.040** [+0.010, +0.080]
Qwen2.5-3B	GPTQ-3	100	15.32	6.38	−1.63 [−2.42, −0.90]	−7.71 [−8.60, −6.81]	6%	45%	11.25	55% [43, 65]	**+0.120** [+0.040, +0.200]	+0.000 [−0.080, +0.080]	−0.010 [−0.060, +0.040]
Qwen2.5-3B	AWQ-3	100	15.32	8.73	−2.05 [−2.73, −1.44]	−5.62 [−6.45, −4.81]	5%	61%	14.14	82% [67, 98]	**+0.060** [+0.020, +0.110]	+0.050 [−0.030, +0.130]	−0.010 [−0.060, +0.040]

Table 17: Weight-activation quantization, MEMIT: W8A8 with SmoothQuant smoothing ($\alpha = 0.5$; int8 weights per output channel, int8 activations per token, the SmoothQuant scale on the linears that read a LayerNorm output, the edited down-projection's input, the key, quantized as it is) and W4A8 (the zero-point group-128 RTN-4 weights of Table 15 with the same int8 activations). keys-only quantizes every weight but the edited matrix's and every activation, the key included; weights-only quantizes the edited matrix's weights alone. ppl: ratio of WikiText-103 perplexity to the unquantized model's; margin changes in nats (paired 95% CI); flips: edits whose efficacy / paraphrase success is lost in deployment; $\|\delta k\|/\|k^*\|$: the key's displacement. 100 probes ($n = 1$) or the first batch of 100.

model	quant	n	ppl	fp	weights-only	keys-only	full	K/F	flips	$\|\delta k\|/\|k^*\|$
GPT-2-XL	W8A8	1	1.00	13.21	-0.00 $[-0.00, +0.00]$	-0.05 $[-0.12, +0.00]$	-0.07 $[-0.13, -0.02]$	73.5%	0/0	0.071
GPT-2-XL	W8A8	100	1.00	13.13	-0.00 $[-0.00, +0.00]$	-0.08 $[-0.14, -0.03]$	-0.11 $[-0.16, -0.05]$	78.5%	0/0	0.071
GPT-2-XL	W4A8	1	1.02	13.21	-0.00 $[-0.02, +0.01]$	-1.17 $[-1.37, -0.98]$	-1.20 $[-1.40, -1.01]$	97.2%	0/3	0.261
GPT-2-XL	W4A8	100	1.02	13.13	-0.01 $[-0.03, +0.01]$	-1.23 $[-1.46, -1.02]$	-1.26 $[-1.49, -1.05]$	97.7%	0/4	0.261
Llama-3.2-3B	W8A8	1	1.01	14.03	-0.01 $[-0.02, +0.01]$	-0.19 $[-0.28, -0.09]$	-0.16 $[-0.25, -0.08]$	112.5%	0/1	0.167
Llama-3.2-3B	W8A8	100	1.01	13.76	-0.00 $[-0.02, +0.02]$	-0.14 $[-0.26, -0.03]$	-0.14 $[-0.25, -0.02]$	104.4%	0/0	0.167
Llama-3.2-3B	W4A8	1	1.11	14.03	-0.01 $[-0.05, +0.03]$	-1.50 $[-1.88, -1.12]$	-1.45 $[-1.83, -1.08]$	103.4%	0/2	0.439
Llama-3.2-3B	W4A8	100	1.11	13.76	$+0.01$ $[-0.03, +0.06]$	-1.53 $[-1.93, -1.14]$	-1.47 $[-1.87, -1.08]$	103.9%	0/1	0.439
Llama-3.1-8B	W8A8	1	1.02	15.33	$+0.00$ $[-0.02, +0.03]$	-0.46 $[-0.65, -0.28]$	-0.33 $[-0.51, -0.15]$	141.2%	0/0	0.222
Llama-3.1-8B	W8A8	100	1.02	15.19	$+0.03$ $[+0.00, +0.05]$	-0.43 $[-0.61, -0.25]$	-0.34 $[-0.53, -0.15]$	126.0%	0/0	0.222
Llama-3.1-8B	W4A8	1	1.14	15.33	$+0.00$ $[-0.04, +0.04]$	-2.23 $[-2.69, -1.77]$	-2.15 $[-2.64, -1.68]$	103.7%	0/1	0.499
Llama-3.1-8B	W4A8	100	1.14	15.19	$+0.01$ $[-0.02, +0.05]$	-2.40 $[-2.90, -1.90]$	-2.22 $[-2.70, -1.76]$	108.0%	0/0	0.499

F EDITING AT DEPLOYMENT PRECISION

Each editor is best where it was solved. Each editor's margin is largest in the channel it was solved for (MEMIT in fp, KAE upstream, value-aware editing downstream, QAE in the full channel), and the paired interval against its next-best channel excludes zero in 44 of the 48 editor–cell comparisons on the four main models and ZsRE at int3, $n \leq 100$. The exceptions are value-aware editing, whose downstream margin ties its full-channel margin on Llama-3.2-3B ($+0.14$ $[-0.25, +0.54]$ at $n = 1$, $+0.29$ $[-0.19, +0.77]$ at $n = 100$) and GPT-J-6B ($+0.16$ $[-0.11, +0.44]$, $+0.31$ $[-0.09, +0.70]$), since its residual also carries part of the key shift. On Qwen2.5-3B three of the eight editor–cell comparisons invert.

§3.3 substitutes two quantities into the same closed form. k_q^* is one forward pass through the quantized backbone. v_q^* comes from the same optimization the editor already runs (25 Adam steps on a change δ to the edited down-projection's output at the subject's last token, with ROME's KL term and norm clamp; Appendix A), with the forward passes taken through the quantized backbone and the edited matrix itself held at full precision, so $v_q^* = Wk_q^* + \delta$ and the residual $v_q^* - Wk_q^*$ of §3.3 is exactly the optimized δ; the edited matrix's own rounding at deployment is the weight channel, which QAE leaves alone. No weight gradients are involved, so the quantizer is a constant and nothing is straight-through; the KL term's reference distribution and the clamp's reference norm are those of the unedited model with this quantized backbone (the clamp is $4\|Wk_q^*\|$), and gradients reach δ through the dequantized weights, stored in the checkpoint's 16-bit format (fp16 or bf16), with unquantized activations; we do not test backpropagation through the integer kernels of an on-device stack, the setting in which Lu et al. (2026) report it failing. The two are cached once per (model, quantizer) and reused by every batch size. QAE therefore costs one forward pass plus the editor's own value optimization run once more per served precision, on the dequantized weights of the served quantizer: for the same 200 facts on the same GPU, back to back, the full-precision values took 1715, 1192 and 1516 seconds (Llama-3.1-8B on CounterFact, GPT-2-XL and Llama-3.2-3B on ZsRE) and the deployment-precision values 1736, 1194 and 1519, a ratio of 1.002–1.012. Value-aware editing solves with k^* and residual $v_q^* - Wk^* = \delta + W(k_q^* - k^*)$, so its residual carries the key's displacement through W and the key/value split below is not a clean one; KAE, QAE and value-aware editing all keep the fp second moment C, and re-estimating it on quantized keys is untested. Table 18 reports all four editors in the fp, upstream, downstream and full channels at int3, and Tables 19 and 20 the deployed damage of each against MEMIT's full-precision margin, with deployed ES, PS and NS and the paired ΔPS and ΔNS against MEMIT. QAE's weights-only channel is not measured (its matrix-rounding cost on the quantized backbone is, at $n = 1000$ below). Neither QAE nor value-aware editing is combined with the alignment ridge or with the matched-control spreading of Appendix S (QAE is run on the released bands of Appendix T). QAE's transfer to a precision other than the one it was solved for is measured only against fp: its unquantized margin is 21–57% of

MEMIT's on the four main models ($6.59/12.84$, $4.04/13.69$, $8.50/14.91$, $3.13/14.80$ at $n = 1$) and negative on Qwen2.5-3B; Appendix P runs the full solve×deploy matrix for KAE.

What the objective says about the deployment. The value of the v^* objective's likelihood term after optimization (the new object's mean per-token negative log-probability over the four prefixed prompts at the last step; the KL term is not included) is a diagnostic of whether the deployed precision can host the edit, available once v_q^* is computed and before anything is deployed. Its mean over the 200 single-edit probes, full precision → int3, is $0.033 \to 0.020$ on GPT-2-XL, $0.001 \to 0.002$ on Llama-3.2-3B, $0.019 \to 0.021$ on GPT-J-6B, $0.001 \to 0.001$ on Llama-3.1-8B and $0.076 \to 5.42$ on Qwen2.5-3B ($0.008 \to 5.51$ on the first 100 facts, which form the $n = 100$ batch; one of the other 100 is at 11.3 already at full precision); at int4 and NF4 it is at most 0.031 on the two models run (GPT-2-XL and Llama-3.2-3B). The one model on which it rises by more than an order of magnitude is the one on which the fix returns 49% of the loss at a locality cost. That is one failure among five models, so the diagnostic is a post-hoc observation with a natural threshold (an order of magnitude), not a validated rule. Since v_q^* is computed per fact and shared by every batch size, it cannot flag the failure at $n = 1000$ on GPT-2-XL, which is the rounding of QAE's own update (below).

Margins QAE's objective does not see. v_q^* is optimized on the rewrite prompt with four fixed prefixes; the two paraphrase prompts and the three neighbourhood prompts of each record never enter it, so their margins are held out for QAE as they are for MEMIT. Table 21 (`code/qae_heldout.py`) computes "recovered" on the paraphrase margin and the paired change of the neighbourhood margin. On the four main models at int3, $n = 1$, QAE returns 72–95% of MEMIT's paraphrase-margin loss ($+1.7$ to $+3.1$ nats, every interval above zero) against 90–100% of the rewrite-margin loss it optimizes; under the served 3-bit GPTQ and AWQ it returns 66–124% on GPT-2-XL and the Llamas and 36% and 47% on Qwen2.5-3B (every gain above zero), and on ZsRE 109–110%. The held-out share is lower than the in-sample one in most cells (by about half on the served Qwen2.5-3B) but of the same order, so the fix is not a fit to its own prompt; the paraphrases name the same subject, though, so they test the edit at nearby keys in other contexts, not on other facts. It costs neighbourhood margin: 0.09–0.24 nats at int3 on the four models (every interval below zero), too little to move the neighbourhood score by more than 0.03, and 4.59 nats on Qwen2.5-3B (paraphrase share 87%), the locality collapse of §4.5. Net of that cost (paraphrase-margin gain minus neighbourhood-margin loss, over the paraphrase-margin loss) the held-out share is 66–92% on the four models and 15% on Qwen2.5-3B. Over all 55 deployment-precision cells (the 19 CounterFact cells of Tables 19 and 20, the four ZsRE cells and the 32 served GPTQ and AWQ cells of Table 16, whose eight zero-point rounding cells are left out, before the fresh-fact replication; `code/qae_heldout.py` prints every cell with its intervals) the paraphrase-margin gain excludes zero in 40, and the neighbourhood-margin change is below zero in 14 and above it in 4.

QAE's matrix-rounding cost at $n = 1000$. Table 18 reads QAE in the fp, upstream, downstream and full channels, so the 2.56-nat weight channel quoted in §4.1 for $n = 1000$ is MEMIT's weights-only damage. Evaluating QAE's own update with only the backbone quantized (keys-only) and with its edited matrix quantized on top (full), on the same 200 probes of GPT-2-XL at int3 and a rebuilt cache (margins: fp 2.41, keys-only 6.33, full 4.23; MEMIT 7.25, 2.21, 1.21), quantizing the edited matrix on top of the quantized backbone (full minus keys-only, which includes the interaction, and not the weights-only damage of §4.1) costs QAE 2.10 nats $[1.78, 2.43]$, against 0.99 $[0.76, 1.23]$ for MEMIT: at $n = 1000$ it is the rounding of QAE's own update that is material: it accounts for 2.10 of the 3.02 nats by which QAE's deployed margin falls short of MEMIT's unquantized one (50% recovered), and the conditioning for the other 0.92. Leaving the edited matrix in floating point, as Lu et al. (2026) do, is this keys-only channel, so at $n = 1000$ that deployment keeps 2.10 nats more than QAE's fully quantized one; QAE's keys-only channel was not run at $n \leq 100$.

Intervals on "recovered", and what MEMIT keeps behaviourally. Table 22 puts a bootstrap interval on QAE's "recovered" in Tables 2, 19, 20 and 29 and replaces the percentages above 100% by what they mean: QAE's deployed margin minus MEMIT's unquantized one. Where the damage is small the percentages are loose (GPT-2-XL NF4, $n = 100$: 102% [48, 165]); where it is large they are tight (GPT-2-XL int3, $n = 1$: 100% [95, 105]). The overshoots on ZsRE are real but small ($+0.45$ and $+0.36$ nats); the others cover zero. Its intervals are resampled separately from those of Tables 19 and 20, so where a bound sits within 0.01 of zero the two tables can disagree on its side. Table 23 gives MEMIT's efficacy and paraphrase success before and after quantization: at int3 and $n = 1$ efficacy stays at 0.915–0.985 on the four main models while paraphrase success

falls by 0.025–0.160, so in behaviour int3 mostly weakens edits rather than erasing them, except on Qwen2.5-3B ($0.995 \to 0.490$) and at $n = 1000$ ($0.930 \to 0.635$ on GPT-2-XL).

Table 18: Every editor in the fp, upstream, downstream and full channels: mean rewrite margin in nats (200 probes at $n = 1$, 100 at $n = 100$, 200 at $n = 1000$), int3. "up" quantizes only the matrices that move the key, "down" only those after the edited layer, "full" is the deployed model with the edited matrix quantized as well. At $n \leq 100$ each editor is largest in the channel it was solved for (point estimates; paired intervals exclude zero in 44 of the 48 editor–cell comparisons outside Qwen2.5-3B, the exceptions being +value against the full channel on Llama-3.2-3B and GPT-J-6B, e.g. $+0.14$ $[-0.25, +0.54]$ nats on Llama-3.2-3B at $n = 1$), except on Qwen2.5-3B, where the fix fails and three of the eight cells invert; at $n = 1000$ on GPT-2-XL, QAE is largest in the upstream channel. This deployment-precision run is a separate solve from Table 11, so a shared MEMIT cell can differ from it by up to 0.21 nats on CounterFact; its ZsRE cells are drawn from a different probe set (Appendix A).

model	editor	n	fp	up	down	full
GPT-2-XL	MEMIT	1	12.84	7.86	10.29	6.36
GPT-2-XL	+key	1	9.92	12.05	8.05	9.76
GPT-2-XL	+value	1	9.49	6.20	11.30	7.96
GPT-2-XL	+both	1	6.59	10.77	8.14	12.85
GPT-2-XL	MEMIT	100	13.13	8.29	10.70	6.69
GPT-2-XL	+key	100	9.50	12.04	7.80	9.75
GPT-2-XL	+value	100	9.85	6.54	11.53	8.24
GPT-2-XL	+both	100	6.30	10.70	7.61	12.53
GPT-2-XL	MEMIT	1000	7.26	2.74	5.81	1.22
GPT-2-XL	+key	1000	3.26	6.35	2.62	3.11
GPT-2-XL	+value	1000	5.58	2.35	6.18	1.45
GPT-2-XL	+both	1000	2.41	5.68	2.68	4.22
Llama-3.2-3B	MEMIT	1	13.69	10.08	9.03	7.06
Llama-3.2-3B	+key	1	9.47	13.81	6.64	9.18
Llama-3.2-3B	+value	1	6.48	5.75	9.62	9.48
Llama-3.2-3B	+both	1	4.04	8.44	6.25	13.20
Llama-3.2-3B	MEMIT	100	13.55	9.94	9.07	6.89
Llama-3.2-3B	+key	100	8.79	13.40	6.21	8.95
Llama-3.2-3B	+value	100	7.16	6.29	9.90	9.61
Llama-3.2-3B	+both	100	4.31	8.88	6.14	12.97
GPT-J-6B	MEMIT	1	14.91	12.89	12.30	10.19
GPT-J-6B	+key	1	12.96	14.36	10.72	11.35
GPT-J-6B	+value	1	10.32	9.97	12.89	12.73
GPT-J-6B	+both	1	8.50	11.61	10.92	14.46
GPT-J-6B	MEMIT	100	15.02	12.91	12.54	10.42
GPT-J-6B	+key	100	13.11	13.98	10.91	11.16
GPT-J-6B	+value	100	10.38	9.98	12.92	12.60
GPT-J-6B	+both	100	8.40	11.46	10.79	14.00
Llama-3.1-8B	MEMIT	1	14.80	9.00	9.28	6.65
Llama-3.1-8B	+key	1	7.38	15.22	5.80	9.47
Llama-3.1-8B	+value	1	7.38	5.75	10.72	8.76
Llama-3.1-8B	+both	1	3.13	9.26	5.16	14.17
Llama-3.1-8B	MEMIT	100	14.89	8.96	9.52	6.88
Llama-3.1-8B	+key	100	6.84	14.93	5.46	9.61
Llama-3.1-8B	+value	100	7.75	5.86	11.27	8.92
Llama-3.1-8B	+both	100	3.17	9.14	5.25	13.64
Qwen2.5-3B	MEMIT	1	14.56	−0.27	5.07	−0.22
Qwen2.5-3B	+key	1	−5.12	0.79	−3.26	0.53
Qwen2.5-3B	+value	1	−2.16	−0.23	−1.09	−0.16
Qwen2.5-3B	+both	1	−5.20	3.24	−3.26	6.95
Qwen2.5-3B	MEMIT	100	15.40	−0.20	5.96	0.11
Qwen2.5-3B	+key	100	−2.46	0.01	−1.76	0.19
Qwen2.5-3B	+value	100	−1.99	−0.22	−1.01	0.14
Qwen2.5-3B	+both	100	−3.95	0.45	−2.46	1.74
GPT-2-XL/ZsRE	MEMIT	1	8.41	4.44	6.03	3.25
GPT-2-XL/ZsRE	+key	1	5.86	7.62	4.21	5.44
GPT-2-XL/ZsRE	+value	1	4.68	3.09	6.75	4.91
GPT-2-XL/ZsRE	+both	1	2.89	5.99	4.41	8.86
GPT-2-XL/ZsRE	MEMIT	100	8.23	4.73	5.87	3.45
GPT-2-XL/ZsRE	+key	100	4.53	7.31	3.30	5.18
GPT-2-XL/ZsRE	+value	100	4.73	3.37	6.60	5.20
GPT-2-XL/ZsRE	+both	100	2.28	5.66	3.43	8.23
Llama-3.2-3B/ZsRE	MEMIT	1	8.30	5.74	5.36	3.77
Llama-3.2-3B/ZsRE	+key	1	5.53	8.13	3.87	5.08
Llama-3.2-3B/ZsRE	+value	1	3.71	3.34	6.45	6.11
Llama-3.2-3B/ZsRE	+both	1	2.40	4.81	4.12	8.67
Llama-3.2-3B/ZsRE	MEMIT	100	8.35	5.87	5.54	3.85
Llama-3.2-3B/ZsRE	+key	100	5.08	7.79	3.86	4.70
Llama-3.2-3B/ZsRE	+value	100	3.78	3.24	6.50	5.74
Llama-3.2-3B/ZsRE	+both	100	2.30	4.61	3.95	7.86

Table 19: GPT-2-XL and Llama-3.2-3B: deployed rewrite margin, damage against MEMIT's unquantized margin (paired 95% CI), the fraction of MEMIT's damage each editor returns (in sample for +value and +both, whose value is optimized through the quantized network; held-out: Table 21), deployed ES, PS and NS, and the paired ΔPS and ΔNS against MEMIT (bold = CI excludes zero). Same run as Table 18, which prints its int3 cells.

model	quant	n	editor	deployed	damage vs fp MEMIT	recovered	ES	PS	NS	ΔPS	ΔNS
GPT-2-XL	NF4	1	MEMIT	12.29	+0.55 [+0.43, +0.68]	—	0.99	0.85	0.82	—	—
GPT-2-XL	NF4	1	+key	12.65	+0.19 [+0.08, +0.31]	65%	0.99	0.88	0.82	**+0.025** [+0.005, +0.050]	+0.000 [+0.000, +0.000]
GPT-2-XL	NF4	1	+value	12.45	+0.39 [+0.19, +0.60]	29%	0.99	0.85	0.82	+0.000 [−0.035, +0.035]	+0.000 [+0.000, +0.000]
GPT-2-XL	NF4	1	+both	12.82	+0.02 [−0.17, +0.22]	97%	0.99	0.87	0.82	+0.015 [−0.020, +0.050]	+0.000 [+0.000, +0.000]
GPT-2-XL	NF4	100	MEMIT	12.62	+0.51 [+0.35, +0.67]	—	1.00	0.88	0.80	—	—
GPT-2-XL	NF4	100	+key	12.92	+0.21 [+0.06, +0.36]	59%	1.00	0.92	0.81	**+0.040** [+0.010, +0.080]	+0.010 [+0.000, +0.030]
GPT-2-XL	NF4	100	+value	12.85	+0.28 [−0.02, +0.57]	45%	1.00	0.89	0.80	+0.010 [+0.000, +0.030]	+0.000 [+0.000, +0.000]
GPT-2-XL	NF4	100	+both	13.14	−0.01 [−0.29, +0.27]	102%	1.00	0.91	0.80	+0.030 [+0.000, +0.070]	+0.000 [+0.000, +0.000]
GPT-2-XL	int3	1	MEMIT	6.36	+6.49 [+5.98, +7.02]	—	0.92	0.72	0.80	—	—
GPT-2-XL	int3	1	+key	9.76	+3.08 [+2.72, +3.46]	52%	0.98	0.88	0.78	**+0.150** [+0.100, +0.200]	−0.015 [−0.035, +0.000]
GPT-2-XL	int3	1	+value	7.96	+4.88 [+4.32, +5.46]	25%	0.95	0.72	0.80	+0.000 [−0.025, +0.030]	+0.000 [+0.000, +0.000]
GPT-2-XL	int3	1	+both	12.85	−0.01 [−0.34, +0.32]	100%	0.99	0.87	0.78	**+0.145** [+0.095, +0.195]	−0.015 [−0.035, +0.000]
GPT-2-XL	int3	100	MEMIT	6.69	+6.44 [+5.72, +7.19]	—	0.94	0.73	0.78	—	—
GPT-2-XL	int3	100	+key	9.75	+3.38 [+2.87, +3.91]	48%	1.00	0.88	0.77	**+0.150** [+0.080, +0.220]	−0.010 [−0.030, +0.000]
GPT-2-XL	int3	100	+value	8.24	+4.89 [+4.11, +5.71]	24%	0.95	0.73	0.78	+0.000 [−0.060, +0.050]	+0.000 [−0.030, +0.030]
GPT-2-XL	int3	100	+both	12.53	+0.60 [+0.13, +1.09]	91%	1.00	0.87	0.76	**+0.140** [+0.060, +0.220]	−0.020 [−0.050, +0.000]
GPT-2-XL	int3	1000	MEMIT	1.22	+6.04 [+5.46, +6.65]	—	0.64	0.48	0.62	—	—
GPT-2-XL	int3	1000	+key	3.11	+4.15 [+3.60, +4.72]	31%	0.79	0.57	0.61	**+0.085** [+0.035, +0.140]	−0.010 [−0.065, +0.045]
GPT-2-XL	int3	1000	+value	1.45	+5.81 [+5.16, +6.44]	4%	0.63	0.47	0.64	−0.020 [−0.070, +0.025]	+0.015 [−0.035, +0.065]
GPT-2-XL	int3	1000	+both	4.22	+3.04 [+2.44, +3.66]	50%	0.83	0.63	0.65	**+0.145** [+0.085, +0.205]	+0.025 [−0.025, +0.075]
GPT-2-XL	int4	1	MEMIT	11.71	+1.14 [+0.96, +1.33]	—	0.99	0.86	0.82	—	—
GPT-2-XL	int4	1	+key	12.29	+0.55 [+0.41, +0.71]	52%	0.99	0.88	0.82	+0.015 [+0.000, +0.035]	+0.000 [+0.000, +0.000]
GPT-2-XL	int4	1	+value	12.20	+0.65 [+0.40, +0.89]	43%	0.99	0.83	0.82	**−0.030** [−0.060, −0.005]	+0.000 [−0.015, +0.015]
GPT-2-XL	int4	1	+both	12.83	+0.01 [−0.22, +0.25]	99%	0.99	0.86	0.81	+0.005 [−0.020, +0.030]	−0.005 [−0.015, +0.000]
GPT-2-XL	int4	100	MEMIT	12.00	+1.13 [+0.91, +1.36]	—	1.00	0.89	0.80	—	—
GPT-2-XL	int4	100	+key	12.52	+0.60 [+0.42, +0.79]	46%	1.00	0.89	0.79	+0.000 [+0.000, +0.000]	−0.010 [−0.030, +0.000]
GPT-2-XL	int4	100	+value	12.36	+0.77 [+0.47, +1.07]	32%	1.00	0.87	0.82	−0.020 [−0.050, +0.000]	+0.020 [+0.000, +0.050]
GPT-2-XL	int4	100	+both	12.87	+0.25 [−0.03, +0.53]	77%	1.00	0.89	0.83	+0.000 [−0.030, +0.030]	+0.030 [+0.000, +0.070]
Llama-3.2-3B	NF4	1	MEMIT	12.70	+1.00 [+0.80, +1.20]	—	0.99	0.95	0.85	—	—
Llama-3.2-3B	NF4	1	+key	13.18	+0.51 [+0.33, +0.69]	49%	1.00	0.95	0.84	+0.005 [−0.010, +0.020]	−0.010 [−0.025, +0.000]
Llama-3.2-3B	NF4	1	+value	13.30	+0.39 [+0.05, +0.72]	61%	1.00	0.94	0.84	−0.010 [−0.040, +0.020]	−0.005 [−0.020, +0.010]
Llama-3.2-3B	NF4	1	+both	13.84	−0.14 [−0.48, +0.18]	115%	1.00	0.94	0.84	−0.005 [−0.040, +0.025]	−0.010 [−0.030, +0.010]
Llama-3.2-3B	NF4	100	MEMIT	12.44	+1.11 [+0.81, +1.43]	—	1.00	0.93	0.81	—	—
Llama-3.2-3B	NF4	100	+key	12.81	+0.74 [+0.46, +1.02]	34%	1.00	0.93	0.80	+0.000 [+0.000, +0.000]	−0.010 [−0.030, +0.000]
Llama-3.2-3B	NF4	100	+value	13.04	+0.51 [+0.05, +0.97]	54%	1.00	0.93	0.81	+0.000 [−0.050, +0.050]	+0.000 [+0.000, +0.000]
Llama-3.2-3B	NF4	100	+both	13.54	+0.01 [−0.42, +0.45]	99%	1.00	0.95	0.79	+0.020 [−0.030, +0.070]	−0.020 [−0.060, +0.020]
Llama-3.2-3B	int3	1	MEMIT	7.06	+6.63 [+6.14, +7.13]	—	0.97	0.88	0.81	—	—
Llama-3.2-3B	int3	1	+key	9.18	+4.52 [+4.03, +5.01]	32%	0.98	0.94	0.80	**+0.060** [+0.020, +0.100]	−0.020 [−0.045, +0.000]
Llama-3.2-3B	int3	1	+value	9.48	+4.21 [+3.69, +4.75]	37%	0.98	0.86	0.81	−0.015 [−0.075, +0.045]	−0.005 [−0.025, +0.010]
Llama-3.2-3B	int3	1	+both	13.20	+0.50 [+0.08, +0.90]	93%	1.00	0.95	0.79	**+0.075** [+0.030, +0.120]	**−0.030** [−0.055, −0.010]
Llama-3.2-3B	int3	100	MEMIT	6.89	+6.66 [+5.98, +7.35]	—	0.95	0.85	0.80	—	—
Llama-3.2-3B	int3	100	+key	8.95	+4.60 [+3.91, +5.31]	31%	0.96	0.92	0.75	**+0.070** [+0.020, +0.130]	−0.050 [−0.110, +0.010]
Llama-3.2-3B	int3	100	+value	9.61	+3.95 [+3.20, +4.72]	41%	0.99	0.88	0.82	+0.030 [−0.060, +0.120]	+0.020 [−0.040, +0.080]
Llama-3.2-3B	int3	100	+both	12.97	+0.58 [−0.00, +1.16]	91%	1.00	0.93	0.78	+0.080 [+0.000, +0.160]	−0.020 [−0.070, +0.030]
Llama-3.2-3B	int4	1	MEMIT	12.29	+1.40 [+1.15, +1.66]	—	1.00	0.94	0.83	—	—
Llama-3.2-3B	int4	1	+key	12.81	+0.89 [+0.65, +1.13]	37%	1.00	0.95	0.83	+0.010 [+0.000, +0.025]	−0.005 [−0.015, +0.000]
Llama-3.2-3B	int4	1	+value	13.24	+0.45 [+0.13, +0.77]	68%	1.00	0.95	0.84	+0.010 [−0.025, +0.050]	+0.005 [−0.010, +0.025]
Llama-3.2-3B	int4	1	+both	13.88	−0.19 [−0.50, +0.13]	113%	1.00	0.95	0.82	+0.015 [−0.020, +0.050]	−0.010 [−0.030, +0.010]
Llama-3.2-3B	int4	100	MEMIT	12.23	+1.32 [+0.97, +1.68]	—	1.00	0.95	0.81	—	—
Llama-3.2-3B	int4	100	+key	12.62	+0.93 [+0.60, +1.27]	29%	1.00	0.95	0.79	+0.000 [+0.000, +0.000]	−0.020 [−0.050, +0.000]
Llama-3.2-3B	int4	100	+value	13.19	+0.37 [−0.07, +0.80]	72%	1.00	0.96	0.81	+0.010 [−0.020, +0.050]	+0.000 [−0.040, +0.040]
Llama-3.2-3B	int4	100	+both	13.68	−0.13 [−0.56, +0.30]	110%	1.00	0.96	0.77	+0.010 [−0.020, +0.050]	−0.040 [−0.090, +0.000]

Table 20: GPT-J-6B, Llama-3.1-8B, Qwen2.5-3B and ZsRE: the same columns as Table 19.

model	quant	n	editor	deployed	damage vs fp MEMIT	recovered	ES	PS	NS	ΔPS	ΔNS
GPT-J-6B	int3	1	MEMIT	10.19	+4.72 [+4.30, +5.14]	—	0.98	0.93	0.80	—	—
GPT-J-6B	int3	1	+key	11.35	+3.56 [+3.15, +3.98]	25%	0.99	0.96	0.77	**+0.030** [+0.010, +0.055]	**−0.035** [−0.065, −0.010]
GPT-J-6B	int3	1	+value	12.73	+2.18 [+1.71, +2.65]	54%	0.99	0.94	0.81	+0.005 [−0.025, +0.035]	+0.010 [−0.010, +0.030]
GPT-J-6B	int3	1	+both	14.46	+0.45 [+0.00, +0.90]	90%	0.99	0.94	0.78	+0.015 [−0.015, +0.050]	−0.025 [−0.050, +0.000]
GPT-J-6B	int3	100	MEMIT	10.42	+4.60 [+4.06, +5.16]	—	1.00	0.92	0.77	—	—
GPT-J-6B	int3	100	+key	11.16	+3.87 [+3.35, +4.41]	16%	1.00	0.96	0.76	**+0.040** [+0.010, +0.080]	−0.010 [−0.050, +0.020]
GPT-J-6B	int3	100	+value	12.60	+2.42 [+1.83, +3.03]	47%	1.00	0.92	0.76	+0.000 [−0.050, +0.050]	−0.010 [−0.050, +0.030]
GPT-J-6B	int3	100	+both	14.00	+1.02 [+0.51, +1.56]	78%	1.00	0.95	0.79	+0.030 [−0.010, +0.080]	+0.020 [−0.020, +0.060]
Llama-3.1-8B	int3	1	MEMIT	6.65	+8.15 [+7.60, +8.70]	—	0.92	0.85	0.82	—	—
Llama-3.1-8B	int3	1	+key	9.47	+5.33 [+4.81, +5.87]	35%	0.98	0.95	0.81	**+0.100** [+0.055, +0.145]	−0.015 [−0.040, +0.005]
Llama-3.1-8B	int3	1	+value	8.76	+6.04 [+5.39, +6.69]	26%	0.97	0.83	0.82	−0.020 [−0.070, +0.030]	+0.000 [−0.020, +0.020]
Llama-3.1-8B	int3	1	+both	14.17	+0.63 [+0.09, +1.19]	92%	0.99	0.96	0.82	**+0.110** [+0.065, +0.160]	−0.005 [−0.035, +0.025]
Llama-3.1-8B	int3	100	MEMIT	6.88	+8.01 [+7.20, +8.82]	—	0.90	0.82	0.79	—	—
Llama-3.1-8B	int3	100	+key	9.61	+5.28 [+4.53, +6.06]	34%	0.97	0.93	0.73	**+0.110** [+0.050, +0.180]	**−0.060** [−0.120, −0.010]
Llama-3.1-8B	int3	100	+value	8.92	+5.96 [+5.07, +6.86]	26%	0.98	0.84	0.80	+0.020 [−0.040, +0.080]	+0.010 [−0.040, +0.060]
Llama-3.1-8B	int3	100	+both	13.64	+1.25 [+0.54, +1.95]	84%	1.00	1.00	0.76	**+0.180** [+0.110, +0.260]	−0.030 [−0.090, +0.030]
Qwen2.5-3B	int3	1	MEMIT	−0.22	+14.78 [+13.91, +15.67]	—	0.49	0.47	0.52	—	—
Qwen2.5-3B	int3	1	+key	0.53	+14.03 [+13.18, +14.89]	5%	0.58	0.59	0.41	**+0.120** [+0.070, +0.170]	**−0.105** [−0.155, −0.060]
Qwen2.5-3B	int3	1	+value	−0.16	+14.72 [+13.82, +15.61]	0%	0.49	0.47	0.52	−0.005 [−0.015, +0.000]	+0.000 [−0.015, +0.015]
Qwen2.5-3B	int3	1	+both	6.95	+7.61 [+6.69, +8.53]	49%	0.96	0.92	0.10	**+0.445** [+0.375, +0.515]	**−0.420** [−0.490, −0.350]
Qwen2.5-3B	int3	100	MEMIT	0.11	+15.29 [+14.03, +16.54]	—	0.57	0.53	0.48	—	—
Qwen2.5-3B	int3	100	+key	0.19	+15.22 [+13.92, +16.51]	0%	0.53	0.55	0.45	+0.020 [−0.050, +0.090]	−0.030 [−0.100, +0.040]
Qwen2.5-3B	int3	100	+value	0.14	+15.26 [+13.99, +16.53]	0%	0.53	0.54	0.50	+0.010 [−0.040, +0.060]	+0.020 [−0.020, +0.060]
Qwen2.5-3B	int3	100	+both	1.74	+13.67 [+12.46, +14.90]	11%	0.66	0.57	0.48	+0.040 [−0.040, +0.120]	+0.000 [−0.090, +0.090]
GPT-2-XL/ZsRE	int3	1	MEMIT	3.25	+5.16 [+4.76, +5.57]	—	0.89	0.81	0.75	—	—
GPT-2-XL/ZsRE	int3	1	+key	5.44	+2.97 [+2.63, +3.33]	42%	0.99	0.92	0.76	**+0.110** [+0.065, +0.155]	+0.005 [+0.000, +0.015]
GPT-2-XL/ZsRE	int3	1	+value	4.91	+3.50 [+3.08, +3.92]	32%	0.94	0.85	0.75	**+0.045** [+0.010, +0.080]	+0.000 [−0.015, +0.015]
GPT-2-XL/ZsRE	int3	1	+both	8.86	−0.45 [−0.69, −0.21]	109%	1.00	0.97	0.74	**+0.165** [+0.115, +0.220]	−0.005 [−0.015, +0.000]
GPT-2-XL/ZsRE	int3	100	MEMIT	3.45	+4.77 [+4.24, +5.33]	—	0.93	0.84	0.84	—	—
GPT-2-XL/ZsRE	int3	100	+key	5.18	+3.05 [+2.55, +3.57]	36%	0.99	0.95	0.84	**+0.110** [+0.050, +0.170]	+0.000 [+0.000, +0.000]
GPT-2-XL/ZsRE	int3	100	+value	5.20	+3.03 [+2.50, +3.58]	37%	0.97	0.88	0.81	+0.040 [−0.040, +0.120]	−0.030 [−0.070, +0.000]
GPT-2-XL/ZsRE	int3	100	+both	8.23	−0.01 [−0.35, +0.35]	100%	1.00	0.97	0.83	**+0.130** [+0.060, +0.210]	−0.010 [−0.030, +0.000]
Llama-3.2-3B/ZsRE	int3	1	MEMIT	3.77	+4.53 [+4.09, +4.97]	—	0.92	0.88	0.85	—	—
Llama-3.2-3B/ZsRE	int3	1	+key	5.08	+3.22 [+2.80, +3.64]	29%	0.95	0.94	0.85	**+0.065** [+0.030, +0.105]	+0.000 [−0.015, +0.015]
Llama-3.2-3B/ZsRE	int3	1	+value	6.11	+2.20 [+1.78, +2.62]	51%	0.99	0.94	0.86	**+0.060** [+0.020, +0.105]	+0.005 [+0.000, +0.015]
Llama-3.2-3B/ZsRE	int3	1	+both	8.67	−0.36 [−0.67, −0.05]	108%	1.00	0.99	0.85	**+0.110** [+0.070, +0.155]	+0.000 [−0.015, +0.015]
Llama-3.2-3B/ZsRE	int3	100	MEMIT	3.85	+4.50 [+3.83, +5.21]	—	0.92	0.83	0.83	—	—
Llama-3.2-3B/ZsRE	int3	100	+key	4.70	+3.65 [+3.03, +4.27]	19%	0.96	0.91	0.80	**+0.080** [+0.020, +0.140]	−0.030 [−0.080, +0.010]
Llama-3.2-3B/ZsRE	int3	100	+value	5.74	+2.61 [+1.96, +3.31]	42%	0.99	0.94	0.83	**+0.110** [+0.050, +0.180]	+0.000 [−0.040, +0.040]
Llama-3.2-3B/ZsRE	int3	100	+both	7.86	+0.49 [−0.00, +1.00]	89%	1.00	0.97	0.82	**+0.140** [+0.070, +0.210]	−0.010 [−0.060, +0.030]

Table 21: QAE on margins its objective does not see. v_q^* is optimized on the rewrite prompt; the two paraphrase prompts and the three neighbourhood prompts of each record are held out. Single edits at 3 bits: symmetric int3 (200 probes, the run of Table 2) on the five main models and ZsRE, the served GPTQ-3 and AWQ-3 of Appendix E, and a replication on facts no earlier single-edit run had scored (records 201–400; on GPT-2-XL and the Llamas they are batches 2 and 3 of Tables 12 and 13). loss: MEMIT's deployed margin loss (nats); rec.: share of it QAE returns (ratio of means, 95% CI with probes resampled jointly); Δ: QAE minus MEMIT deployed margin (nats, paired 95% CI; neighbourhood: negative is a locality cost). At 4 bits under the served GPTQ-4 and AWQ-4 (single edits, not tabulated) the paraphrase recovery interval includes zero in all eight cells.

model	quant	loss (rewrite)	rec. (rewrite)	loss (para.)	rec. (paraphrase)	Δ para.	Δ neighb.
GPT-2-XL	int3	6.49	100% [95, 105]	3.28	95% [85, 106]	+3.12 [+2.77, +3.47]	−0.09 [−0.14, −0.05]
Llama-3.2-3B	int3	6.63	93% [87, 99]	3.55	72% [61, 85]	+2.57 [+2.07, +3.09]	−0.24 [−0.39, −0.12]
GPT-J-6B	int3	4.72	90% [81, 100]	1.93	89% [67, 114]	+1.72 [+1.31, +2.14]	−0.10 [−0.17, −0.02]
Llama-3.1-8B	int3	8.15	92% [86, 99]	3.60	84% [72, 97]	+3.04 [+2.58, +3.50]	−0.19 [−0.31, −0.07]
Qwen2.5-3B	int3	14.78	49% [44, 53]	6.37	87% [76, 100]	+5.54 [+5.06, +6.03]	−4.59 [−4.98, −4.20]
GPT-2-XL/ZsRE	int3	5.16	109% [104, 114]	3.68	109% [100, 119]	+4.01 [+3.67, +4.37]	−0.02 [−0.03, −0.01]
Llama-3.2-3B/ZsRE	int3	4.53	108% [101, 116]	3.00	110% [99, 122]	+3.29 [+2.96, +3.62]	−0.00 [−0.06, +0.04]
served quantizers, zero-point groups of 128, 100 probes							
GPT-2-XL	GPTQ-3	2.68	104% [90, 119]	1.42	66% [41, 94]	+0.94 [+0.56, +1.32]	−0.04 [−0.09, +0.00]
	AWQ-3	2.98	93% [80, 106]	1.89	69% [49, 90]	+1.29 [+0.92, +1.67]	−0.04 [−0.07, −0.01]
Llama-3.2-3B	GPTQ-3	5.02	99% [89, 109]	1.77	99% [68, 136]	+1.76 [+1.16, +2.38]	−0.06 [−0.20, +0.05]
	AWQ-3	3.83	89% [78, 100]	1.40	94% [58, 137]	+1.32 [+0.79, +1.88]	−0.14 [−0.29, −0.01]
Llama-3.1-8B	GPTQ-3	6.33	98% [88, 109]	2.36	82% [58, 111]	+1.93 [+1.36, +2.49]	+0.15 [+0.02, +0.30]
	AWQ-3	5.16	90% [78, 102]	1.47	124% [83, 178]	+1.82 [+1.12, +2.54]	+0.01 [−0.36, +0.31]
Qwen2.5-3B	GPTQ-3	8.73	77% [65, 88]	3.28	36% [14, 59]	+1.18 [+0.44, +1.93]	−0.02 [−0.12, +0.08]
	AWQ-3	6.73	86% [72, 100]	2.29	47% [10, 88]	+1.07 [+0.22, +1.91]	+0.00 [−0.19, +0.25]
fresh facts: CounterFact records 201–400, int3, not scored before as single edits (200 probes; 100 on Qwen2.5-3B)							
GPT-2-XL	int3	6.07	103% [99, 108]	3.28	101% [90, 114]	+3.32 [+2.94, +3.73]	−0.06 [−0.09, −0.04]
Llama-3.2-3B	int3	6.75	98% [92, 104]	3.03	83% [70, 98]	+2.52 [+2.05, +3.00]	−0.27 [−0.38, −0.18]
Llama-3.1-8B	int3	8.08	97% [91, 104]	3.72	84% [72, 97]	+3.13 [+2.62, +3.63]	−0.19 [−0.29, −0.09]
Qwen2.5-3B	int3	13.16	53% [47, 60]	5.73	97% [80, 118]	+5.53 [+4.90, +6.18]	−4.94 [−5.46, −4.43]

Table 22: QAE's "recovered" (in sample; held-out: Table 21) with its interval (ratio of means, probes resampled jointly) and, because values above 100% mean QAE's deployed margin exceeds MEMIT's unquantized one, the deployed QAE margin minus MEMIT's fp margin in nats (paired 95% CI). D is MEMIT's deployed damage.

model	quant	n	D	recovered	QAE deployed − MEMIT fp
GPT-2-XL	NF4	1	0.55	97% [60, 133]	-0.02 [$-0.22, +0.18$]
GPT-2-XL	NF4	100	0.51	102% [48, 165]	$+0.01$ [$-0.26, +0.29$]
GPT-2-XL	int3	1	6.49	100% [95, 105]	$+0.01$ [$-0.32, +0.34$]
GPT-2-XL	int3	100	6.44	91% [84, 98]	-0.60 [$-1.09, -0.13$]
GPT-2-XL	int3	1000	6.04	50% [42, 57]	-3.04 [$-3.67, -2.42$]
GPT-2-XL	int4	1	1.14	99% [78, 119]	-0.01 [$-0.24, +0.22$]
GPT-2-XL	int4	100	1.13	77% [52, 102]	-0.25 [$-0.54, +0.03$]
Llama-3.2-3B	NF4	1	1.00	115% [82, 149]	$+0.14$ [$-0.18, +0.48$]
Llama-3.2-3B	NF4	100	1.11	99% [60, 140]	-0.01 [$-0.45, +0.43$]
Llama-3.2-3B	int3	1	6.63	93% [87, 99]	-0.50 [$-0.90, -0.09$]
Llama-3.2-3B	int3	100	6.66	91% [83, 100]	-0.58 [$-1.17, -0.00$]
Llama-3.2-3B	int4	1	1.40	113% [91, 139]	$+0.19$ [$-0.13, +0.50$]
Llama-3.2-3B	int4	100	1.32	110% [78, 147]	$+0.13$ [$-0.31, +0.56$]
GPT-J-6B	int3	1	4.72	90% [81, 100]	-0.45 [$-0.90, -0.01$]
GPT-J-6B	int3	100	4.60	78% [67, 89]	-1.02 [$-1.56, -0.51$]
Llama-3.1-8B	int3	1	8.15	92% [86, 99]	-0.63 [$-1.19, -0.07$]
Llama-3.1-8B	int3	100	8.01	84% [76, 93]	-1.25 [$-1.95, -0.55$]
Qwen2.5-3B	int3	1	14.78	49% [44, 53]	-7.61 [$-8.53, -6.69$]
Qwen2.5-3B	int3	100	15.29	11% [8, 13]	-13.67 [$-14.87, -12.46$]
GPT-2-XL/ZsRE	int3	1	5.16	109% [104, 114]	$+0.45$ [$+0.20, +0.70$]
GPT-2-XL/ZsRE	int3	100	4.77	100% [93, 108]	$+0.01$ [$-0.35, +0.36$]
Llama-3.2-3B/ZsRE	int3	1	4.53	108% [101, 116]	$+0.36$ [$+0.06, +0.67$]
Llama-3.2-3B/ZsRE	int3	100	4.50	89% [80, 100]	-0.49 [$-1.01, +0.01$]
GPT-2-XL, unlearning	int3	1	10.33	103% [99, 108]	$+0.35$ [$-0.13, +0.83$]
GPT-2-XL, unlearning	int3	100	10.00	93% [86, 100]	-0.72 [$-1.46, +0.01$]
Llama-3.2-3B, unlearning	int3	1	12.91	95% [91, 100]	-0.59 [$-1.25, +0.04$]
Llama-3.2-3B, unlearning	int3	100	12.27	78% [71, 84]	-2.76 [$-3.61, -1.94$]

Table 23: MEMIT's efficacy and paraphrase success unquantized (fp) and deployed, on the probes of the QAE runs.

model	quant	n	ES fp	ES deployed	PS fp	PS deployed
GPT-2-XL	NF4	1	0.995	0.995	0.885	0.855
GPT-2-XL	int3	1	0.995	0.920	0.885	0.725
GPT-2-XL	int4	1	0.995	0.995	0.885	0.860
GPT-2-XL	NF4	100	1.000	1.000	0.910	0.880
GPT-2-XL	int3	100	1.000	0.940	0.910	0.730
GPT-2-XL	int4	100	1.000	1.000	0.910	0.890
GPT-2-XL	int3	1000	0.930	0.635	0.685	0.485
Llama-3.2-3B	NF4	1	1.000	0.995	0.950	0.950
Llama-3.2-3B	int3	1	1.000	0.970	0.950	0.875
Llama-3.2-3B	int4	1	1.000	1.000	0.950	0.940
Llama-3.2-3B	NF4	100	1.000	1.000	0.950	0.930
Llama-3.2-3B	int3	100	1.000	0.950	0.950	0.850
Llama-3.2-3B	int4	100	1.000	1.000	0.950	0.950
GPT-J-6B	int3	1	1.000	0.985	0.955	0.930
GPT-J-6B	int3	100	1.000	1.000	0.950	0.920
Llama-3.1-8B	int3	1	1.000	0.915	0.960	0.850
Llama-3.1-8B	int3	100	1.000	0.900	0.950	0.820
Qwen2.5-3B	int3	1	0.995	0.490	0.900	0.475
Qwen2.5-3B	int3	100	1.000	0.570	0.920	0.530

G SPLITTING THE CONDITIONING: KEY AND READ-OUT

The keys-only channel of §4 quantizes every backbone matrix except the edited one, which conflates two effects: the matrices that produce k^* move it to k_q^*, and the matrices after the edited layer degrade the path from the edit's output to the logits. `code/channel_split.py` separates them on GPT-2-XL and Llama-3.2-3B. *Upstream* quantizes the matrices that feed the key (all blocks below the edited layer, and that layer's attention and MLP input projections: 71 matrices on GPT-2-XL, 34 on Llama-3.2-3B; the key read this way is identical to the key read under the full quantized backbone, maximum relative difference 0), with every downstream matrix and the edited matrix in full precision; *downstream* quantizes the matrices after the edited layer (120 and 161) with k^* intact. The same 200 probes (100 at $n = 100$) are evaluated in both, for MEMIT, KAE and the unedited model; the split runs solve every editor afresh (Appendix A, independent solves), so their MEMIT fp and keys-only margins differ from Table 11's by up to 0.03 nats. Tables 24 and 25 give the rewrite margins and the loss of each channel relative to the same editor's fp margin. The split is model-dependent, and since the models also differ in edit depth, MLP type (GELU or gated), instruction tuning and outlier structure (Appendix U), this describes the dependence without explaining it. On GPT-2-XL quantizing only the layers below the edit accounts for 76%, 76% and 89% of MEMIT's keys-only loss at int3 ($n = 1, 100, 1000$) and 62–79% at int4, quantizing only those above it for 22–41% (int3 and int4). The upstream channel moves the residual stream at the prediction position too, not only the key: the unedited model's upstream margin moves as well, and KAE, which fixes only the key, leaves 16% and 27% of MEMIT's upstream loss on GPT-2-XL and GPT-J-6B at int3, $n = 1$ (Tables 24 and 26; on the two Llamas it returns all of it, 104% and 107%, Tables 25 and 26). On Llama-3.2-3B the read-out half is the larger at int3 (53%, 53%, 56% against 70%, 67%, 63%) and more so at int4 for $n \leq 100$ (32–34% against 69–72%; 51% against 56% at $n = 1000$); the two halves' losses sum to more than the keys-only loss on both models. The unedited model's own margin on the same prompts moves by $+0.53$ $[+0.37, +0.69]$ and $+1.39$ $[+1.11, +1.67]$ nats at int3, towards the new object and well below the edit's loss. KAE, solved at k_q^*, recovers most of the upstream half at $n \leq 100$ on both models and about half to four fifths at $n = 1000$: 84% of MEMIT's upstream loss on GPT-2-XL at int3, $n = 1$ ($+4.19$ $[+3.75, +4.64]$ of 4.98 nats; 80% at int4, 78% at $n = 100$, 80% at $n = 1000$) and 104% on Llama-3.2-3B ($+3.68$ $[+3.30, +4.09]$ of 3.54; 123% at int4, 95% at $n = 100$, 55% at $n = 1000$), and under the upstream channel it delivers more than its own fp margin (12.05 against 9.91; 13.87 against 9.54), because the key it was solved for is the key it is read at. Of the whole keys-only loss it recovers 53% on GPT-2-XL ($+3.44$ $[+3.07, +3.84]$ of 6.52) and 33% on Llama-3.2-3B ($+2.24$ $[+1.90, +2.61]$ of 6.73): what it does not recover is the read-out half, which a re-solve at the key cannot reach and which a solve of v^* against the quantized read-out might. Table 26 runs the same split on the remaining three main models (upstream/downstream matrices 31/136 on GPT-J-6B, whose parallel block puts the edited layer's attention downstream, 41/182 on Llama-3.1-8B, 41/210 on Qwen2.5-3B; 200 probes at $n = 1$, 100 at $n = 100$). Over all five models and every int3 batch size run ($n = 1000$ only on GPT-2-XL and Llama-3.2-3B) the upstream share is 43–102% and the downstream share 29–70%, so which half is the larger depends on the model and the bit width: at int3 the upstream half dominates on GPT-2-XL, the two are comparable on Llama-3.1-8B (71% and 68%), the downstream half is the larger on Llama-3.2-3B and GPT-J-6B, and at int4 it is the larger on every model but GPT-2-XL (29% against 71% on Llama-3.1-8B at $n = 1$); on Qwen2.5-3B the upstream channel alone accounts for the whole loss, but Qwen is the model int3 damages outright (its *unedited* margin moves $+4.93$ $[+4.24, +5.63]$ nats), so its shares are the least interpretable of the five. KAE returns 73%, 104%, 107% and 84% of the upstream loss on GPT-J-6B, Llama-3.2-3B, Llama-3.1-8B and GPT-2-XL at int3, $n = 1$, and 7% on Qwen2.5-3B, where int3 moves the key by $1.68\times$ its own norm; of the whole keys-only loss it returns 25–53% on the four and 5% on Qwen (a different denominator from the 25–52% of MEMIT's deployed damage that the key-aware half recovers in §4.5).

Table 24: GPT-2-XL: the fp rewrite margin in nats (mean over probes) and, under the upstream-only, downstream-only and full keys-only channels, the change Δm of that margin relative to the same editor's own fp margin, with its paired 95% CI (negative = margin lost); shares are the channel's loss divided by the keys-only loss, for MEMIT. The unedited rows at $n = 1$ and $n = 1000$ share the same 200 probes and are identical by construction.

editor	quant	n	fp margin	upstream Δm	downstream Δm	keys-only Δm	up. share	down. share
unedited	int3	1	-2.89	$+0.18$ $[+0.07, +0.31]$	$+0.33$ $[+0.24, +0.42]$	$+0.53$ $[+0.37, +0.69]$	—	—
MEMIT	int3	1	$+12.84$	-4.98 $[-5.50, -4.48]$	-2.55 $[-2.86, -2.26]$	-6.52 $[-7.05, -5.99]$	76%	39%
KAE	int3	1	$+9.91$	$+2.14$ $[+1.73, +2.55]$	-1.86 $[-2.09, -1.64]$	-0.15 $[-0.56, +0.26]$	—	—
unedited	int3	100	-2.80	$+0.21$ $[+0.05, +0.39]$	$+0.41$ $[+0.27, +0.55]$	$+0.62$ $[+0.39, +0.87]$	—	—
MEMIT	int3	100	$+13.12$	-4.84 $[-5.56, -4.15]$	-2.43 $[-2.82, -2.04]$	-6.37 $[-7.12, -5.64]$	76%	38%
KAE	int3	100	$+9.50$	$+2.53$ $[+2.02, +3.07]$	-1.70 $[-2.04, -1.38]$	$+0.34$ $[-0.21, +0.91]$	—	—
unedited	int3	1000	-2.89	$+0.18$ $[+0.07, +0.31]$	$+0.33$ $[+0.24, +0.42]$	$+0.53$ $[+0.37, +0.69]$	—	—
MEMIT	int3	1000	$+7.26$	-4.51 $[-5.10, -3.93]$	-1.45 $[-1.70, -1.21]$	-5.05 $[-5.65, -4.45]$	89%	29%
KAE	int3	1000	$+3.26$	$+3.10$ $[+2.57, +3.65]$	-0.64 $[-0.82, -0.46]$	$+1.84$ $[+1.35, +2.33]$	—	—
unedited	int4	1	-2.89	$+0.04$ $[+0.01, +0.08]$	$+0.04$ $[+0.01, +0.07]$	$+0.09$ $[+0.04, +0.14]$	—	—
MEMIT	int4	1	$+12.84$	-0.77 $[-0.92, -0.63]$	-0.41 $[-0.52, -0.31]$	-1.13 $[-1.33, -0.95]$	68%	36%
KAE	int4	1	$+12.33$	$+0.36$ $[+0.24, +0.49]$	-0.38 $[-0.49, -0.28]$	-0.03 $[-0.20, +0.13]$	—	—
unedited	int4	100	-2.80	$+0.06$ $[+0.01, +0.12]$	$+0.04$ $[-0.01, +0.09]$	$+0.11$ $[+0.04, +0.19]$	—	—
MEMIT	int4	100	$+13.12$	-0.68 $[-0.85, -0.53]$	-0.45 $[-0.60, -0.30]$	-1.11 $[-1.34, -0.89]$	62%	41%
KAE	int4	100	$+12.46$	$+0.50$ $[+0.34, +0.67]$	-0.39 $[-0.54, -0.25]$	$+0.07$ $[-0.14, +0.27]$	—	—
unedited	int4	1000	-2.89	$+0.04$ $[+0.01, +0.08]$	$+0.04$ $[+0.01, +0.07]$	$+0.09$ $[+0.04, +0.14]$	—	—
MEMIT	int4	1000	$+7.26$	-1.00 $[-1.30, -0.72]$	-0.28 $[-0.38, -0.17]$	-1.27 $[-1.59, -0.95]$	79%	22%
KAE	int4	1000	$+6.13$	$+1.16$ $[+0.86, +1.47]$	-0.26 $[-0.36, -0.16]$	$+0.89$ $[+0.58, +1.21]$	—	—

Table 25: Llama-3.2-3B: the same split (34 upstream and 161 downstream matrices), same columns and conventions as Table 24.

editor	quant	n	fp margin	upstream Δm	downstream Δm	keys-only Δm	up. share	down. share
unedited	int3	1	-4.57	$+0.86$ $[+0.64, +1.09]$	$+0.69$ $[+0.48, +0.89]$	$+1.39$ $[+1.11, +1.67]$	—	—
MEMIT	int3	1	$+13.72$	-3.54 $[-3.96, -3.12]$	-4.69 $[-5.11, -4.28]$	-6.73 $[-7.24, -6.22]$	53%	70%
KAE	int3	1	$+9.54$	$+4.33$ $[+3.86, +4.83]$	-2.84 $[-3.22, -2.47]$	-0.30 $[-0.84, +0.26]$	—	—
unedited	int3	100	-4.37	$+0.84$ $[+0.53, +1.18]$	$+0.57$ $[+0.28, +0.87]$	$+1.32$ $[+0.91, +1.73]$	—	—
MEMIT	int3	100	$+13.75$	-3.70 $[-4.34, -3.07]$	-4.62 $[-5.13, -4.12]$	-6.93 $[-7.60, -6.26]$	53%	67%
KAE	int3	100	$+9.04$	$+4.53$ $[+3.84, +5.27]$	-2.71 $[-3.18, -2.24]$	-0.09 $[-0.90, +0.79]$	—	—
unedited	int3	1000	-4.57	$+0.86$ $[+0.64, +1.09]$	$+0.69$ $[+0.48, +0.89]$	$+1.39$ $[+1.11, +1.67]$	—	—
MEMIT	int3	1000	$+11.51$	-3.67 $[-4.19, -3.16]$	-4.16 $[-4.57, -3.75]$	-6.56 $[-7.17, -5.96]$	56%	63%
KAE	int3	1000	$+6.69$	$+3.19$ $[+2.57, +3.81]$	-2.23 $[-2.58, -1.88]$	-0.59 $[-1.23, +0.08]$	—	—
unedited	int4	1	-4.57	$+0.21$ $[+0.10, +0.32]$	$+0.10$ $[+0.02, +0.17]$	$+0.34$ $[+0.20, +0.49]$	—	—
MEMIT	int4	1	$+13.72$	-0.53 $[-0.69, -0.38]$	-1.08 $[-1.28, -0.88]$	-1.56 $[-1.81, -1.31]$	34%	69%
KAE	int4	1	$+13.26$	$+0.58$ $[+0.42, +0.76]$	-1.02 $[-1.22, -0.82]$	-0.56 $[-0.81, -0.31]$	—	—
unedited	int4	100	-4.37	$+0.15$ $[+0.00, +0.31]$	$+0.12$ $[+0.02, +0.22]$	$+0.28$ $[+0.11, +0.47]$	—	—
MEMIT	int4	100	$+13.75$	-0.47 $[-0.69, -0.26]$	-1.07 $[-1.34, -0.80]$	-1.48 $[-1.84, -1.13]$	32%	72%
KAE	int4	100	$+12.87$	$+0.85$ $[+0.62, +1.09]$	-0.99 $[-1.25, -0.74]$	-0.23 $[-0.60, +0.13]$	—	—
unedited	int4	1000	-4.57	$+0.21$ $[+0.10, +0.32]$	$+0.10$ $[+0.02, +0.17]$	$+0.34$ $[+0.20, +0.49]$	—	—
MEMIT	int4	1000	$+11.51$	-0.86 $[-1.10, -0.64]$	-0.95 $[-1.16, -0.74]$	-1.69 $[-2.01, -1.38]$	51%	56%
KAE	int4	1000	$+10.10$	$+0.92$ $[+0.64, +1.21]$	-0.72 $[-0.92, -0.52]$	$+0.02$ $[-0.32, +0.37]$	—	—

Table 26: The same split on the remaining three main models: the fp rewrite margin in nats and, under the upstream-only, downstream-only and full keys-only channels, the change Δm of that margin relative to the same editor's own fp margin, with its paired 95% CI (negative = margin lost). Shares are the channel's loss divided by the keys-only loss, for MEMIT.

editor	quant	n	fp margin	upstream Δm	downstream Δm	keys-only Δm	up. share	down. share
GPT-J-6B								
unedited	int3	1	-4.15	$+0.36$ $[+0.22, +0.51]$	$+0.23$ $[+0.10, +0.37]$	$+0.64$ $[+0.43, +0.87]$	—	—
MEMIT	int3	1	$+14.91$	-2.02 $[-2.27, -1.78]$	-2.62 $[-2.95, -2.28]$	-4.69 $[-5.11, -4.27]$	43%	56%
KAE	int3	1	$+12.96$	$+1.41$ $[+1.08, +1.74]$	-2.24 $[-2.54, -1.95]$	-1.57 $[-1.99, -1.15]$	—	—
unedited	int3	100	-4.08	$+0.39$ $[+0.21, +0.58]$	$+0.29$ $[+0.12, +0.47]$	$+0.71$ $[+0.44, +0.98]$	—	—
MEMIT	int3	100	$+15.02$	-2.11 $[-2.49, -1.75]$	-2.48 $[-2.94, -2.04]$	-4.54 $[-5.08, -4.00]$	46%	55%
KAE	int3	100	$+13.11$	$+0.87$ $[+0.45, +1.30]$	-2.20 $[-2.63, -1.78]$	-1.94 $[-2.47, -1.40]$	—	—
unedited	int4	1	-4.15	$+0.07$ $[+0.01, +0.14]$	$+0.03$ $[-0.02, +0.08]$	$+0.10$ $[+0.01, +0.19]$	—	—
MEMIT	int4	1	$+14.91$	-0.33 $[-0.42, -0.25]$	-0.36 $[-0.48, -0.24]$	-0.65 $[-0.82, -0.50]$	51%	55%
KAE	int4	1	$+14.60$	$+0.12$ $[+0.03, +0.22]$	-0.35 $[-0.47, -0.23]$	-0.21 $[-0.38, -0.05]$	—	—
unedited	int4	100	-4.08	$+0.13$ $[+0.04, +0.24]$	$+0.04$ $[-0.03, +0.11]$	$+0.17$ $[+0.04, +0.30]$	—	—
MEMIT	int4	100	$+15.02$	-0.32 $[-0.42, -0.22]$	-0.39 $[-0.55, -0.22]$	-0.66 $[-0.86, -0.46]$	48%	59%
KAE	int4	100	$+14.78$	$+0.11$ $[-0.01, +0.23]$	-0.38 $[-0.54, -0.22]$	-0.22 $[-0.42, -0.03]$	—	—
Llama-3.1-8B								
unedited	int3	1	-5.90	$+1.36$ $[+1.03, +1.72]$	$+0.83$ $[+0.63, +1.03]$	$+2.06$ $[+1.70, +2.44]$	—	—
MEMIT	int3	1	$+14.80$	-5.80 $[-6.41, -5.20]$	-5.52 $[-5.98, -5.07]$	-8.14 $[-8.69, -7.60]$	71%	68%
KAE	int3	1	$+7.38$	$+7.85$ $[+7.18, +8.53]$	-1.57 $[-1.95, -1.21]$	$+2.12$ $[+1.37, +2.89]$	—	—
unedited	int3	100	-5.92	$+1.75$ $[+1.21, +2.34]$	$+0.94$ $[+0.69, +1.20]$	$+2.48$ $[+1.92, +3.07]$	—	—
MEMIT	int3	100	$+14.89$	-5.93 $[-6.77, -5.09]$	-5.37 $[-6.00, -4.77]$	-8.03 $[-8.86, -7.25]$	74%	67%
KAE	int3	100	$+6.84$	$+8.10$ $[+7.19, +9.04]$	-1.38 $[-1.88, -0.90]$	$+2.81$ $[+1.76, +3.88]$	—	—
unedited	int4	1	-5.90	$+0.10$ $[-0.04, +0.24]$	$+0.17$ $[+0.08, +0.26]$	$+0.19$ $[+0.02, +0.36]$	—	—
MEMIT	int4	1	$+14.80$	-0.54 $[-0.77, -0.32]$	-1.34 $[-1.56, -1.11]$	-1.88 $[-2.19, -1.57]$	29%	71%
KAE	int4	1	$+13.77$	$+1.25$ $[+1.02, +1.48]$	-1.12 $[-1.34, -0.90]$	-0.22 $[-0.53, +0.10]$	—	—
unedited	int4	100	-5.92	$+0.18$ $[-0.03, +0.43]$	$+0.25$ $[+0.11, +0.40]$	$+0.35$ $[+0.11, +0.64]$	—	—
MEMIT	int4	100	$+14.89$	-0.56 $[-0.88, -0.24]$	-1.17 $[-1.49, -0.85]$	-1.87 $[-2.27, -1.46]$	30%	62%
KAE	int4	100	$+13.41$	$+1.65$ $[+1.35, +1.95]$	-0.86 $[-1.16, -0.56]$	$+0.32$ $[-0.10, +0.74]$	—	—
Qwen2.5-3B								
unedited	int3	1	-5.18	$+4.87$ $[+4.19, +5.55]$	$+1.90$ $[+1.51, +2.31]$	$+4.93$ $[+4.24, +5.63]$	—	—
MEMIT	int3	1	$+14.56$	-14.83 $[-15.68, -13.98]$	-9.50 $[-10.13, -8.85]$	-14.80 $[-15.67, -13.93]$	100%	64%
KAE	int3	1	-5.12	$+5.91$ $[+5.19, +6.65]$	$+1.86$ $[+1.46, +2.26]$	$+5.66$ $[+4.93, +6.41]$	—	—
unedited	int3	100	-5.17	$+4.98$ $[+3.96, +5.99]$	$+1.93$ $[+1.37, +2.49]$	$+5.27$ $[+4.22, +6.32]$	—	—
MEMIT	int3	100	$+15.40$	-15.60 $[-16.82, -14.35]$	-9.45 $[-10.34, -8.57]$	-15.28 $[-16.53, -14.02]$	102%	62%
KAE	int3	100	-2.46	$+2.47$ $[+1.62, +3.33]$	$+0.71$ $[+0.26, +1.16]$	$+2.82$ $[+1.95, +3.72]$	—	—
unedited	int4	1	-5.18	$+0.29$ $[+0.11, +0.47]$	$+0.46$ $[+0.28, +0.65]$	$+0.55$ $[+0.32, +0.79]$	—	—
MEMIT	int4	1	$+14.56$	-2.39 $[-3.09, -1.74]$	-2.88 $[-3.32, -2.45]$	-4.22 $[-4.96, -3.51]$	57%	68%
KAE	int4	1	$+12.02$	$+1.31$ $[+0.71, +1.92]$	-2.08 $[-2.47, -1.70]$	-0.93 $[-1.57, -0.29]$	—	—
unedited	int4	100	-5.17	$+0.25$ $[-0.00, +0.52]$	$+0.48$ $[+0.21, +0.78]$	$+0.56$ $[+0.20, +0.91]$	—	—
MEMIT	int4	100	$+15.40$	-2.88 $[-3.93, -1.90]$	-2.90 $[-3.60, -2.18]$	-4.34 $[-5.44, -3.29]$	66%	67%
KAE	int4	100	$+12.46$	$+0.96$ $[+0.09, +1.83]$	-2.31 $[-2.96, -1.70]$	-0.73 $[-1.71, +0.22]$	—	—

H MARGIN-MATCHED CONTROL AND THE EDIT'S OWN EFFECT

A quantized backbone compresses confident predictions in general, and compression is known to degrade what a model knows (Namburi et al., 2023; Jaiswal et al., 2024; Wang et al., 2026), so part of what the keys-only channel removes from an edit could be removed from any fact. Two controls separate the two. Both use the unedited rows of the split runs (Appendix G; symmetric grid only, not the served quantizers), which carry, for the same 200 prompts, the unedited model's margin of the true object over the counterfactual one in every channel.

Matched on confidence. Table 27 restricts both the MEMIT edits ($n = 1$) and these natural facts to an fp margin of 6–12 nats (two different subsets of the 200 prompts, so the comparison is unpaired; the window takes the weakest edits and the most confident natural facts), where both populations are represented (59–79 edits and 29–83 facts per model on the four main models). At int3 in the keys-only channel an edit loses 3.57–5.97 nats against 1.27–2.86 for a natural fact of the same confidence (1.8–3.8×), and all twelve upstream, downstream and keys-only differences on the four models have intervals above zero. The generic part, what int3 does to any fact that confident, is 26–55% of an edit's keys-only loss, 15–52% of its upstream loss and 28–41% of its downstream loss; the rest is specific to the edit. The window matches the two populations in range, not in mean: inside it the edits' mean fp margin is 9.07–10.06 nats and the facts' 7.87–8.49 on the four main models, and the loss grows with the margin, so these shares understate the generic part; a linear adjustment for fp margin inside the window (one slope shared by edits and facts) puts it at 33–69% of the keys-only loss (33% [20, 46] on GPT-2-XL, 69% [57, 84] on Llama-3.2-3B, 41% [28, 56] on GPT-J-6B and 53% [39, 69] on Llama-3.1-8B, two-sample bootstrap; 91% on Qwen2.5-3B), and the edit/fact ratio at 1.4–3.1×. At int4 the keys-only ratio is 2.0–6.3×, with 5 of the 12 differences covering zero. Qwen2.5-3B is the exception: int3 costs its own facts 8.04 nats in the same window against 9.75 for its edits (1.2×), the signature of a backbone that int3 damages outright.

The edit's own effect. Table 28 measures the edit's effect on each prompt as the edited minus the unedited margin in the same channel. Because quantization moves the unedited margin towards the new object (Appendix G), the effect loses more than the raw margin does: 5.33–10.20 nats at int3 on the four models against raw losses of 4.69–8.14, with the upstream half removing 45–73% of it and the downstream half 41–66%, the same model dependence as the raw split. The conditioning account therefore survives both controls; what the controls change is how much of each half is specific to edits. Appendix O makes the related comparison against the edited fact's own neighbours, which start at a few nats and are therefore not matched on confidence.

Table 27: Margin-matched control. Edits (MEMIT, $n = 1$) and facts the unedited model already knows (its margin of the true object over the counterfactual one, on the same prompts) whose fp margin lies in 6–12 nats. Loss is the fp margin minus the channel margin (nats); the difference is edits minus natural facts with a two-sample bootstrap 95% CI, and the ratio is edit loss over fact loss. At int3 keys-only the raw fact/edit share is 26–55% (82% on Qwen2.5-3B); adjusted for the edits' higher mean fp margin inside the window it is 33–69% (91%), the figure the main text quotes (text of this appendix).

model	quant	channel	edits	edit loss	facts	fact loss	difference	ratio
GPT-2-XL	int3	upstream	79	4.39	29	0.65	+3.74 [+2.91, +4.55]	6.8×
GPT-2-XL	int3	downstream	79	1.92	29	0.68	+1.24 [+0.79, +1.69]	2.8×
GPT-2-XL	int3	keys-only	79	5.38	29	1.42	+3.96 [+3.10, +4.80]	3.8×
GPT-2-XL	int4	upstream	79	0.83	29	0.20	+0.63 [+0.38, +0.89]	4.1×
GPT-2-XL	int4	downstream	79	0.36	29	0.09	+0.27 [+0.09, +0.46]	4.0×
GPT-2-XL	int4	keys-only	79	1.18	29	0.31	+0.87 [+0.58, +1.19]	3.8×
Llama-3.2-3B	int3	upstream	76	3.26	69	1.70	+1.56 [+0.85, +2.26]	1.9×
Llama-3.2-3B	int3	downstream	76	3.56	69	1.46	+2.10 [+1.46, +2.74]	2.4×
Llama-3.2-3B	int3	keys-only	76	5.20	69	2.86	+2.34 [+1.54, +3.16]	1.8×
Llama-3.2-3B	int4	upstream	76	0.71	69	0.47	+0.24 [−0.08, +0.55]	1.5×
Llama-3.2-3B	int4	downstream	76	0.84	69	0.17	+0.67 [+0.34, +1.01]	4.9×
Llama-3.2-3B	int4	keys-only	76	1.43	69	0.70	+0.73 [+0.27, +1.19]	2.0×
GPT-J-6B	int3	upstream	61	1.53	56	0.67	+0.86 [+0.45, +1.28]	2.3×
GPT-J-6B	int3	downstream	61	1.95	56	0.54	+1.41 [+0.83, +2.02]	3.6×
GPT-J-6B	int3	keys-only	61	3.57	56	1.27	+2.31 [+1.60, +3.02]	2.8×
GPT-J-6B	int4	upstream	61	0.31	56	0.18	+0.13 [−0.06, +0.33]	1.7×
GPT-J-6B	int4	downstream	61	0.25	56	0.08	+0.16 [−0.06, +0.40]	2.9×
GPT-J-6B	int4	keys-only	61	0.57	56	0.27	+0.30 [−0.05, +0.66]	2.1×
Llama-3.1-8B	int3	upstream	59	4.89	83	1.82	+3.07 [+2.02, +4.10]	2.7×
Llama-3.1-8B	int3	downstream	59	3.90	83	1.15	+2.76 [+2.07, +3.45]	3.4×
Llama-3.1-8B	int3	keys-only	59	5.97	83	2.71	+3.26 [+2.33, +4.19]	2.2×
Llama-3.1-8B	int4	upstream	59	0.55	83	0.22	+0.33 [−0.18, +0.84]	2.5×
Llama-3.1-8B	int4	downstream	59	1.35	83	0.19	+1.16 [+0.79, +1.53]	7.1×
Llama-3.1-8B	int4	keys-only	59	1.84	83	0.29	+1.55 [+0.98, +2.14]	6.3×
Qwen2.5-3B	int3	upstream	72	9.86	70	7.98	+1.88 [+1.02, +2.75]	1.2×
Qwen2.5-3B	int3	downstream	72	7.44	70	3.30	+4.14 [+3.11, +5.22]	2.3×
Qwen2.5-3B	int3	keys-only	72	9.75	70	8.04	+1.70 [+0.72, +2.68]	1.2×
Qwen2.5-3B	int4	upstream	72	2.03	70	0.46	+1.57 [+0.60, +2.59]	4.4×
Qwen2.5-3B	int4	downstream	72	1.91	70	0.81	+1.11 [+0.40, +1.81]	2.4×
Qwen2.5-3B	int4	keys-only	72	3.14	70	1.05	+2.09 [+1.05, +3.15]	3.0×

Table 28: The edit's own effect, edited minus unedited margin on the same prompt, and how much of it each channel removes (MEMIT, $n = 1$, 200 probes, paired 95% CI; from the split runs of Appendix G, so "raw" can differ from Table 11 by up to 0.03 nats). "raw" is the keys-only loss of the edited margin itself; shares are each half's effect lost over the keys-only effect lost (the halves interact sub-additively, so their shares can sum past 100%).

model	quant	raw	effect lost, keys-only	upstream	downstream	up. share	down. share
GPT-2-XL	int3	6.52	+7.05 [+6.50, +7.61]	+5.16 [+4.67, +5.68]	+2.88 [+2.59, +3.19]	73%	41%
GPT-2-XL	int4	1.13	+1.22 [+1.03, +1.41]	+0.81 [+0.67, +0.97]	+0.45 [+0.35, +0.56]	66%	37%
Llama-3.2-3B	int3	6.73	+8.12 [+7.56, +8.70]	+4.40 [+3.93, +4.88]	+5.37 [+4.96, +5.80]	54%	66%
Llama-3.2-3B	int4	1.56	+1.90 [+1.62, +2.19]	+0.75 [+0.55, +0.94]	+1.17 [+0.98, +1.37]	39%	62%
GPT-J-6B	int3	4.69	+5.33 [+4.87, +5.80]	+2.38 [+2.11, +2.67]	+2.85 [+2.50, +3.20]	45%	53%
GPT-J-6B	int4	0.65	+0.75 [+0.58, +0.94]	+0.41 [+0.30, +0.51]	+0.39 [+0.26, +0.52]	54%	51%
Llama-3.1-8B	int3	8.14	+10.20 [+9.56, +10.88]	+7.16 [+6.41, +7.94]	+6.35 [+5.88, +6.83]	70%	62%
Llama-3.1-8B	int4	1.88	+2.07 [+1.73, +2.41]	+0.64 [+0.37, +0.91]	+1.50 [+1.27, +1.74]	31%	73%
Qwen2.5-3B	int3	14.80	+19.73 [+18.75, +20.71]	+19.70 [+18.71, +20.70]	+11.39 [+10.66, +12.12]	100%	58%
Qwen2.5-3B	int4	4.22	+4.77 [+4.04, +5.54]	+2.68 [+2.03, +3.37]	+3.34 [+2.87, +3.82]	56%	70%

I UNLEARNING

An unlearning update suppresses a fact the model knows; a knowledge edit replaces it with another. Both are the same closed form with a different target value, so if the mechanism is about what the edit is conditioned on rather than about what it says, it should carry over. The script `code/make_unlearn.py` builds the unlearning variant of the same 1,600 CounterFact records by replacing the target with the placeholder "unknown" and leaving `target_true` alone, so the rewrite margin $\log P(\text{placeholder}) - \log P(\text{true object})$ is positive when the placeholder is preferred to the true object. Editor, closed form and probes are unchanged; the run covers GPT-2-XL and Llama-3.2-3B at int3, $n \in \{1, 100\}$, in the fp, upstream, downstream and deployed channels (no

weights-only or keys-only channel, so this run does not isolate the rounding of the update and gives no conditioning share: the evidence here is the size of the loss, its upstream/downstream split and QAE's recovery of it).

Table 29 is the result. At int3 the suppression margin is largely undone (MEMIT loses 10.33 and 12.91 of 15.39- and 18.94-nat margins on GPT-2-XL and Llama-3.2-3B), while deployed efficacy stays at 0.80–0.90 (unquantized 0.99–1.00, not tabulated), i.e. one suppression in five to ten fails outright and the rest are weakened. This is the direction of the failure Zhang et al. (2025) report, not a reproduction of it: theirs is gradient-based unlearning at 4 bits on unlearning benchmarks, ours a closed-form placeholder edit at int3 on CounterFact, and we do not run int4 or NF4 here. The split is upstream-heavy on GPT-2-XL (the upstream channel alone takes 8.30 nats against 4.43 downstream) and balanced on Llama-3.2-3B (8.75 against 8.36), where the knowledge edits' split leans downstream (3.54 against 4.69 nats; Appendix G). QAE returns 78–103% of it and lifts deployed efficacy to 0.99–1.00. QAE's single suppressions cost no measurable neighbourhood score (ΔNS $+0.000$ $[-0.015, +0.015]$ and -0.005 $[-0.030, +0.020]$); batches of 100 do (-0.100 $[-0.160, -0.050]$ and -0.340 $[-0.430, -0.250]$); the key-aware solve alone pays at least as much there (-0.130 $[-0.200, -0.070]$, -0.400 $[-0.500, -0.300]$) and the value-aware solve alone does not ($+0.020$, $+0.030$), so this is a cost of the key-aware half at batch scale, whose cause we have not isolated. The QAE unlearning update also barely suppresses the fact at full precision (fp margin 3.87 and 1.12 at $n = 1$, 1.96 and 0.41 at $n = 100$, against MEMIT's 15.39 and 18.94 at $n = 1$ and 15.84 and 19.29 at $n = 100$), so a checkpoint solved for int3 and served unquantized would be unlearned far more weakly: QAE unlearning is specific to the precision it was solved for. The "recovered" fractions are, as for edits, the objective QAE optimizes, and the placeholder margin measures a preference for "unknown" over the true object rather than the removal of the fact.

Table 29: Unlearning at int3: rewrite margin in nats in the fp, upstream ("up"), downstream ("down") and deployed channels, the deployed damage against MEMIT's unquantized margin (paired 95% CI), the fraction of MEMIT's deployed damage each editor returns, deployed efficacy and neighbourhood score, and ΔNS against MEMIT (intervals for QAE, and for the key-aware solve at $n = 100$, in the text). +key (KAE), +value and +both (QAE) as in Table 2. 200 probes at $n = 1$, 100 at $n = 100$.

model	n	editor	fp	up	down	deployed	damage	recovered	ES	NS	ΔNS
GPT-2-XL	1	MEMIT	15.39	7.10	10.96	5.07	$+10.33$ $[+9.73, +10.92]$	0%	0.80	0.82	—
GPT-2-XL	1	+key	10.54	13.87	7.56	10.21	$+5.19$ $[+4.72, +5.66]$	50%	0.98	0.82	$+0.005$
GPT-2-XL	1	+value	7.62	3.42	12.31	7.37	$+8.02$ $[+7.28, +8.79]$	22%	0.87	0.82	$+0.000$
GPT-2-XL	1	**+both (QAE)**	3.87	9.99	7.55	15.74	-0.35 $[-0.84, +0.13]$	103%	0.99	0.82	$+0.000$
GPT-2-XL	100	MEMIT	15.84	7.92	11.47	5.84	$+10.00$ $[+9.22, +10.78]$	0%	0.85	0.78	—
GPT-2-XL	100	+key	7.95	14.20	5.75	10.69	$+5.15$ $[+4.51, +5.82]$	48%	0.98	0.65	-0.130
GPT-2-XL	100	+value	7.73	3.83	12.49	7.72	$+8.12$ $[+7.09, +9.17]$	19%	0.88	0.80	$+0.020$
GPT-2-XL	100	**+both (QAE)**	1.96	9.92	5.03	15.12	$+0.72$ $[-0.01, +1.47]$	93%	1.00	0.68	-0.100
Llama-3.2-3B	1	MEMIT	18.94	10.19	10.58	6.03	$+12.91$ $[+12.13, +13.68]$	0%	0.90	0.88	—
Llama-3.2-3B	1	+key	10.56	17.79	6.46	10.10	$+8.84$ $[+8.13, +9.55]$	32%	0.98	0.86	-0.015
Llama-3.2-3B	1	+value	4.28	3.06	12.15	11.20	$+7.74$ $[+6.96, +8.52]$	40%	0.99	0.90	$+0.020$
Llama-3.2-3B	1	**+both (QAE)**	1.12	6.88	6.16	18.35	$+0.59$ $[-0.03, +1.22]$	95%	1.00	0.88	-0.005
Llama-3.2-3B	100	MEMIT	19.29	11.06	10.96	7.02	$+12.27$ $[+11.34, +13.21]$	0%	0.88	0.82	—
Llama-3.2-3B	100	+key	8.76	17.35	5.50	10.64	$+8.65$ $[+7.78, +9.54]$	30%	0.97	0.42	-0.400
Llama-3.2-3B	100	+value	4.45	3.52	12.14	11.28	$+8.01$ $[+7.03, +9.04]$	35%	0.98	0.85	$+0.030$
Llama-3.2-3B	100	**+both (QAE)**	0.41	7.76	4.96	16.53	$+2.76$ $[+1.94, +3.60]$	78%	1.00	0.48	-0.340

J GRADIENT-BASED UNLEARNING: WHERE ROUNDING DOES ERASE THE UPDATE

The sub-bin explanation was proposed for gradient-based unlearning (Zhang et al., 2025), which our closed-form placeholder edits do not test. Released checkpoints let us test it without training: the TOFU (Maini et al., 2024) target model released by open-unlearning (Dorna et al., 2025; Llama-3.2-1B-Instruct fine-tuned on TOFU, `tofu_Llama-3.2-1B-Instruct_full`; `results/unlearn_sweep.json` lists the 28 unlearned checkpoints of Table 32 by their released names, and `results/unlearn_channels.json` the four of Table 30) and checkpoints unlearned from it on forget10 with NPO (Zhang et al., 2024) and GradDiff at learning rates 10^{-5} and 5×10^{-5} (ten epochs; the two GradDiff checkpoints also differ in their retain weight, $\alpha = 5$ and 2, and Table 32 compares learning rates at a fixed $\alpha = 2$, where the 10^{-5} GradDiff checkpoint

touches the same 8.1% of entries and int4 delivers the same 0.05 of its update), all stored in bf16. The update $\Delta W = W_{\text{unlearned}} - W_{\text{target}}$ spans all 112 linear matrices of the decoder blocks but changes only 8–37% of their entries, and they are quantized with int4 round-to-nearest or NF4, group 64 (int3 destroys this 1B model: the target's retain-set NLL goes from 0.137 to 4.78, against 0.673 at int4, and int4 raises the unlearned checkpoints' retain-set NLL 1.6–5.5×, Table 30). The deployed matrix splits exactly as for an edit, $\mathcal{Q}(W+\Delta W) = W + \Delta W + [\mathcal{Q}(W) - W] + \tilde{E}$, so the weight channel (the update as rounding delivers it, $W + \mathcal{Q}(W+\Delta W) - \mathcal{Q}(W)$, on an exact base) and the conditioning (the exact update on a quantized base, $\mathcal{Q}(W) + \Delta W$) can be run separately (`code/unlearn_channels.py`). Unlike an edit's weights-only channel, which quantizes the edited matrix together with its base, this weight channel keeps every base exact: all 112 matrices carry part of the update, so the base error of every one of them is counted in the conditioning. The unlearning effect is the forget-set NLL per answer token of the unlearned model minus the target's on the same base, teacher-forced on the first 160 forget10 questions (retain-set NLL on the first 60 retain90 questions; 200 and 100 in the sweep of Table 32) with the checkpoints' chat template; Table 30 gives the fraction of it each channel keeps.

At the small learning rate the sub-bin explanation is right, and the lemma says why. At 10^{-5} rounding the update alone keeps 2–29% of the unlearning at int4 and 27–69% at NF4 (NF4's codebook is densest near zero, which presumably narrows its zero cell and weakens the erasure; we have not measured NF4's boundary density), while quantizing the base under an exact update keeps 88–166% of it; deployed, the unlearning is gone at int4 (-24% to 14% kept) and halved at NF4 (39–53%), the failure Zhang et al. (2025) report; here the weight channel is the larger single factor; for NPO the deployed loss exceeds either factor alone, while for GradDiff quantizing the base partly offsets the weight channel, an interaction we do not decompose. Table 31 shows the mechanism (`code/unlearn_lattice.py`). Stored in bf16, which keeps about eight significant bits per weight, an update of this size survives only where the weight is itself near zero (whether the rest is lost at each bf16 optimizer step, the update cancellation of low-precision training (Zamirai et al., 2020), or when the checkpoint is saved, released checkpoints cannot tell): the checkpoint changes 98.0–98.4% of the weights below a tenth of an int4 step and 0.15–0.30% of those beyond half a step. A near-zero weight sits at the centre of the zero bin, as far from a rounding boundary as a weight can be, so the changed entries' phase density at the boundary is 0.111–0.176 of uniform, and int4 rounding delivers 0.050–0.105 of the update (pooled over the 112 matrices, $\sum\langle \mathcal{Q}(W+\Delta W) - \mathcal{Q}(W), \Delta W\rangle / \sum\|\Delta W\|^2$). This is assumption (ii) of Lemma 1 failing in a specific way: phases are close to uniform over a whole matrix (Appendix B), but not over the entries a small stored update touches, so the phase is not independent of the displacement.

At five times the learning rate the lemma holds and the loss moves to the quantized base. At 5×10^{-5} the stored NPO update changes 36.6% of the entries, the boundary density of the changed entries is 0.971, and int4 delivers 0.996 of the update, as the lemma predicts on that grid (on the int3 grid, which this model does not survive, it delivers only 0.73, so the lemma's assumptions fail there as well; the boundary density was measured on the int4 grid only). Rounding it then keeps 98–100% of the unlearning, and what the deployed model loses (68–88% kept) comes from quantizing the base (72–89% kept; the partition defined above, unlike an edit's, puts the base error of every updated matrix in the conditioning, so this is a loss to the quantized base, not the key and read-out loss of an edit). The GradDiff checkpoint at 5×10^{-5} collapses on the forget set (its NLL rises by 106.4 nats per token) and keeps 98–99% of that in every channel.

Eight methods, every released learning rate. Table 32 repeats the split on 28 released checkpoints, in a separate GPU run on the first 200 questions (the three checkpoints shared with Table 30, NPO at both rates and GradDiff at 5×10^{-5}, differ by at most 3 points in "kept" and in E_{fp} by at most 0.02 nats on NPO and 0.16 nats (0.15%) on GradDiff's E_{fp} of 106, all of it from the 40 added questions: on the 160 both score, their unquantized NLLs agree to 3×10^{-4} nats; Table 32's GradDiff runs all have $\alpha = 2$): GradDiff, NPO (Zhang et al., 2024), SimNPO (Fan et al., 2025), AltPO (Mekala et al., 2025), IdkNLL, IdkDPO (GradDiff and an "I don't know" objective are among TOFU's baselines, Maini et al., 2024), RMU (Li et al., 2024) and UNDIAL (Dong et al., 2025) as implemented by open-unlearning, at every learning rate open-unlearning releases for each, with the other hyperparameters fixed (epoch 10). At 10^{-5} int4 rounding delivers 0.05–0.29 of the stored update on seven of the eight methods (0.59 for RMU, whose update touches 0.9% of the entries), and the whole pattern of NPO (rounding erases the update at 10^{-5} and keeps it at the higher rates, where the deployed loss follows the conditioning) holds on five (GradDiff, NPO, SimNPO except at 2×10^{-5}, IdkDPO and

UNDIAL; not RMU, not AltPO, which loses most of its unlearning to quantizing the base, and not IdkNLL, whose update has the same form but whose ratios are not interpretable, below): on those five and IdkNLL the stored update changes only 0.15–1.4% of the weights more than half a step from zero, and over the six methods whose unlearning moves the forget-set NLL by more than 0.1 nats (GradDiff, NPO, SimNPO, AltPO, IdkDPO and UNDIAL), rounding alone keeps a median 7% of it at int4 (1–34%; 7–10% for any threshold from 0.05 to 0.3 nats, RMU's E_{fp} being 0.047; medians of unrounded values throughout) while quantizing the base keeps a median 98% (32–143%; AltPO loses 68% to the base alone). At 2×10^{-5} and above (20 checkpoints) rounding delivers 0.66–1.02 of the update and keeps a median 98% of the unlearning, and the deployed loss follows the conditioning (median kept 76% by the conditioning alone, 75% deployed; without the four GradDiff checkpoints that collapse on the forget set, 98%, 74% and 71%); SimNPO at 2×10^{-5}, whose rounding keeps 49%, is the one case below 85%. IdkNLL and RMU at 10^{-5} barely move the forget-set NLL (0.08 and 0.05 nats), so their ratios are not interpretable. Over the 28 checkpoints the sweep's retain-set NLL (100 questions, not tabulated; the target's is 0.14 unquantized, 0.72 at int4 and 0.40 at NF4) is 0.12–2.37 unquantized, and deployment raises it 1.0–6.8× at int4 and 1.0–3.6× at NF4.

The boundary of the paper's claim is therefore not closed-form against gradient-based, but whether the stored update is correlated with the grid: an update small enough that the checkpoint's own precision keeps it only on near-zero weights is erased by rounding, as the sub-bin explanation says; an update that survives storage on typical weights is dithered, not erased, as Lemma 1 says, and the damage moves to the quantized base. This is consistent with the larger-learning-rate remedy that Zhang et al. (2025) propose, which we do not run. These are released checkpoints of one 1B model on one benchmark, measured by teacher-forced NLL rather than the benchmark's own forget-quality metrics; QAE is not applied to them.

Table 30: Gradient-based unlearning (TOFU forget10, Llama-3.2-1B-Instruct, released checkpoints) under quantization. E_{fp} is the unlearning effect, forget-set NLL per answer token of the unlearned model minus the target's; the next three columns give the fraction of it kept when the model is deployed quantized (against the quantized target), when only the update is rounded on an exact base, and when only the base is quantized under an exact update (ratio of means over 160 questions, 95% CI). Last column: retain-set NLL unquantized / deployed. The two GradDiff checkpoints also differ in α; Table 32 compares learning rates at $\alpha = 2$.

checkpoint	quant	E_{fp}	kept, deployed	weight channel only	conditioning only	retain NLL
GradDiff, 10^{-5}, α=5	NF4	0.401	39% [35, 45]	27% [24, 30]	113% [102, 128]	0.128 / 0.449
GradDiff, 10^{-5}, α=5	int4	0.401	14% [12, 16]	2% [1, 2]	166% [144, 193]	0.128 / 0.706
GradDiff, 5×10^{-5}, α=2	NF4	106.375	99% [99, 99]	99% [99, 100]	99% [99, 99]	0.626 / 0.782
GradDiff, 5×10^{-5}, α=2	int4	106.375	98% [97, 98]	99% [98, 99]	98% [98, 98]	0.626 / 1.001
NPO, 10^{-5}, β=0.5, α=1	NF4	0.452	53% [50, 57]	69% [66, 71]	92% [88, 95]	0.148 / 0.387
NPO, 10^{-5}, β=0.5, α=1	int4	0.452	−24% [−30, −19]	29% [26, 32]	88% [83, 94]	0.148 / 0.551
NPO, 5×10^{-5}, β=0.5, α=1	NF4	1.327	88% [86, 90]	100% [99, 101]	89% [87, 91]	0.294 / 0.442
NPO, 5×10^{-5}, β=0.5, α=1	int4	1.327	68% [65, 72]	98% [97, 99]	72% [69, 75]	0.294 / 0.597

Table 31: Where the stored update sits on the int4 grid, over all 112 quantized matrices: the fraction of entries the bf16 checkpoint changed, the same by the target model's weight magnitude in int4 steps, the probability mass of the changed entries' grid phase within 0.03 of a rounding boundary relative to uniform (1 under Lemma 1's assumption (ii)), and the fraction of the update rounding delivers on the int4 grid and (last column) on the int3 grid, pooled over the 112 matrices, $\sum\langle \mathcal{Q}(W + \Delta W) - \mathcal{Q}(W), \Delta W\rangle / \sum \|\Delta W\|^2$.

checkpoint	changed	$\lvert W\rvert < 0.1$ step	0.1–0.5	> 0.5	boundary density	delivered, int4	int3
NPO, 10^{-5}, β=0.5, α=1	8.6%	98.4%	42.5%	0.30%	0.176	0.105	0.006
GradDiff, 10^{-5}, α=5	8.1%	98.0%	39.8%	0.15%	0.111	0.050	0.002
NPO, 5×10^{-5}, β=0.5, α=1	36.6%	99.7%	97.6%	25.35%	0.971	0.996	0.728
GradDiff, 5×10^{-5}, α=2	35.3%	99.7%	96.9%	23.91%	0.982	0.969	0.725

Table 32: Released TOFU unlearning checkpoints of Llama-3.2-1B-Instruct (forget10), eight methods, every released learning rate at one setting of the other hyperparameters (epoch 10; GradDiff at $\alpha = 2$; every released name, with its hyperparameters, is in `results/unlearn_sweep.json`). E_{fp}: forget-set NLL of the unlearned minus the target model (nats/token); kept: the share of E_{fp} that survives deployment (dep.), rounding of the update alone (w.) or quantizing the base alone (cond.), as in Table 30; touched: entries the stored update changes, overall and among the weights more than half an int4 step from zero; delivered: the fraction of the update int4 rounding delivers, pooled over the 112 matrices as in Table 31 (Lemma 1: 1).

method	lr (10^{-5})	E_{fp}	int4 dep.	int4 w.	int4 cond.	NF4 dep.	NF4 w.	NF4 cond.	touched	> 0.5 step	delivered
GradDiff	1	2.22	7%	1%	96%	33%	32%	80%	8.1%	0.16%	0.05
GradDiff	2	100.25	76%	94%	85%	95%	98%	95%	16.2%	3.74%	0.77
GradDiff	3	98.33	86%	98%	93%	98%	99%	98%	22.9%	10.09%	0.99
GradDiff	4	100.05	96%	98%	98%	98%	99%	99%	29.4%	17.19%	0.98
GradDiff	5	106.53	97%	99%	98%	99%	99%	99%	35.3%	23.91%	0.97
NPO	1	0.46	−21%	29%	91%	54%	69%	92%	8.6%	0.30%	0.10
NPO	2	0.85	69%	98%	73%	79%	95%	80%	16.5%	3.90%	0.79
NPO	5	1.31	70%	98%	74%	89%	100%	90%	36.6%	25.35%	1.00
SimNPO	1	0.44	13%	2%	143%	40%	30%	101%	8.1%	0.15%	0.05
SimNPO	2	1.90	39%	49%	74%	66%	85%	75%	15.7%	3.37%	0.66
SimNPO	5	2.17	67%	99%	68%	82%	99%	81%	35.7%	24.32%	1.00
AltPO	1	0.59	−10%	10%	32%	57%	77%	73%	9.0%	0.34%	0.11
AltPO	2	1.38	83%	89%	63%	76%	96%	79%	16.1%	3.67%	0.70
AltPO	5	2.96	72%	99%	72%	85%	99%	85%	34.7%	23.13%	1.01
IdkNLL	1	0.08	252%	116%	317%	159%	97%	191%	11.8%	1.42%	0.29
IdkNLL	2	0.14	210%	122%	220%	128%	101%	152%	19.2%	6.00%	0.96
IdkNLL	3	0.25	112%	110%	125%	108%	104%	111%	27.4%	14.93%	1.02
IdkNLL	4	0.38	78%	108%	83%	96%	105%	93%	33.5%	21.73%	1.01
IdkNLL	5	0.50	64%	106%	65%	92%	105%	89%	39.0%	28.08%	0.99
IdkDPO	1	0.69	17%	34%	99%	58%	65%	89%	11.4%	1.27%	0.23
IdkDPO	2	1.46	70%	88%	78%	75%	91%	81%	18.8%	5.58%	0.94
IdkDPO	5	2.15	68%	96%	72%	83%	100%	82%	39.1%	28.23%	0.99
RMU	1	0.05	508%	66%	641%	324%	93%	349%	0.9%	0.18%	0.59
RMU	2	2.72	139%	107%	131%	108%	100%	109%	1.7%	1.05%	1.02
RMU	5	10.28	89%	97%	91%	89%	100%	90%	3.2%	2.84%	0.98
UNDIAL	1	0.27	18%	5%	141%	58%	55%	109%	7.8%	0.17%	0.07
UNDIAL	10	1.94	71%	100%	70%	87%	101%	84%	62.4%	55.62%	0.99
UNDIAL	30	2.37	73%	98%	74%	89%	98%	89%	93.9%	92.85%	1.00

K THE WEIGHT-CHANNEL DEFENCES, RUN AS POST-HOC TRANSFORMS

Three quantization-robust unlearning methods target this failure: a floor that lifts update entries to the step (Sadhu et al., 2026), a hinge loss in logit space whose margin is half the quantization step (Mishra & Mehreen, 2026), and straight-through training at the target precision (Khan & Sohel, 2026). We test the floor and a weight-space analogue of the hinge (not the logit-space objective itself) as post-hoc transforms of a closed-form update rather than as the published training objectives, which are unlearning objectives we do not train. `code/magnitude_defences.py` applies each to MEMIT's own update (*floor* Δ and *floor* $\Delta/2$ lift every entry below that threshold out to it; *hinge* sets every entry with $|\delta| < \Delta/2$ to $\pm 0.5001\Delta$, whatever the weight's position in its bin; *rescale*, our own control, multiplies the whole update by one factor, never below 1, until its mean per-entry magnitude equals the mean step of the edited matrix) and evaluates each in the deployed channel against plain MEMIT on the same 200 probes. The hinge's dead zone $|\delta| < \Delta/2$ is the sub-bin explanation's notion, exact only for a weight at the centre of its bin (Appendix B); pushed just past it (to $\pm 0.5001\Delta$, against the half-step floor's $\pm 0.5\Delta$), the hinge nearly coincides with the half-step floor, and their rows agree to 0.02 nats, as the construction implies, so the two are one intervention rather than two; a hinge that uses each weight's own distance to its boundary is not run. The transforms are applied at $n = 1$, where the weight channel is small, and not in the $n = 1000$ or small-learning-rate unlearning regimes where it is not (Appendix J); the defences' premise, that rounding erases the update, holds only in the latter, since at $n = 1000$ the update coarsens the grid it is rounded on (Appendix B). Straight-through training (Khan & Sohel, 2026) is not run.

Table 33 is the result, and it is not that the defences do nothing. They enlarge the update by 3 to $15\times$ MEMIT's norm (the rescale up to $30\times$), and on GPT-2-XL at int3 that buys deployed margin ($+1.65$ to $+4.58$ nats), and at int4 all but the rescale buy $+0.99$ to $+1.99$ (the rescale loses 1.64). This cannot come from removing MEMIT's rounding error, since MEMIT's own weight channel in these cells is at most 0.05 nats (Table 11). The run scores only the deployed channel, so the defended edits' fp margins are not measured and their updates are not decomposed by channel: whether the gains come from a larger fp margin or from a smaller conditioning loss is not measured, except that at int4 the half-step floor and the hinge deploy at 13.69 nats, above MEMIT's unquantized 12.84 (Table 2), so at least 0.85 of their $+1.99$ nats is not MEMIT's int4 loss (1.14 nats, Table 19) removed. On Llama-3.2-3B the floors and the hinge lose margin significantly in four of their six cells (three of the four floor cells, the hinge repeating the half-step floor) and the rescale in one of its two (-1.25 to -4.72; at int3 only the full-step floor), and the other three cells lose it not significantly, so the sign is not stable across two models. The locality cost is stable: every defence costs locality, ΔNS from -0.020 to -0.605 with the interval below zero in 14 of 16 cells (10 of the 12 floor and hinge cells, -0.020 to -0.505), and on GPT-2-XL at int3 the full-step floor turns the neighbourhood margin from $+2.09$ to -2.12: the model now prefers the edited object on unrelated subjects. That follows from the size of the update, not from Corollary 2, which concerns the readout at the edit's own key: an update 3–$30\times$ larger moves the outputs of unrelated keys more the larger it is (ΔNS falls with the norm ratio in each of the four model–precision blocks, three distinct norms each), though not in proportion. QAE, on that same model and cell, buys $+6.49$ nats (in-sample for QAE; Table 19) with no magnitude intervention at all and ΔNS -0.015 $[-0.035, +0.000]$.

Table 33: The magnitude defences applied to MEMIT's update and evaluated in the deployed channel, $n = 1$, 200 probes, paired against plain MEMIT in the same run (a run separate from the grid, reusing the k^*/v^* cache of the deployment-precision run of Table 2, whose MEMIT margins it reproduces to 0.01 nats on average, so MEMIT's NS can differ from Table 39's by one probe). NS is the deployed neighbourhood score and the last column its mean margin in nats.

model	quant	defence	$\|\Delta W\|_F/\|\Delta W_{\text{MEMIT}}\|_F$	deployed	vs MEMIT	NS	ΔNS	neigh. margin
GPT-2-XL	int3	MEMIT	1.00	6.36	—	0.795	—	+2.09
GPT-2-XL	int3	floor Δ	14.73	9.82	+3.46 [+2.75, +4.18]	0.290	−0.505 [−0.575, −0.435]	−2.12
GPT-2-XL	int3	floor $\Delta/2$	7.39	10.94	+4.58 [+3.90, +5.29]	0.585	−0.210 [−0.265, −0.155]	+0.43
GPT-2-XL	int3	hinge	7.39	10.94	+4.58 [+3.90, +5.27]	0.585	−0.210 [−0.265, −0.155]	+0.43
GPT-2-XL	int3	rescale	29.89	8.00	+1.65 [+0.81, +2.50]	0.250	−0.545 [−0.615, −0.475]	−2.95
GPT-2-XL	int4	MEMIT	1.00	11.71	—	0.825	—	+2.36
GPT-2-XL	int4	floor Δ	6.34	12.70	+0.99 [+0.44, +1.53]	0.685	−0.140 [−0.190, −0.095]	+1.18
GPT-2-XL	int4	floor $\Delta/2$	3.24	13.69	+1.99 [+1.57, +2.40]	0.765	−0.060 [−0.095, −0.030]	+1.89
GPT-2-XL	int4	hinge	3.24	13.69	+1.99 [+1.57, +2.40]	0.765	−0.060 [−0.095, −0.030]	+1.89
GPT-2-XL	int4	rescale	12.81	10.06	−1.64 [−2.41, −0.88]	0.445	−0.380 [−0.450, −0.315]	−0.94
Llama-3.2-3B	int3	MEMIT	1.00	7.05	—	0.815	—	+2.49
Llama-3.2-3B	int3	floor Δ	14.26	5.81	−1.25 [−1.78, −0.70]	0.655	−0.160 [−0.220, −0.100]	+1.08
Llama-3.2-3B	int3	floor $\Delta/2$	7.14	6.79	−0.27 [−0.74, +0.21]	0.780	−0.035 [−0.080, +0.005]	+2.04
Llama-3.2-3B	int3	hinge	7.14	6.78	−0.27 [−0.75, +0.21]	0.780	−0.035 [−0.080, +0.005]	+2.04
Llama-3.2-3B	int3	rescale	24.62	6.68	−0.37 [−1.00, +0.26]	0.210	−0.605 [−0.675, −0.535]	−3.45
Llama-3.2-3B	int4	MEMIT	1.00	12.29	—	0.835	—	+3.31
Llama-3.2-3B	int4	floor Δ	6.12	7.57	−4.72 [−5.25, −4.20]	0.775	−0.060 [−0.100, −0.025]	+2.65
Llama-3.2-3B	int4	floor $\Delta/2$	3.11	10.32	−1.97 [−2.37, −1.57]	0.815	−0.020 [−0.040, −0.005]	+3.05
Llama-3.2-3B	int4	hinge	3.11	10.32	−1.97 [−2.38, −1.57]	0.815	−0.020 [−0.040, −0.005]	+3.05
Llama-3.2-3B	int4	rescale	10.55	9.65	−2.64 [−3.26, −2.01]	0.455	−0.380 [−0.445, −0.315]	−1.45

L MULTIPLICITY AND THE DIFFICULTY CONFOUND

Multiplicity. This paper reports several hundred paired 95% intervals and marks the ones that exclude zero. Two families invite a correction; both, and the served-quantizer contrasts as a third, are corrected by Benjamini–Hochberg (Benjamini & Hochberg, 1995; $q = 0.05$, bootstrap p-values; `code/multiplicity.py`). The fix's own contrasts are untouched: of the 105 deployed-margin contrasts against MEMIT in the deployment-precision runs on the symmetric grid and the GPTQ backbone (key-aware, value-aware and both, in the 23 cells of Tables 19 and 20, the four of Table 29 and the eight of Table 55, whose value-aware arm is printed only in `results/dae_*_gptq.jsonl`), 98 are significant uncorrected and 98 survive; the served-quantizer, 13B and released-editor contrasts are not in the family. Corrected as a family of their own, the 120 contrasts of the served-quantizer runs (three editors in the 40 cells of `results/prod_dae_<model>.jsonl`, four models) give 111 significant uncorrected and all 111 survive, among them all 32 of QAE's GPTQ and AWQ contrasts. The locality cost we report loses a little: of the 78 cells in which KAE's neighbourhood

margin is compared with MEMIT's, 32 (31 losses and one gain, GPT-2-XL/ZsRE at int3, $n = 100$) are significant uncorrected and 29 survive (largest rejected $p = 0.0149$; the three that do not are GPT-J-6B at NF4, $n = 100$, Llama-3.1-8B at int4, $n = 1$, and Llama-3.2-3B at int3, $n = 1000$). A bootstrap p-value is twice the smaller of the fractions of resampled means at or below zero and at or above it, so a resampled mean of exactly zero counts on both sides. Every other interval in the paper is reported per comparison and uncorrected, and should be read that way.

Is the bin-width criterion's sub-chance AUROC only edit difficulty? A hard fact needs a large update ($\|\Delta W\|_F = \|a\|/(\|k^*\| \cos(u, k^*))$) and fails at every precision, which would produce an AUROC below chance with no reference to quantization at all. Recomputing it over only the edits that already succeed unquantized closes the crudest form of that path (not the finer one, a small fp margin among successes), and it does not change the picture. Table 34 uses the same subsets and orientation as the body: over the 18 of the 19 cells with at least 20 deployed failures that have a defined fp-success AUROC, conditioning on unquantized success moves the AUROC by at most 0.067, and all 14 cells that were below chance remain below chance (one of them, Qwen2.5-3B int4, only just: 0.487 over all edits and 0.467 over the fp successes), and Qwen2.5-3B NF4 crosses from 0.502 to 0.439. On Qwen2.5-3B and on ZsRE the criterion sits near 0.5 (0.49–0.60 over all edits, 0.44–0.60 over the fp successes): uninformative rather than inverted, which we report rather than pool away. The cells pool every editor run on that grid (six; three on ZsRE) and every batch size at a given model and quantizer, and the inversion lives in that pooling: configurations with larger $\bar{\delta}/\Delta$ (larger batches, less aligned editors) fail more. With editor, quantizer and batch size all fixed, the criterion cannot rank the edits of a batch at all: every edit of a batch shares the batch's $\bar{\delta}/\Delta$, so in the 12 batch cells among the 18 per-configuration MEMIT cells (editor, quantizer and batch size fixed; five models and ZsRE) with at least ten deployed failures and ten successes its AUROC is 0.5 by construction, ties counting one half. In the six single-edit cells it is 0.344 $[0.218, 0.477]$ on GPT-2-XL at int3, 0.635 $[0.517, 0.748]$ on Llama-3.1-8B at int3, 0.594 $[0.513, 0.671]$ and 0.433 $[0.275, 0.597]$ on Qwen2.5-3B at int3 and int4, and 0.487 $[0.370, 0.602]$ and 0.491 $[0.376, 0.607]$ on ZsRE at int3 (GPT-2-XL, Llama-3.2-3B; stratified bootstrap): median 0.489, and of the three intervals that exclude 0.5 one points the wrong way and two the criterion's way, so it is not a consistent per-edit predictor (the eleven cells on the four main models are those of Appendix X, which scores them with $-\mathrm{ESNR}^2$ rather than $-\bar{\delta}/\Delta$, hence its different verdicts). This is expected when deployed failures are driven by the conditioning (§4.1), which per-parameter magnitude does not see; across configurations it points the wrong way.

Table 34: The bin-width criterion's AUROC for deployed efficacy failure, over all edits and over only the edits that succeed unquantized. Score is $-\bar{\delta}/\Delta$, so below 0.5 means the criterion calls the surviving edits the doomed ones. Cells pool all editors and batch sizes; shown are the cells with at least 20 deployed failures (Llama-3.2-3B at int8, whose 30 failures all fail unquantized too, has no fp-success AUROC and is omitted; its AUROC over all edits is 0.121).

model	quant	deployed failures	AUROC, all	AUROC, fp successes
GPT-2 (124M)	int3	107	0.351	0.354
GPT-2-XL	NF4	179	0.156	0.152
GPT-2-XL	int3	486	0.272	0.305
GPT-2-XL	int4	212	0.165	0.169
GPT-2-XL	int8	167	0.162	0.220
GPT-J-6B	NF4	68	0.198	0.194
GPT-J-6B	int3	277	0.212	0.212
GPT-J-6B	int4	70	0.206	0.203
GPT-J-6B	int8	52	0.197	0.130
Llama-3.1-8B	int3	269	0.400	0.403
Llama-3.2-3B	NF4	31	0.184	0.230
Llama-3.2-3B	int3	244	0.272	0.279
Llama-3.2-3B	int4	55	0.123	0.123
Qwen2.5-3B	NF4	37	0.502	0.439
Qwen2.5-3B	int3	848	0.522	0.537
Qwen2.5-3B	int4	115	0.487	0.467
GPT-2-XL/ZsRE	int3	77	0.599	0.600
Llama-3.2-3B/ZsRE	int3	93	0.517	0.535

M GRANULARITY

Proposition 1 is written in the step map Δ and Proposition 2 depends on it through δk, and the group size sets Δ: the main grid uses group 64 (§4.2 uses 128); serving formats also use finer blocks (32 or 16), which we do not sweep. Table 35 repeats the four-channel ablation at $n = 1$ on 100 probes (MEMIT) over group sizes from 64 to per-channel, one scale per output row; per-channel groups are a factor of 100 (GPT-2-XL) and 128 (Llama-3.2-3B) coarser on the edited matrix and 25 and 48 on the backbone's input projections. The main grid's quantizer needs a group size that divides every d_{in} in the model (the served quantizer of Appendix E instead ends each 1600-wide GPT-2-XL row with a group of 64), which is why the two models use different intermediate values (GPT-2-XL's matrices are 1600 and 6400 wide, Llama-3.2-3B's 3072 and 8192).

Over all 16 cells the conditioning share stays between 96.3% and 100.5% and the weight channel never exceeds 7.8% of the total, while the total damage grows by a factor of 2.0 to 4.6 (int3 roughly doubles, int4 grows 3.3–$4.6\times$). The two per-channel int3 cells say little: they leave a mean deployed margin of 0.006 and 0.26 nats on GPT-2-XL and Llama-3.2-3B (margins are not clipped; single edits range from -8.4 to $+11.6$ and from -5.5 to $+6.2$) and deployed efficacy 0.51 and 0.56, so the keys-only channel alone already removes almost the whole margin and leaves the weight channel little to add (the unedited margin under this quantizer was not measured); the claim rests on the other cells. The quantity the half-bin account is written in moves as that account needs: as the grid coarsens, the mean step grows 1.7–$1.8\times$ and $\bar{\delta}/\Delta$ falls ($0.033 \to 0.019$ on GPT-2-XL at int3, $0.041 \to 0.024$ on Llama-3.2-3B), and the total loss grows; but the growth is in the keys-only channel, so the correlation it would cite is not the mechanism; the weight channel does grow with coarseness (to 0.26 and 1.06 nats at int3), as the larger step in Lemma 1 implies (the channel also carries the rounding of the edited matrix's own weights, which grows with the step too), but stays at most 7.8% of the total. What a coarser grid mostly does is degrade the conditioning (key and read-out; the channel is not split here), not round the update away. The sweep is $n = 1$ only; at $n = 1000$, where coarser groups would inflate the absmax further, it is not run.

Table 35: The four-channel ablation as a function of the quantization group size, $n = 1$, 100 probes, margin changes in nats relative to the same editor's fp margin (negative = margin lost; the group size does not change that margin: 13.21 on GPT-2-XL and 13.85 on Llama-3.2-3B in every row).

model	quant	group	$\bar{\delta}/\Delta$	weight channel	keys-only	full	K/F
GPT-2-XL	int3	64	0.033	$+0.01$ $[-0.02, +0.04]$	-6.39	-6.36	100.5%
GPT-2-XL	int3	160	0.029	-0.02 $[-0.06, +0.02]$	-8.16	-8.16	100.0%
GPT-2-XL	int3	400	0.025	-0.06 $[-0.10, -0.02]$	-10.74	-10.75	99.9%
GPT-2-XL	int3	per-channel (6400)	0.019	-0.26 $[-0.36, -0.17]$	-13.18	-13.21	99.8%
GPT-2-XL	int4	64	0.077	-0.00 $[-0.02, +0.01]$	-1.09	-1.09	99.9%
GPT-2-XL	int4	160	0.067	$+0.01$ $[-0.01, +0.02]$	-1.78	-1.78	100.3%
GPT-2-XL	int4	400	0.059	-0.01 $[-0.04, +0.01]$	-2.41	-2.44	98.9%
GPT-2-XL	int4	per-channel (6400)	0.043	-0.09 $[-0.12, -0.06]$	-4.84	-5.01	96.5%
Llama-3.2-3B	int3	64	0.041	-0.04 $[-0.10, +0.03]$	-6.65	-6.73	98.9%
Llama-3.2-3B	int3	128	0.037	-0.07 $[-0.14, -0.01]$	-7.87	-7.95	99.1%
Llama-3.2-3B	int3	256	0.034	-0.15 $[-0.24, -0.06]$	-9.35	-9.43	99.1%
Llama-3.2-3B	int3	per-channel (8192)	0.024	-1.06 $[-1.26, -0.87]$	-13.53	-13.59	99.5%
Llama-3.2-3B	int4	64	0.095	-0.01 $[-0.04, +0.02]$	-1.34	-1.33	100.4%
Llama-3.2-3B	int4	128	0.087	-0.02 $[-0.05, +0.01]$	-1.84	-1.83	100.2%
Llama-3.2-3B	int4	256	0.080	-0.01 $[-0.05, +0.02]$	-2.22	-2.23	99.6%
Llama-3.2-3B	int4	per-channel (8192)	0.057	-0.25 $[-0.32, -0.18]$	-4.27	-4.44	96.3%

N ALIGNMENT AND MAGNITUDE ACROSS EDITORS

Figure 4 plots, and Table 36 tabulates, the alignment and relative norm of every editor at $n = 1$ on the four main models. On Qwen2.5-3B MEMIT's alignment is 0.219 against a random-direction baseline of 0.0095. KAE's alignment with the *fp* key is lower than MEMIT's by design ($0.718 \to 0.531$ at int3 on GPT-2-XL, $0.574 \to 0.431$ at $n = 1000$), because it is aligned with the key that is read; its int3 rows are tabulated (at int8 it coincides with MEMIT, 0.718).

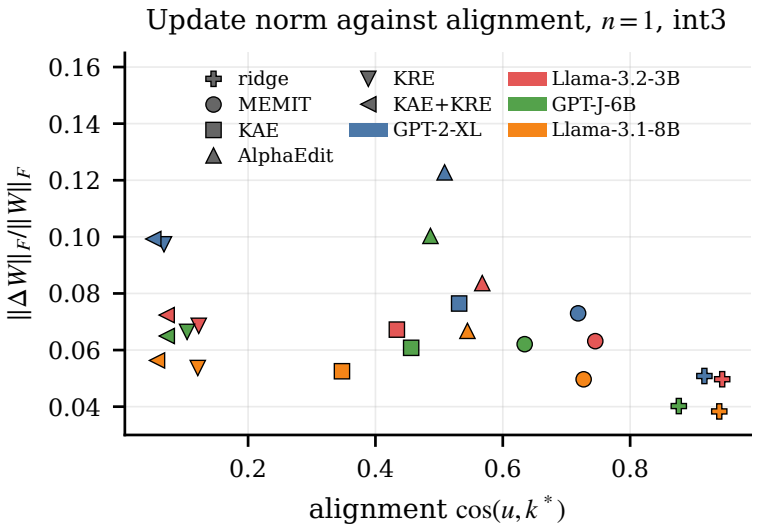


Figure 4: Every editor on the four main models at $n = 1$, int3: relative update norm against alignment. For the three rank-one editors that share a (ridge, MEMIT, AlphaEdit) the ordering by $\cos(u, k^*)$ within each model is the inverse of the ordering by norm, as $\|\Delta W\|_F = \|a\|/(\|k^*\| \cos(u, k^*))$ requires for a shared a (the identity holds per edit: the mean of $\cos(u, k^*)\|\Delta W\|_F/\|W\|_F$ is 0.046, 0.047, 0.035 and 0.036 for all three editors on the four models; the products of the tabulated means differ because means do not multiply, by up to 39% for AlphaEdit on GPT-2-XL and GPT-J, where two facts get $\cos(u, k^*) < 0.05$ and 35–47× the median norm). KAE is solved for a different residual at a different key, and KRE and KAE+KRE are five-key batch solves, so the identity does not bind them. KRE, which destroys alignment, is significantly harmful at scale (Table 37); the AlphaEdit-style variant is too, through a lower fp margin and the keys-only channel. "AlphaEdit" here is the single-layer variant of Appendix A.

Table 36: $\cos(u, k^*)$ against the fp key and $\|\Delta W\|_F/\|W\|_F$ at $n = 1$ (KAE and KAE+KRE solved for int3); the random-direction baseline $1/\sqrt{d_{\text{in}}}$ is 0.0125, 0.0110, 0.0078 and 0.0084 for the four models. All rows are the main grid at int3, as plotted in Figure 4; "AlphaEdit" is the single-layer AlphaEdit-style variant of Appendix A.

	GPT-2-XL		Llama-3.2-3B		GPT-J-6B		Llama-3.1-8B	
editor	cos	norm	cos	norm	cos	norm	cos	norm
alignment ridge	0.917	0.051	0.945	0.050	0.876	0.040	0.940	0.038
MEMIT	0.718	0.073	0.745	0.063	0.634	0.062	0.727	0.050
KAE (int3)	0.531	0.076	0.434	0.067	0.456	0.061	0.347	0.053
AlphaEdit	0.509	0.123	0.568	0.084	0.486	0.100	0.544	0.067
KRE	0.068	0.097	0.122	0.069	0.104	0.066	0.121	0.054
KAE+KRE (int3)	0.051	0.099	0.072	0.072	0.072	0.065	0.057	0.056

The other editors at deployment precision. Table 37 gives the deployed efficacy and paraphrase success of the alignment ridge, the AlphaEdit-style variant, KRE and KAE against MEMIT in the cells Table 2 does not show. The ridge's efficacy gains are at $n = 1000$, where the weight channel is material (in the two int3 cells with a gain it returns 2.10 and 1.83 nats of MEMIT's weight-channel loss and trims the keys-only loss by 0.03 and 0.45, Table 38; at $n = 1$ it adds a small, significant $+0.020$ in PS on GPT-2-XL); over the 36 int3, int4 and NF4 cells of the four main models, the AlphaEdit-style variant's efficacy is significantly below MEMIT's in seven (at $n = 1000$: GPT-2-XL at int4 and NF4, Llama-3.2-3B at int3 and int4, GPT-J-6B at int4 and Llama-3.1-8B at int3; at $n = 1$: GPT-2-XL at int3), KRE's in six, all at $n = 1000$ (GPT-2-XL at int4 and NF4, Llama-3.2-3B at int3, int4 and NF4, GPT-J-6B at int4), and neither's is significantly above it in any; Table 37 shows a selection; its KAE rows repeat, for comparison, KAE's ΔES and ΔPS from Table 39.

Table 37: Main grid, full channel: the editor's deployed ES and PS, and paired 95% CIs against MEMIT on the same probes (200; 100 at $n = 100$); bold = CI excludes zero; int3 unless marked; "ridge" is the alignment ridge and "AlphaEdit" the single-layer AlphaEdit-style variant of Appendix A.

model	editor	n	ES	PS	ΔES vs MEMIT	ΔPS vs MEMIT
GPT-2-XL	ridge	1	0.94	0.75	+0.015 [+0.000, +0.035]	**+0.020** [+0.005, +0.040]
GPT-2-XL	ridge	1000	0.78	0.61	**+0.145** [+0.085, +0.210]	**+0.135** [+0.080, +0.190]
GPT-2-XL	AlphaEdit	1	0.90	0.70	**−0.025** [−0.050, −0.005]	**−0.030** [−0.055, −0.010]
GPT-2-XL	AlphaEdit, int4	1000	0.76	0.52	**−0.125** [−0.190, −0.060]	**−0.095** [−0.165, −0.025]
GPT-2-XL	KRE, int4	1000	0.69	0.49	**−0.190** [−0.260, −0.120]	**−0.125** [−0.205, −0.045]
GPT-2-XL	KAE	100	1.00	0.88	**+0.060** [+0.020, +0.110]	**+0.150** [+0.080, +0.220]
Llama-3.2-3B	KAE	100	0.97	0.93	+0.030 [−0.010, +0.080]	**+0.070** [+0.010, +0.140]
Llama-3.2-3B	AlphaEdit	1000	0.78	0.69	**−0.085** [−0.140, −0.035]	**−0.075** [−0.130, −0.020]
Llama-3.2-3B	KRE	1000	0.79	0.69	**−0.070** [−0.125, −0.020]	**−0.070** [−0.125, −0.015]
Llama-3.2-3B	KRE, int4	1000	0.91	0.87	**−0.075** [−0.115, −0.035]	**−0.050** [−0.090, −0.010]
GPT-J-6B	ridge	1000	0.85	0.70	**+0.100** [+0.050, +0.150]	**+0.090** [+0.035, +0.145]
GPT-J-6B	AlphaEdit	1000	0.71	0.59	−0.040 [−0.105, +0.025]	−0.025 [−0.085, +0.035]
Llama-3.1-8B	KAE	100	0.97	0.93	**+0.070** [+0.020, +0.130]	**+0.110** [+0.050, +0.180]
Llama-3.1-8B	AlphaEdit	1000	0.77	0.69	**−0.085** [−0.140, −0.035]	**−0.090** [−0.145, −0.035]
Qwen2.5-3B	KAE	1	0.58	0.59	**+0.090** [+0.040, +0.145]	**+0.120** [+0.070, +0.170]
Qwen2.5-3B	AlphaEdit	1	0.49	0.47	+0.005 [+0.000, +0.015]	−0.005 [−0.025, +0.010]
Qwen2.5-3B	ridge	1	0.48	0.47	−0.005 [−0.025, +0.010]	−0.010 [−0.030, +0.010]
GPT-2-XL/ZsRE	KAE	1	0.98	0.94	**+0.120** [+0.075, +0.165]	**+0.185** [+0.135, +0.240]
GPT-2-XL/ZsRE	ridge	1	0.87	0.77	+0.005 [+0.000, +0.015]	**+0.020** [+0.005, +0.040]
Llama-3.2-3B/ZsRE	KAE	1	0.95	0.92	**+0.090** [+0.050, +0.135]	**+0.075** [+0.030, +0.120]
Llama-3.2-3B/ZsRE	ridge	1	0.89	0.84	+0.025 [+0.000, +0.050]	+0.000 [−0.020, +0.020]

Corollary 2 by intervention. The alignment ridge raises $\cos(u, k^*)$ and, by the identity of §3, shrinks the update with it (the two cannot be separated for a rank-one editor); Table 38 reports what happens to the weight channel. Its grid is a different set of 36: the int3 and int4 cells of the seven model/dataset tags (every batch size; GPT-2 and GPT-2-medium, which have no ridge arm, excluded). In the six cells where MEMIT's weight channel exceeds 0.6 nats (all at $n = 1000$ or on Qwen2.5-3B at $n = 100$, none a single edit) the ridge removes 73–95% of it, with every paired interval above zero; in the 29 cells where that channel is below 0.30 nats it moves it by at most 0.13 nats; the 36th, Llama-3.2-3B at $n = 1000$/int3 (exactly 0.30), is shown. Outside that grid the two NF4 cells with a weight channel above 0.3 nats (GPT-2-XL 0.62, GPT-J 0.31, both at $n = 1000$) lose 92% and 79% of it (so over the seven cells above 0.6 nats in any quantizer the range is again 73–95%, and elsewhere at most 0.24 nats moves). These cells are overloaded or Qwen's, outside the rank-one sub-bin hypotheses of Corollary 2, and the ridge's $\lambda = 3\bar{\lambda}$ also regularizes an ill-conditioned batch solve, so this is evidence that the ridge helps where the weight channel is large, not a test of the corollary's constant. The ridge also trims the keys-only channel, with the interval above zero in 25 of these 36 cells (40 of the 68 cells these tags have with a ridge arm, NF4 and int8 included) and by up to 0.56 nats (Llama-3.2-3B, int3, $n = 100$; 0.54 and 0.51 at $n = 1000$ on the two Llamas), although its $1/\mathrm{KSNR}$ at $n = 1$ is only slightly lower (0.277 against 0.291 on GPT-2-XL/int3, Appendix U); in one NF4 cell it moves the other way.

The AlphaEdit-style variant per channel. Its fp efficacy equals MEMIT's at $n = 1$ (0.995 / 0.995 on GPT-2-XL, 1.000 / 1.000 elsewhere) and its extra deployed loss there is in the keys-only channel: −0.16 [−0.21, −0.11] nats on GPT-2-XL, −0.20 [−0.26, −0.13] on Llama-3.2-3B, −0.05 [−0.07, −0.02] on GPT-J-6B and −0.23 [−0.31, −0.16] on Llama-3.1-8B, consistent with its higher $1/\mathrm{KSNR}$ (0.303 / 0.291, 0.461 / 0.442, 0.294 / 0.288, 0.526 / 0.512). At $n = 1000$ its fp margin is lower (5.02 / 7.25 on GPT-2-XL, 10.00 / 11.50 on Llama-3.2-3B, 7.16 / 9.04 on GPT-J, 12.61 / 13.34 on Llama-3.1-8B); its weight-channel loss in nats is smaller than MEMIT's on GPT-2-XL and GPT-J (1.23 / 2.56, 1.21 / 2.41) and slightly larger on the Llamas (0.35 / 0.30, 0.31 / 0.14), where its keys-only loss equals or exceeds MEMIT's (6.56 / 6.55, 8.76 / 8.09), so on Llama-3.2-3B the extra deployed loss is mostly the lower fp margin; yet, consistent with a lower margin letting an equal nat loss flip more edits, the efficacy it loses to quantization net of its fp baseline is +0.080 [+0.030, +0.135] and +0.080 [+0.030, +0.130] more than MEMIT on the two Llamas, −0.005 [−0.090, +0.075] on GPT-2-XL and +0.020 [−0.050, +0.090] on GPT-J (its ΔES against MEMIT

is -0.085 $[-0.140, -0.035]$ on both Llamas, the same interval on each). The variant projects K onto the lowest half of C's spectrum rather than onto the null space of the preserved keys of the published method, so the comparison is between MEMIT and that variant.

Table 38: Alignment ridge against MEMIT, weight channel (only the edited matrix quantized), margin change in nats relative to each editor's own fp margin (negative = margin lost), paired per probe; "removed" is the fraction of MEMIT's weight-channel loss the ridge takes away; "keys" is each editor's keys-only change and ΔES the ridge's deployed efficacy against MEMIT's. The cells shown are the six int3/int4 cells where MEMIT's weight channel exceeds 0.6 nats, the two NF4 cells above 0.3, and the two Llama $n = 1000$/int3 cells; the remaining 28 int3/int4 cells (MEMIT's weight channel below 0.30 nats) are in the supplementary material.

model	quant	n	cos MEMIT→ridge	weights MEMIT	weights ridge	ridge − MEMIT	removed	keys MEMIT	keys ridge	ΔES
GPT-2-XL	NF4	1000	0.57→0.68	-0.62	-0.05	$+0.58$ [$+0.43, +0.72$]	92%	-0.53	-0.58	$+0.040$ [$+0.010, +0.075$]
GPT-2-XL	int4	1000	0.57→0.68	-0.81	-0.04	$+0.77$ [$+0.56, +0.99$]	95%	-1.27	-1.12	$+0.060$ [$+0.020, +0.105$]
GPT-2-XL	int3	1000	0.57→0.68	-2.56	-0.45	$+2.10$ [$+1.77, +2.45$]	82%	-5.05	-5.02	$+0.145$ [$+0.085, +0.210$]
GPT-J-6B	NF4	1000	0.58→0.76	-0.31	-0.07	$+0.24$ [$+0.13, +0.36$]	79%	-0.74	-0.51	$+0.015$ [$-0.015, +0.045$]
GPT-J-6B	int4	1000	0.58→0.76	-0.62	-0.17	$+0.45$ [$+0.27, +0.63$]	73%	-0.61	-0.28	$+0.020$ [$+0.000, +0.045$]
GPT-J-6B	int3	1000	0.58→0.76	-2.41	-0.58	$+1.83$ [$+1.50, +2.18$]	76%	-5.48	-5.03	$+0.100$ [$+0.050, +0.150$]
Qwen2.5-3B	int4	100	0.21→0.88	-0.61	-0.06	$+0.56$ [$+0.29, +0.84$]	90%	-4.34	-4.26	$+0.020$ [$+0.000, +0.050$]
Qwen2.5-3B	int3	100	0.21→0.88	-1.81	-0.25	$+1.55$ [$+1.01, +2.11$]	86%	-15.28	-14.85	-0.020 [$-0.050, +0.000$]
Llama-3.2-3B	int3	1000	0.66→0.80	-0.30	-0.19	$+0.11$ [$-0.07, +0.30$]	37%	-6.55	-6.01	$+0.025$ [$-0.015, +0.070$]
Llama-3.1-8B	int3	1000	0.66→0.81	-0.14	-0.08	$+0.06$ [$-0.09, +0.22$]	45%	-8.09	-7.58	$+0.025$ [$-0.015, +0.070$]

O KAE AND THE NEIGHBOURHOOD

Table 39 gives KAE's neighbourhood score and neighbourhood margin against MEMIT's in the full channel, paired per probe, for every int3 and int4 cell (the int8 and NF4 cells, and the analysis script that emits all 68, are in the supplementary material; the 78 cells of Appendix L add the ten GPT-2 and GPT-2-medium cells). Over all 68 cells the ΔNS interval excludes zero on the negative side in three, about the $68 \times 0.025 = 1.7$ that chance alone would give if the cells were independent ($P(\geq 3) = 0.24$; they are not, since they share probe facts across quantizers and batch sizes), all at int3 and $n = 1$ (Llama-3.2-3B, GPT-J-6B, Qwen2.5-3B), and in none at $n \geq 100$ in this run (the separate run of Table 20 has one, Llama-3.1-8B int3, $n = 100$), a binary score too coarse to show the cost the margin shows; the neighbourhood margin falls significantly in 14 of the 15 CounterFact single-edit cells below int8 (five models, int3/int4/NF4; not in Qwen2.5-3B's NF4 cell; 13 of the 15 after the correction of Appendix L, which drops Llama-3.1-8B's int4 cell, $p = 0.039$), by 0.02–0.68 nats, against rewrite-margin gains in the same run (the main grid; not tabulated) of 0.11–3.40 nats on the four main models (the two GPT-J gains at int4 and NF4 not significant; Table 40's separate run gives the same gains to within 0.02), so the trade is favourable but not free. On ZsRE, whose locality probe is an unrelated question with its own answer, the largest $|\Delta\mathrm{NS}|$ is 0.03 and the one significant neighbourhood-margin change is a gain ($+0.147$, GPT-2-XL, int3, $n = 100$). On CounterFact KAE is not free in locality; on GPT-2-XL its neighbourhood score does not move, though its neighbourhood margin does (-0.114 at int3).

Neighbourhood damage. MEMIT at int3, full channel, paired against each probe's own fp margin (rewrite margin for the edited fact, neighbourhood margin for its neighbours), in nats and as a fraction of the fp value (edited fact / neighbours), for $n = 1$, 100, 1000: GPT-2-XL -6.49 / -0.35 (50.5% / 14.2%), -6.44 / -0.41 (49.0% / 16.2%), -6.04 / -0.05 (83.3% / 6.1%); Llama-3.2-3B -6.75 / -1.01 (49.2% / 28.8%), -7.00 / -0.97 (50.8% / 32.1%), -6.81 / -0.86 (59.2% / 40.8%); GPT-J-6B -4.70 / -0.51 (31.5% / 15.8%), -4.56 / -0.55 (30.4% / 18.2%), -6.47 / -0.05 (71.6% / 8.7%); Llama-3.1-8B -8.14 / -1.50 (55.0% / 32.4%), -8.02 / -1.49 (53.8% / 37.6%), -8.29 / -1.14 (62.1% / 43.0%). The ratio of the two relative losses runs from 1.4 to 13.6 (GPT-2-XL at $n = 1000$; 8.2 for GPT-J there, where the overload has already moved the neighbour margins at fp), with median 1.7 over the twelve cells (1.8 in terms of the binary NS and ES scores).

Table 39: KAE minus MEMIT, full channel, paired 95% CI over the probes of each cell; NS = neighbourhood score, margin in nats; bold = CI excludes zero; a cell without an interval is one where every paired difference is zero.

model	quant	n	NS (MEMIT)	ΔNS	Δ neighbour margin	ΔES	ΔPS
GPT-2-XL	int4	1	0.825	−0.005 [−0.015, +0.000]	**−0.022** [−0.042, −0.004]	+0.000	+0.015 [+0.000, +0.035]
GPT-2-XL	int3	1	0.795	−0.015 [−0.035, +0.000]	**−0.114** [−0.162, −0.074]	**+0.065** [+0.035, +0.100]	**+0.145** [+0.100, +0.195]
GPT-2-XL	int4	100	0.800	−0.010 [−0.030, +0.000]	+0.024 [−0.021, +0.063]	+0.000	+0.000
GPT-2-XL	int3	100	0.780	−0.010 [−0.030, +0.000]	−0.014 [−0.101, +0.070]	**+0.060** [+0.020, +0.110]	**+0.150** [+0.080, +0.220]
GPT-2-XL	int4	1000	0.650	−0.005 [−0.045, +0.030]	+0.078 [−0.034, +0.190]	**+0.040** [+0.005, +0.075]	+0.010 [−0.040, +0.060]
GPT-2-XL	int3	1000	0.625	+0.000 [−0.055, +0.055]	+0.022 [−0.150, +0.196]	**+0.165** [+0.100, +0.230]	**+0.080** [+0.030, +0.130]
Llama-3.2-3B	int4	1	0.840	−0.015 [−0.035, +0.000]	**−0.121** [−0.178, −0.074]	+0.000	+0.010 [+0.000, +0.025]
Llama-3.2-3B	int3	1	0.815	**−0.025** [−0.050, −0.005]	**−0.292** [−0.391, −0.204]	+0.015 [−0.005, +0.040]	**+0.085** [+0.045, +0.130]
Llama-3.2-3B	int4	100	0.810	−0.020 [−0.060, +0.020]	**−0.162** [−0.280, −0.055]	+0.000	**+0.040** [+0.010, +0.080]
Llama-3.2-3B	int3	100	0.790	−0.040 [−0.090, +0.000]	**−0.399** [−0.628, −0.187]	+0.030 [−0.010, +0.080]	**+0.070** [+0.010, +0.140]
Llama-3.2-3B	int4	1000	0.715	−0.025 [−0.070, +0.020]	**−0.199** [−0.350, −0.048]	+0.010 [−0.010, +0.030]	+0.005 [−0.020, +0.030]
Llama-3.2-3B	int3	1000	0.665	+0.000 [−0.060, +0.060]	**−0.287** [−0.536, −0.038]	**+0.065** [+0.020, +0.110]	**+0.085** [+0.030, +0.140]
GPT-J-6B	int4	1	0.845	−0.015 [−0.035, +0.000]	**−0.043** [−0.074, −0.022]	−0.005 [−0.015, +0.000]	+0.015 [+0.000, +0.035]
GPT-J-6B	int3	1	0.805	**−0.040** [−0.070, −0.015]	**−0.185** [−0.232, −0.145]	+0.010 [−0.010, +0.030]	**+0.035** [+0.010, +0.060]
GPT-J-6B	int4	100	0.820	+0.000	−0.020 [−0.057, +0.016]	+0.000	+0.000
GPT-J-6B	int3	100	0.780	−0.020 [−0.060, +0.020]	**−0.198** [−0.319, −0.086]	+0.000	**+0.040** [+0.010, +0.080]
GPT-J-6B	int4	1000	0.595	−0.025 [−0.065, +0.015]	−0.042 [−0.148, +0.065]	+0.010 [−0.015, +0.035]	+0.000 [−0.035, +0.035]
GPT-J-6B	int3	1000	0.575	+0.000 [−0.050, +0.050]	−0.109 [−0.290, +0.068]	**+0.105** [+0.050, +0.165]	+0.025 [−0.030, +0.080]
Llama-3.1-8B	int4	1	0.915	+0.005 [−0.010, +0.025]	**−0.061** [−0.126, −0.003]	+0.000	+0.010 [+0.000, +0.025]
Llama-3.1-8B	int3	1	0.820	−0.015 [−0.035, +0.000]	**−0.216** [−0.302, −0.125]	**+0.065** [+0.030, +0.105]	**+0.095** [+0.050, +0.140]
Llama-3.1-8B	int4	100	0.840	+0.000 [−0.030, +0.030]	−0.101 [−0.232, +0.012]	+0.000	+0.010 [+0.000, +0.030]
Llama-3.1-8B	int3	100	0.790	−0.050 [−0.100, +0.000]	**−0.497** [−0.724, −0.284]	**+0.070** [+0.020, +0.130]	**+0.110** [+0.050, +0.180]
Llama-3.1-8B	int4	1000	0.775	+0.005 [−0.025, +0.035]	+0.041 [−0.108, +0.190]	+0.000 [−0.015, +0.015]	+0.005 [−0.010, +0.025]
Llama-3.1-8B	int3	1000	0.670	+0.015 [−0.045, +0.075]	−0.110 [−0.344, +0.132]	**+0.095** [+0.045, +0.145]	**+0.135** [+0.075, +0.195]
Qwen2.5-3B	int4	1	0.845	−0.005 [−0.025, +0.015]	**−0.106** [−0.150, −0.060]	**+0.045** [+0.015, +0.080]	+0.005 [−0.025, +0.035]
Qwen2.5-3B	int3	1	0.520	**−0.105** [−0.155, −0.060]	**−0.679** [−0.831, −0.519]	**+0.090** [+0.040, +0.145]	**+0.120** [+0.070, +0.170]
Qwen2.5-3B	int4	100	0.820	−0.040 [−0.100, +0.010]	**−0.184** [−0.307, −0.065]	+0.030 [−0.020, +0.080]	+0.040 [−0.030, +0.110]
Qwen2.5-3B	int3	100	0.480	−0.030 [−0.100, +0.040]	+0.012 [−0.186, +0.211]	−0.040 [−0.120, +0.040]	+0.020 [−0.050, +0.090]
GPT-2-XL/ZsRE	int4	1	0.850	+0.000	−0.001 [−0.004, +0.002]	+0.000	+0.010 [−0.010, +0.030]
GPT-2-XL/ZsRE	int3	1	0.820	+0.000	−0.008 [−0.016, +0.001]	**+0.120** [+0.075, +0.165]	**+0.185** [+0.135, +0.240]
GPT-2-XL/ZsRE	int4	100	0.860	−0.010 [−0.030, +0.000]	+0.009 [−0.008, +0.026]	+0.000	+0.010 [+0.000, +0.030]
GPT-2-XL/ZsRE	int3	100	0.820	+0.030 [+0.000, +0.070]	**+0.147** [+0.080, +0.216]	**+0.110** [+0.050, +0.170]	**+0.110** [+0.050, +0.170]
Llama-3.2-3B/ZsRE	int4	1	0.915	+0.000	+0.000 [−0.014, +0.014]	+0.000	+0.000 [−0.015, +0.015]
Llama-3.2-3B/ZsRE	int3	1	0.845	−0.010 [−0.025, +0.000]	−0.022 [−0.058, +0.014]	**+0.090** [+0.050, +0.135]	**+0.075** [+0.030, +0.120]
Llama-3.2-3B/ZsRE	int4	100	0.950	+0.000	−0.028 [−0.122, +0.066]	+0.010 [+0.000, +0.030]	−0.010 [−0.030, +0.000]
Llama-3.2-3B/ZsRE	int3	100	0.810	+0.030 [−0.020, +0.080]	+0.134 [−0.124, +0.393]	+0.030 [−0.030, +0.090]	+0.050 [+0.000, +0.100]

P PRECISION TRANSFER

KAE is solved under one quantizer q_{solve} and deployed under another q_{deploy} (all 25 pairs of fp, int8, NF4, int4 and int3), full channel, on 200 probes per cell on the four main models and 100 on GPT-2 (45,000 rows); $q_{\text{solve}} = \text{fp}$ is MEMIT and every comparison is paired on the same probes. Table 40 gives the diagonal's gain over MEMIT and the mean cost of a mismatched solve. The diagonal is where each solve was fitted, so that it wins is expected; the content is the size and ordering of the off-diagonal cost.

Table 40: Top: rewrite-margin gain (nats) of the matched-precision KAE over MEMIT at each deployment precision; bold = 95% CI excludes zero, ns = not significant. Bottom: mean cost versus MEMIT of an edit solved at the row precision, averaged over the four deployments that do not match it (fp and the other three quantizers). A separate run from Tables 18 and 19, so the same nominal cell can differ (by up to 0.11 nats).

deploy	GPT-2 (124M)		GPT-2-XL		Llama-3.2-3B		GPT-J-6B		Llama-3.1-8B	
	n=1	n=150	n=1	n=1000	n=1	n=1000	n=1	n=1000	n=1	n=1000
int3	**+4.11**	**+2.46**	**+3.40**	**+1.94**	**+2.23**	**+1.11**	**+1.16**	**+1.30**	**+2.80**	**+2.15**
int4	**+0.65**	−0.11 ns	**+0.58**	**+0.71**	**+0.53**	**+0.30**	+0.14 ns	**+0.33**	**+0.65**	+0.21 ns
NF4	**+0.31**	**−0.45**	**+0.36**	**+0.41**	**+0.44**	+0.16 ns	+0.11 ns	**+0.66**	**+0.55**	**+0.32**
int8	−0.00 ns	−0.03 ns	**+0.01**	−0.01 ns	+0.00 ns	+0.01 ns	−0.00 ns	−0.00 ns	+0.01 ns	−0.02 ns
solve										
int8	−0.01	+0.01	+0.00	+0.02	+0.01	−0.02	−0.00	−0.03	−0.00	−0.05
NF4	−0.82	−1.62	−0.31	−0.80	−0.04	−0.51	−0.14	−0.45	−0.23	−0.65
int4	−1.18	−2.09	−0.46	−0.78	−0.24	−0.91	−0.24	−0.84	−0.77	−1.28
int3	−6.87	−7.16	−2.79	−3.49	−3.78	−4.25	−1.86	−5.16	−6.81	−7.04

The matched solve beats MEMIT, or ties, at every deployment precision on all four main models at both batch sizes (the gain is significant in 20 of the 24 cells below int8 and never significantly negative); at int8 the two coincide (the one int8 gain whose interval excludes zero, GPT-2-XL at $n = 1$, is $+0.005$ nats), the key having moved by only 1.5–4.2% of its norm. The bottom panel

averages the cost of a mismatched solve over the four deployments (fp, and the three quantizers other than the solve's); pair by pair, a mismatched NF4, int4 or int3 solve is significantly worse than MEMIT in 86 of the 96 pairs on the four main models and significantly better in three, each a finer solve deployed at int3 on a Llama at $n = 1$ ($+0.17$ to $+0.24$ nats). A mismatched int3 solve is worse than plain MEMIT by 1.86–7.16 nats on average, several times what 4-bit quantization costs MEMIT itself. The gain shrinks with batch size at int3 ($+3.40 \to +1.94$, $+2.23 \to +1.11$, $+2.80 \to +2.15$), consistent with cross-talk between the constraints of a batch solve, and not at int4 and NF4 on GPT-2-XL and GPT-J, where it grows; GPT-J at int3 is the counterexample ($+1.16 \to +1.30$). On GPT-2 (124M), the smallest model, the NF4 batch cell goes the other way (-0.45 at $n = 150$); no larger model reproduces it. The cost of a mismatch is ordered by solve precision at $n = 1$ on every model, at $n = 150$ on GPT-2 and at $n = 1000$ in all but one pair (GPT-2-XL, NF4 -0.80 against int4 -0.78), in line with Proposition 2: the coarser the solve, the larger the δk it was fitted to, and the further the solved edit sits from the fp optimum.

Q AWQ WITH A FIXED SCALE

Table 41: AWQ against round-to-nearest at matched bit width, MEMIT, full channel, damage in nats, paired per probe, all 24 cells (bold = CI excludes zero; "reduction" is the paired difference over the RTN damage). Bottom: the step-map ratio $\Delta_{\mathrm{AWQ}}/\Delta_{\mathrm{RTN}}$ (averaged over output rows) at the median input channel of the edited matrix and averaged over the top-1% input channels by mean $|k_j^*|$ over the probes, and its Pearson correlation with $\log|k_j^*|$ over input channels; the bit width cancels out of the ratio where AWQ picks the same exponent at both widths ($\alpha = 0.35$, 0.25 and 0.40 on GPT-2-XL, Llama-3.2-3B and GPT-J-6B); on Llama-3.1-8B it picks 0.30 at int3 and 0.25 at int4, and its row is one of the two (which one was not recorded). A separate run from Table 11.

model	n	bits	damage RTN	damage AWQ	RTN − AWQ	reduction
GPT-2-XL	1	3	6.487	5.255	**+1.232** [+0.955, +1.526]	19.0%
GPT-2-XL	1	4	1.136	0.886	**+0.250** [+0.106, +0.397]	22.0%
GPT-2-XL	100	3	6.433	5.329	**+1.104** [+0.665, +1.569]	17.2%
GPT-2-XL	100	4	1.128	0.838	**+0.290** [+0.061, +0.532]	25.7%
GPT-2-XL	1000	3	6.045	5.313	**+0.732** [+0.421, +1.040]	12.1%
GPT-2-XL	1000	4	1.908	1.729	+0.179 [−0.104, +0.452]	9.4%
Llama-3.2-3B	1	3	6.753	6.040	**+0.713** [+0.322, +1.107]	10.6%
Llama-3.2-3B	1	4	1.561	1.363	+0.198 [−0.028, +0.428]	12.7%
Llama-3.2-3B	100	3	6.994	6.365	**+0.628** [+0.049, +1.225]	9.0%
Llama-3.2-3B	100	4	1.489	1.249	+0.240 [−0.061, +0.542]	16.1%
Llama-3.2-3B	1000	3	6.808	6.527	+0.281 [−0.231, +0.790]	4.1%
Llama-3.2-3B	1000	4	1.782	1.392	**+0.391** [+0.057, +0.719]	21.9%
Llama-3.1-8B	1	3	8.139	7.680	**+0.459** [+0.056, +0.862]	5.6%
Llama-3.1-8B	1	4	1.893	1.340	**+0.553** [+0.267, +0.843]	29.2%
Llama-3.1-8B	100	3	8.017	7.834	+0.183 [−0.425, +0.806]	2.3%
Llama-3.1-8B	100	4	1.895	1.354	**+0.541** [+0.131, +0.962]	28.5%
Llama-3.1-8B	1000	3	8.288	7.420	**+0.868** [+0.388, +1.347]	10.5%
Llama-3.1-8B	1000	4	1.745	1.248	**+0.498** [+0.154, +0.832]	28.5%
GPT-J-6B	1	3	4.698	3.432	**+1.266** [+0.969, +1.575]	26.9%
GPT-J-6B	1	4	0.662	0.672	−0.010 [−0.128, +0.104]	−1.5%
GPT-J-6B	100	3	4.557	3.298	**+1.260** [+0.848, +1.704]	27.6%
GPT-J-6B	100	4	0.675	0.697	−0.022 [−0.175, +0.130]	−3.2%
GPT-J-6B	1000	3	6.473	5.596	**+0.877** [+0.539, +1.215]	13.5%
GPT-J-6B	1000	4	1.121	1.290	−0.170 [−0.456, +0.108]	−15.1%

step-map selectivity	median channel	top-1% $\lvert k^*\rvert$	corr
GPT-2-XL	0.990	0.907	−0.522
Llama-3.2-3B	1.015	0.887	−0.550
Llama-3.1-8B	1.020	0.864	−0.475
GPT-J-6B	0.990	0.887	−0.578

Table 41 gives every cell; the scale exponent chosen on the unedited down-projection is $\alpha = 0.35$ (GPT-2-XL), 0.25 (Llama-3.2-3B), 0.40 (GPT-J-6B) and 0.30 and 0.25 (Llama-3.1-8B at 3 and 4

bits), so none of the four is round-to-nearest ($\alpha = 0$). Over all 24 MEMIT cells (four models, three batch sizes, two bit widths) AWQ reduces the damage in 21 and the paired CI excludes zero in 16; its calibration ratios sit with those of its round-to-nearest partners (Figure 3: 0.90–0.99 at $n \leq 100$, inside the pre-logged band, and 0.56–0.78 at $n = 1000$, below it with them); the bit width at which it pays more, in relative reduction, is model-dependent and the same at every batch size on three models (int3 on GPT-J, int4 on the two Llamas; on GPT-2-XL int4 at $n \leq 100$ and int3 at $n = 1000$), which we can describe and cannot explain. The two channels need not move together under AWQ: at int3, $n = 1$ on Llama-3.2-3B it makes the weight channel significantly *worse* (-0.059 [$-0.100, -0.020$]) while making the keys-only channel better ($+0.758$ [$+0.354, +1.157$]), although AWQ lowers the edit's own readout error in that cell ($0.0640 \rightarrow 0.0635$, paired change [$-0.0006, -0.0002$]), so the extra weight-channel loss is not in the edit's readout at its key and we have not located it; at $n = 1000$ on GPT-J, where the weight channel carries 26% of the damage by the even split, AWQ helps both ($+0.905$ and $+0.548$; these channel contrasts are at int3, paired within one run). Under the AWQ backbone at int3, $\|\delta k\|/\|k^*\|$ falls from 0.593 to 0.548 (GPT-2-XL) and 0.832 to 0.748 (Llama-3.2-3B). AWQ and KAE are substitutes: KAE's advantage over MEMIT (each editor's damage against its own fp margin, MEMIT's minus KAE's) shrinks under AWQ in 22 of the 24 cells, with the interval above zero in 21 and below zero in none (the two exceptions are GPT-2-XL at $n = 1000$), e.g. at int3, $n = 1$, from $+6.331$ to $+5.485$ nats on GPT-2-XL (by $+0.847$ [$+0.623, +1.088$]) and from $+6.415$ to $+4.547$ on Llama-3.2-3B (by $+1.868$ [$+1.490, +2.266$]); the advantage shrinks from both sides, MEMIT's damage falling under AWQ in 21 of the 24 cells and KAE's own damage rising in 17; KAE's own fp margin differs between its RTN and AWQ solves, so these are not the differences of the deployed margins below. AWQ does not replace KAE: at int3, $n = 1$ MEMIT under AWQ delivers $+7.587$ nats on GPT-2-XL where KAE under AWQ delivers $+10.288$ ($+9.758$ under round-to-nearest), and $+7.685$ against $+8.982$ ($+9.198$) on Llama-3.2-3B.

Table 42 gives AWQ's own four-channel decomposition: on the eight AWQ single-edit cells (each with at least 0.67 nats of damage) the conditioning share is 97.9–101.0%, so the attribution holds for a channel-rescaled grid as well (one whose step differs from rounding's by 9–14% on average over the salient channels); at $n = 1000$, as under round-to-nearest, the weight channel takes a visible share where the batch overloads the matrix (a weight share of 44.9% and 30.7% by the even split on GPT-2-XL at AWQ-4 and AWQ-3, 29.7% and 19.4% on GPT-J-6B).

Table 42: AWQ with a fixed scale, MEMIT: mean damage (nats; positive = loss, as in Table 41) in the weights-only (W), keys-only (K) and full (F) channels, the conditioning share K/F and the even split with its 95% CI.

model	quant	n	W	K	F	K/F	even split
GPT-2-XL	AWQ-3	1	0.01	5.27	5.26	100.3%	100.1% [99.7, 100.5]
GPT-2-XL	AWQ-4	1	0.01	0.87	0.89	98.5%	98.8% [97.6, 99.9]
GPT-2-XL	AWQ-3	100	0.02	5.26	5.33	98.8%	99.1% [98.4, 99.9]
GPT-2-XL	AWQ-4	100	−0.00	0.84	0.84	100.3%	100.3% [98.4, 102.2]
GPT-2-XL	AWQ-3	1000	2.30	4.35	5.31	81.9%	69.3% [65.9, 72.7]
GPT-2-XL	AWQ-4	1000	0.83	1.00	1.73	58.0%	55.1% [45.7, 63.5]
Llama-3.2-3B	AWQ-3	1	0.10	5.97	6.04	98.9%	98.6% [97.9, 99.2]
Llama-3.2-3B	AWQ-4	1	0.01	1.35	1.36	99.1%	99.2% [97.8, 100.6]
Llama-3.2-3B	AWQ-3	100	0.05	6.34	6.37	99.6%	99.4% [98.3, 100.6]
Llama-3.2-3B	AWQ-4	100	0.01	1.23	1.25	98.8%	98.9% [96.4, 101.4]
Llama-3.2-3B	AWQ-3	1000	0.20	6.27	6.53	96.0%	96.5% [94.7, 98.3]
Llama-3.2-3B	AWQ-4	1000	0.11	1.31	1.39	94.2%	93.1% [89.0, 96.7]
GPT-J-6B	AWQ-3	1	0.04	3.36	3.43	97.9%	98.3% [97.5, 99.1]
GPT-J-6B	AWQ-4	1	0.01	0.67	0.67	99.5%	99.3% [97.6, 101.0]
GPT-J-6B	AWQ-3	100	0.04	3.26	3.30	98.8%	98.8% [97.5, 100.0]
GPT-J-6B	AWQ-4	100	0.01	0.69	0.70	98.4%	98.4% [95.9, 100.9]
GPT-J-6B	AWQ-3	1000	1.51	4.94	5.60	88.2%	80.6% [77.5, 83.7]
GPT-J-6B	AWQ-4	1000	0.40	0.93	1.29	71.9%	70.3% [61.5, 78.4]
Llama-3.1-8B	AWQ-3	1	0.09	7.61	7.68	99.1%	98.9% [98.3, 99.6]
Llama-3.1-8B	AWQ-4	1	0.01	1.35	1.34	101.0%	100.3% [98.6, 102.0]
Llama-3.1-8B	AWQ-3	100	0.01	7.89	7.83	100.7%	100.3% [99.2, 101.4]
Llama-3.1-8B	AWQ-4	100	0.01	1.32	1.35	97.4%	98.3% [95.4, 101.1]
Llama-3.1-8B	AWQ-3	1000	0.16	7.47	7.42	100.6%	99.2% [97.7, 100.7]
Llama-3.1-8B	AWQ-4	1000	0.00	1.20	1.25	96.4%	98.1% [94.2, 101.5]

R FT-L

Table 43: FT-L, the control: $n = 1$, int3, 200 probes. Rounding gives a dense, less aligned, gradient-produced update a 2.9–3.7× larger relative readout error ("readout") than MEMIT's, which changes its margin by $+0.02$ (n.s.) on GPT-2-XL and -0.28 on Llama-3.2-3B; the closed form predicts its readout error to 1% ("meas./pred." is the mean per-edit ratio, as in Table 5; FT-L's ratio of cell means, the average §4.4 quotes, is 0.993 and 0.990; "rank" is the effective rank of Appendix A; $|\cos|$ is $|\cos(u, k^*)|$; weights-only and keys-only are margin changes in nats, negative = margin lost). MEMIT's rows are the main grid's, on the same 200 probes.

model	editor	$\lvert\cos\rvert$	rank	readout	weights-only	keys-only	K/F	meas./pred.
GPT-2-XL	MEMIT	0.718	1	0.082	+0.001 [−0.024, +0.026]	−6.520	100.5%	0.983
GPT-2-XL	FT-L	0.399	143	0.236	+0.019 [−0.007, +0.048]	−5.195	100.6%	0.993
Llama-3.2-3B	MEMIT	0.745	1	0.064	−0.045 [−0.090, −0.001]	−6.724	99.6%	0.974
Llama-3.2-3B	FT-L	0.380	400	0.235	−0.275 [−0.332, −0.221]	−8.453	97.0%	0.991

FT-L's single edits succeed unquantized (fp margin 8.89 and 13.21 nats, efficacy 0.995 and 1.000 on GPT-2-XL and Llama-3.2-3B) and lose 5.16 [4.71, 5.63] and 8.71 [8.09, 9.33] nats in the full channel at int3 (deployed efficacy 0.840 and 0.845). At int4 the same cells give FT-L a weight channel of -0.003 [-0.014, $+0.008$] on GPT-2-XL (conditioning share 100.4%) and -0.058 [-0.080, -0.036] on Llama-3.2-3B (97.6%), against MEMIT's -0.006 and -0.008, and FT-L's calibration ratios (mean per-edit, as in Table 43) are 0.993 and 0.988 (MEMIT's in the grid 0.987 and 0.974); at int3 its median ESNR^2 is 19.1 and 20.5 against MEMIT's 145.8 and 242.3; and its relative update norm is 0.022 and 0.018 against 0.073 and 0.063.

Batch size 100. At $n = 100$ with this trust region (Appendix A) FT-L barely edits: fp rewrite margin -1.85 with ES 0.29 on GPT-2-XL and $+1.52$ with ES 0.64 on Llama-3.2-3B, against MEMIT's ES of 1.00 on both, reproducing the known result that fine-tuning baselines do not scale to batches (Meng et al., 2023). GPT-2-XL's $n = 100$ cells decompose a failed edit and are not read (quantization

raises its margin by 0.28 and 0.04 nats at int3 and int4, with conditioning shares of 100.1% and 91.7%). Llama-3.2-3B's are a partial edit and are the single strongest support for what the failed prediction P9 (Appendix Z) was reaching for: FT-L's weight channel is -0.440 $[-0.603, -0.282]$ at int3 against MEMIT's -0.040, a weight share (weights-only over full damage) of 12.5% against 0.6%, and -0.112 $[-0.165, -0.061]$ at int4 (4.9% against 1.1%). The conditioning share is still 100.2% and 95.0% in these cells (int3, int4): the channels interact, and the keys-only loss alone reproduces the full loss. P9 asked for a conditioning share below 90% and does not get it even here.

S MULTI-LAYER EDITING

Published MEMIT spreads a fact's residual over a band of consecutive layers, the l-th layer of the band ($l = 1, \ldots, L$, bottom to top) receiving $1/(L - l + 1)$ of what is still missing after the earlier layers are edited. Table 44 runs the four-channel ablation at $L \in \{1, \ldots, 5\}$ on GPT-2 (int3, 40 probes, $n = 1$) under a matched control that holds the same five matrices in fp at every L, so that a wider band cannot lower the keys-only channel for the trivial reason that fewer matrices are quantized. We had logged in advance that the weight channel would get safer, the keys-only channel would be unchanged and the conditioning share would rise; on GPT-2 the second and third were wrong and the first is not supported (the weight channel moves the other way, -0.26 $[-0.52, +0.01]$ paired from $L = 1$ to $L = 5$), and on GPT-2-XL below the second and third were wrong and the first is not supported (the band's weight channel moves from $+0.00$ to -0.05 nats, every interval covering zero; on Llama-3.2-3B two of the three hold at $L = 5$ and one, the rising share, at $L = 3$). The keys-only channel improves by more than half and the total damage roughly halves while the fp margin rises; the conditioning share falls because the weight channel and the interaction do not shrink with them. On GPT-2-medium (band ending at layer 8, 30 probes here against 20 in the grid) the conditioning share falls $99.4\% \to 96.8\% \to 90.1\%$ at $L = 1, 3, 5$ and the total damage $-6.90 \to -3.95 \to -3.36$; no growth of the weight channel is detected there (-0.08, $+0.26$, $+0.13$, two intervals covering zero and the $L = 3$ one, $[+0.002, +0.617]$, above it, where quantizing the band helps), so "spreading makes the weight channel bite" is a GPT-2 observation we do not generalize. At int4 on GPT-2 the conditioning share falls similarly ($98.7\% \to 86.2\% \to 84.1\%$) but the total damage does not ($-1.36 \to -1.01 \to -1.20$), so the halving is claimed at int3 only.

Table 44: GPT-2, MEMIT, int3, $n = 1$, 40 probes, matched control (the down-projections of layers 1–5 held in fp in the keys-only channel at every L, and in the full channel all but the current band, so for $L < 5$ the full channel is not the deployed model: without the control $L = 1$ gives keys-only -8.29 and full -8.44, the grid's -8.27 and -8.44 on these probes; the sweep re-solves the edit, fp margin 11.76 against the grid's 11.69); margin changes in nats relative to the fp margin (negative = margin lost).

L	band	fp margin	weight channel (band)	keys-only	full	K/F
1	5	11.76	-0.07 $[-0.20, +0.07]$	-7.84	-7.93	98.9%
2	4–5	13.41	-0.14 $[-0.31, +0.01]$	-6.57	-6.80	96.7%
3	3–5	13.35	-0.17 $[-0.37, +0.02]$	-5.57	-5.84	95.5%
4	2–5	13.20	-0.35 $[-0.56, -0.13]$	-4.20	-4.57	92.0%
5	1–5	13.01	-0.32 $[-0.60, -0.05]$	-3.57	-4.12	86.7%

At GPU scale the damage stays in the conditioning. Table 45 repeats the ablation on GPT-2-XL and Llama-3.2-3B at int3 ($n = 1$, 100 probes, the same matched control), spreading over MEMIT's published band on GPT-2-XL (layers 13–17) and, on Llama-3.2-3B, for which no band is published, a five-layer band ending at the edited layer (0–4). The picture from GPT-2 carries over only in part: the conditioning share drifts down on GPT-2-XL ($100.5 \to 96.8 \to 94.0\%$), less than on GPT-2, and rises on Llama-3.2-3B ($98.7 \to 102.2 \to 117.7\%$), where the weight channel of the whole band is $+0.50$ $[+0.22, +0.78]$ at $L = 5$, i.e. quantizing the five edited matrices *helps*; in the other five cells it covers zero. What reproduces at both scales is the benefit: the total damage roughly halves on GPT-2 and more than halves here ($-6.33 \to -2.70$ and $-6.75 \to -2.16$ nats; on Llama-3.2-3B the fp margin falls by 2.5 nats with it, so the full-channel margin, with the four other band matrices unquantized at $L = 1$, rises only from 7.10 to 9.19), and the mean per-entry displacement $\bar{\delta}/\Delta$ of the band falls from 0.033 to 0.014 and 0.041 to 0.027, but that is not where the saving is: the weight channel does not shrink ($+0.00 \to -0.05$ on GPT-2-XL), and the loss a spread edit avoids is in the

keys-only channel ($-6.36 \to -2.54$ and $-6.66 \to -2.54$). Five layers are a tenth of GPT-2-XL (48 layers), a sixth of Llama-3.2-3B (28) and two fifths of GPT-2 (12), but also a fifth of GPT-2-medium (24), whose conditioning share does fall to 90.1%, so depth fraction does not explain Llama-3.2-3B's rise, although within the GPT-2 family the fall grows with it (6.5, 9.3 and 12.2 points on GPT-2-XL, GPT-2-medium and GPT-2); we report it rather than resolve it. The matched-control check therefore covers int3, $n = 1$, two main models, without QAE; released bands at int3 and int4 up to $n = 1000$, with QAE, are in Appendix T.

Table 45: MEMIT spread over L layers at GPU scale, int3, $n = 1$, 100 probes, matched control (the five matrices of the $L = 5$ band held in fp in the keys-only channel at every L, and in the full channel all but the current band); margin changes in nats relative to the fp margin (negative = margin lost). $\bar{\delta}/\Delta$ is the mean per-entry displacement of the band in bin widths.

model	L	band	fp margin	$\bar{\delta}/\Delta$	weight channel	keys-only	full	K/F
GPT-2-XL	1	17	13.22	0.033	$+0.00$ $[-0.03, +0.03]$	-6.36	-6.33	100.5%
GPT-2-XL	3	15–17	14.67	0.020	-0.06 $[-0.13, +0.01]$	-3.47	-3.59	96.8%
GPT-2-XL	5	13–17	14.35	0.014	-0.05 $[-0.13, +0.03]$	-2.54	-2.70	94.0%
Llama-3.2-3B	1	4	13.85	0.041	-0.02 $[-0.08, +0.05]$	-6.66	-6.75	98.7%
Llama-3.2-3B	3	2–4	12.66	0.031	-0.08 $[-0.16, +0.00]$	-3.15	-3.08	102.2%
Llama-3.2-3B	5	0–4	11.35	0.027	$+0.50$ $[+0.22, +0.78]$	-2.54	-2.16	117.7%

T MEMIT AND ALPHAEDIT WITH RELEASED SETTINGS

The main grid edits one matrix with the unweighted second moment C, so a thousand facts land on a single layer at full strength: $\|\Delta W\|_F/\|W\|_F$ reaches 7.5–11.6 and the unquantized model's perplexity 5.4–6.5× its unedited value on GPT-2-XL and GPT-J (on the two Llamas the update is 2.2 and 1.7 times the norm and the perplexity 1.27 and 1.13×; Appendix V). Published MEMIT (Meng et al., 2023) solves each layer of a band with $\lambda_w C + KK^\top$ and spreads the residual over the band, and published AlphaEdit (Fang et al., 2025) replaces the covariance term by a projection onto the null space of the preserved keys. We run both with their released regularization and bands (`code/published_memit.py`; the targets and C are ours, below): GPT-2-XL layers 13–17 with $\lambda_w = 2 \times 10^4$, GPT-J layers 3–8 and Llama-3.1-8B layers 4–8 with 1.5×10^4 (MEMIT's and AlphaEdit's released settings; AlphaEdit's release covers Llama-3-8B, whose band we use). The target is the residual stream at the top of the band, $z_i = h_{\text{top}} + \delta_i$, with h_{top} its unedited value and δ_i the value change the single-layer optimization of Appendix A finds at the top layer; each layer in turn reads its keys and the current residual and solves for $R/(L - l + 1)$, with l its position in the band of L layers. AlphaEdit's update for one batch with no earlier edits (so its protection of earlier edits in sequential editing is not exercised) is $\Delta W^\top = [PKK^\top + L_2 I]^{-1} PKR^\top$ with P the projection onto the eigenvectors of C with eigenvalue below 2×10^{-2} and $L_2 = 10$, its released values (applied to our C, the per-token mean of Appendix A, estimated on WikiText-103; the rank of P this gives was not recorded), computed exactly as $PK(L_2 I + K^\top PK)^{-1} R^\top$. QAE solves the same procedure inside the model quantized everywhere except the band, with δ re-optimized through it, and the band is quantized at deployment like any other layer. The four channels are as before with the band in the role of the edited matrix: weights-only quantizes the band, keys-only everything else.

The conditioning share with released settings. Tables 46 and 47 give every cell. Over the 18 MEMIT cells (three models, int3 and int4, $n \in \{1, 100, 1000\}$) the conditioning share is 88.5–99.4% and the even split 87.6–97.5%; at int3 the band's weight channel costs 0.04–0.39 nats against keys-only losses of 3.6–7.3 (conditioning share 91.5–96.5%). Over the 15 AlphaEdit cells the conditioning share is 87.9–98.6% (even split 86.4–98.7%), 92.4–96.1% at int3, where the weight channel costs 0.09–0.34 nats; every cell of either editor below 90% is GPT-J at int4, where the whole loss is 0.24–0.35 nats. Neither the unweighted covariance nor editing a single matrix is what puts the damage in the conditioning: with the released regularization and bands the edited matrices barely move under rounding and the moved keys and read-out carry the loss. Three disjoint $n = 100$ batches of published MEMIT per model agree (conditioning share 91.7–96.1% and even split 93.3–97.5% at int3 over the nine; Table 48). On GPT-2-XL the released λ_w leaves the single-layer target partly undelivered at full precision (fp efficacy 0.74–0.86, margin 5.5–7.3 nats over Tables 46 and 48,

against 0.98–1.00 and 11.3–14.6 nats on GPT-J and Llama-3.1-8B; its $n = 1$ and $n = 1000$ rows agree in fp margin only in the mean, Table 46). AlphaEdit, solved on the same band for the same targets, delivers them (fp efficacy 0.98–1.00, Table 47), so the target δ (our own value optimization, fixed prefixes and clamp $4\times$, where the release samples prefixes and clamps tighter) does not by itself explain the shortfall, which comes with MEMIT's $\lambda_w C$ term as we instantiate it, C from WikiText-103 rather than Wikipedia; this is a reimplementation that under-delivers the published method's efficacy on GPT-2-XL; the attribution is the same on all three models.

A thousand facts are no longer an overload. At $n = 1000$ the band update is 0.29–0.38 of the band's weights in Frobenius norm for MEMIT (0.28–0.58 for AlphaEdit), against 7.5–11.6 for the single-layer solve on GPT-2-XL and GPT-J (1.7 on Llama-3.1-8B, where its weight channel is under 2% of the int3 loss), and the unquantized edited model's perplexity stays within 2.4% of the unedited one for MEMIT (GPT-2-XL 21.07 → 21.25, GPT-J 11.92 → 12.04, Llama-3.1-8B 13.46 → 13.78; this run measures perplexity on 20,000 test tokens rather than the 99,840 of Table 53, so its unedited values differ, and each ratio is taken within its own run) and within 7.1% for AlphaEdit, against 5.4–6.5× on the main grid's GPT-2-XL and GPT-J. In this regime the weight channel stays at 0.34–0.39 nats at int3 for MEMIT (0.20–0.34 for AlphaEdit) and the int3 conditioning share at 91.5–93.9% (MEMIT; 88.5–93.9% with int4) and 92.4–94.8% (AlphaEdit): the $n = 1000$ exception of the main grid (even split 70.6% and 73.7%) does not come with the batch size. This contrast alone does not tie it to the overload, since the released setting changes the covariance weight, the number of edited layers and, on GPT-J and Llama-3.1-8B, the top edited layer (8 against 5) at once; the single-layer grid ties it at a fixed layer and weight, where the GPT-2-XL batches of 100 whose update exceeds the matrix's norm have a conditioning share of 85.2–95.2% at int3 against 98.7–99.8% in the other six (Appendix D).

QAE recovers less than on the single-layer grid. Solved through the same band inside the deployed model, QAE returns 60–96% of MEMIT's int3 damage (72–96% for single edits; GPT-2-XL 60–72%, GPT-J 73–87%, Llama-3.1-8B 78–96%, falling with n on every model; 68–98% over the nine batches of Table 48) and the same construction applied to AlphaEdit returns 72–95% of AlphaEdit's int3 damage (88–95% for single edits, 72–80% at $n = 1000$), and deployed efficacy rises wherever quantization had lowered it: at int3 QAE's is 0.800, 0.870 and 0.770 on GPT-2-XL at $n = 1$, 100 and 1000 (MEMIT's 0.620, 0.720 and 0.665) and 0.990–1.000 on GPT-J and Llama-3.1-8B (MEMIT's 0.855–1.000), and QAE-AlphaEdit's is 0.975–1.000 on all three models (AlphaEdit's 0.845–1.000). At int4, where MEMIT and AlphaEdit lose 0.2–1.5 nats, the recovered fractions are too loosely estimated to rank (intervals as wide as 7–413%). This is less than the 90–100% of the single-layer grid (on GPT-J and Llama-3.1-8B these runs find v^* with the batched pass of Appendix A), and we have not adapted QAE to the band: the covariance term is still the full-precision one and δ is re-optimized at the top layer only.

Table 46: Published MEMIT: covariance weight λ_w (2×10^4 GPT-2-XL, 1.5×10^4 GPT-J and Llama-3.1-8B) and the released bands (13–17, 3–8, 4–8). fp: unquantized rewrite margin; ES fp/dep.: efficacy unquantized / deployed. Margin changes in nats relative to the fp margin (negative = margin lost), paired 95% intervals for the weights-only channel (every keys-only and full-channel interval excludes zero); weights-only = the band quantized, keys-only = everything else. QAE recovered is the share of MEMIT's deployed damage it removes (–: not run). Perplexity: 20,000 WikiText-103 test tokens, unedited→edited, both unquantized (–: not measured for single edits). 200 probes at $n \in \{1, 1000\}$, 100 at $n = 100$. GPT-2-XL's $n = 1$ and $n = 1000$ fp columns agree only in the mean (6.057 nats each; efficacy 0.785 and 0.790): per probe they differ by 1.2 nats on average (correlation 0.94) and in efficacy on 7 of the 200 probes; the means differ by 0.0008 nats against a standard error of 0.17 for the paired difference, and no probe has the same value in both runs.

model	n	quant	fp	ES fp/dep.	weights-only	keys-only	full	K/F	even split	QAE recovered	ppl
GPT-2-XL	1	int3	6.06	0.79/0.62	−0.27 [−0.34, −0.21]	−3.78	−3.97	95.2%	94.1% [92.7, 95.4]	72% [65, 81]	–
GPT-2-XL	1	int4	6.06	0.79/0.76	−0.05 [−0.07, −0.03]	−0.65	−0.69	94.0%	93.4% [90.1, 96.7]	93% [60, 134]	–
GPT-2-XL	100	int3	7.33	0.84/0.72	−0.31 [−0.40, −0.23]	−4.23	−4.51	93.7%	93.4% [91.5, 95.0]	68% [58, 81]	21.1→21.1
GPT-2-XL	100	int4	7.33	0.84/0.83	−0.07 [−0.11, −0.03]	−0.72	−0.79	91.7%	91.5% [86.2, 96.2]	86% [48, 133]	21.1→21.1
GPT-2-XL	1000	int3	6.06	0.79/0.67	−0.34 [−0.42, −0.26]	−3.61	−3.85	93.9%	92.6% [91.0, 94.1]	60% [52, 69]	21.1→21.2
GPT-2-XL	1000	int4	6.06	0.79/0.76	−0.06 [−0.09, −0.03]	−0.53	−0.58	91.3%	90.1% [84.8, 95.0]	–	21.1→21.2
GPT-J-6B	1	int3	14.27	0.99/0.97	−0.09 [−0.18, −0.00]	−3.77	−3.91	96.5%	97.1% [95.2, 98.9]	87% [77, 96]	–
GPT-J-6B	1	int4	14.27	0.99/0.99	−0.04 [−0.07, −0.01]	−0.29	−0.32	88.6%	88.3% [75.4, 97.9]	121% [39, 251]	–
GPT-J-6B	100	int3	14.64	1.00/1.00	−0.04 [−0.16, +0.09]	−3.92	−4.08	96.1%	97.5% [94.8, 100.3]	76% [64, 88]	11.9→11.9
GPT-J-6B	100	int4	14.64	1.00/1.00	−0.04 [−0.10, +0.01]	−0.28	−0.31	89.0%	87.6% [56.4, 104.0]	131% [7, 413]	11.9→11.9
GPT-J-6B	1000	int3	13.63	0.99/0.97	−0.39 [−0.50, −0.29]	−3.73	−4.08	91.5%	91.0% [89.1, 92.8]	73% [64, 83]	11.9→12.0
GPT-J-6B	1000	int4	13.63	0.99/0.99	−0.01 [−0.05, +0.03]	−0.21	−0.24	88.5%	92.5% [71.9, 109.9]	–	11.9→12.0
Llama-3.1-8B	1	int3	13.21	1.00/0.90	−0.20 [−0.31, −0.09]	−7.24	−7.52	96.2%	96.7% [95.4, 98.1]	96% [88, 103]	–
Llama-3.1-8B	1	int4	13.21	1.00/1.00	−0.03 [−0.08, +0.02]	−1.23	−1.28	96.4%	97.0% [93.5, 100.4]	68% [33, 107]	–
Llama-3.1-8B	100	int3	13.22	1.00/0.89	−0.16 [−0.31, +0.00]	−7.34	−7.63	96.1%	97.0% [95.1, 98.9]	94% [83, 106]	13.5→13.5
Llama-3.1-8B	100	int4	13.22	1.00/1.00	−0.08 [−0.16, −0.00]	−1.49	−1.50	99.4%	97.0% [92.6, 101.6]	48% [4, 95]	13.5→13.5
Llama-3.1-8B	1000	int3	12.75	1.00/0.85	−0.38 [−0.51, −0.24]	−7.18	−7.75	92.6%	93.9% [92.4, 95.3]	78% [70, 86]	13.5→13.8
Llama-3.1-8B	1000	int4	12.75	1.00/1.00	−0.10 [−0.16, −0.03]	−1.34	−1.43	93.9%	93.5% [89.1, 97.3]	–	13.5→13.8

Table 47: Published AlphaEdit: the released null-space projection (eigenvalues of C below 2×10^{-2}), $L_2 = 10$ and the same bands. fp: unquantized rewrite margin; ES fp/dep.: efficacy unquantized / deployed. Margin changes in nats relative to the fp margin (negative = margin lost), paired 95% intervals for the weights-only channel (every keys-only and full-channel interval excludes zero); weights-only = the band quantized, keys-only = everything else. QAE-AlphaEdit recovered is the share of AlphaEdit's deployed damage it removes. Perplexity: 20,000 WikiText-103 test tokens, unedited→edited, both unquantized (–: not measured for single edits). 200 probes at $n \in \{1, 1000\}$, 100 at $n = 100$; the $n = 1000$, int4 cells were not run.

model	n	quant	fp	ES fp/dep.	weights-only	keys-only	full	K/F	even split	QAE-AlphaEdit recovered	ppl
GPT-2-XL	1	int3	12.48	0.99/0.92	−0.32 [−0.40, −0.25]	−5.99	−6.31	94.9%	94.9% [94.1, 95.7]	95% [90, 100]	–
GPT-2-XL	1	int4	12.48	0.99/0.99	−0.06 [−0.09, −0.04]	−1.00	−1.05	94.8%	94.5% [91.9, 96.9]	93% [71, 115]	–
GPT-2-XL	100	int3	12.82	1.00/0.96	−0.23 [−0.32, −0.13]	−5.67	−6.05	93.7%	95.0% [93.5, 96.4]	95% [86, 103]	21.1→21.1
GPT-2-XL	100	int4	12.82	1.00/1.00	−0.06 [−0.10, −0.03]	−0.94	−0.98	95.3%	94.4% [91.1, 97.5]	88% [60, 118]	21.1→21.1
GPT-2-XL	1000	int3	11.35	0.98/0.86	−0.34 [−0.44, −0.24]	−5.58	−5.89	94.8%	94.5% [93.3, 95.7]	80% [73, 87]	21.1→21.4
GPT-J-6B	1	int3	14.29	0.99/0.98	−0.09 [−0.17, −0.00]	−3.66	−3.82	95.9%	96.8% [95.0, 98.7]	88% [78, 99]	–
GPT-J-6B	1	int4	14.29	0.99/0.99	−0.05 [−0.08, −0.01]	−0.28	−0.32	87.9%	86.4% [72.0, 96.7]	127% [47, 251]	–
GPT-J-6B	100	int3	14.78	1.00/1.00	−0.19 [−0.31, −0.08]	−3.84	−4.14	92.7%	94.0% [91.5, 96.5]	75% [63, 88]	11.9→12.0
GPT-J-6B	100	int4	14.78	1.00/1.00	−0.03 [−0.08, +0.02]	−0.31	−0.35	89.6%	90.2% [72.2, 106.6]	126% [13, 336]	11.9→12.0
GPT-J-6B	1000	int3	13.84	0.99/0.97	−0.34 [−0.45, −0.24]	−3.69	−3.99	92.4%	91.9% [89.9, 93.9]	72% [62, 83]	11.9→12.1
Llama-3.1-8B	1	int3	13.18	1.00/0.91	−0.23 [−0.34, −0.11]	−6.89	−7.17	96.1%	96.5% [95.1, 97.9]	93% [85, 101]	–
Llama-3.1-8B	1	int4	13.18	1.00/1.00	−0.02 [−0.07, +0.03]	−1.29	−1.31	98.6%	98.7% [95.2, 102.3]	70% [36, 106]	–
Llama-3.1-8B	100	int3	12.86	1.00/0.90	−0.15 [−0.30, +0.00]	−6.89	−7.33	93.9%	95.9% [94.1, 97.7]	91% [79, 103]	13.5→13.6
Llama-3.1-8B	100	int4	12.86	1.00/1.00	−0.05 [−0.11, +0.02]	−1.50	−1.54	97.3%	97.1% [93.2, 100.8]	45% [−1, 87]	13.5→13.6
Llama-3.1-8B	1000	int3	11.74	0.99/0.84	−0.20 [−0.35, −0.06]	−6.69	−7.08	94.5%	95.8% [94.2, 97.4]	80% [72, 89]	13.5→14.4

Table 48: Published MEMIT, three disjoint batches of 100 facts at int3 (batch 0 is the one in Table 46). ES fp: unquantized efficacy. QAE recovered as in Table 46; perplexity of the edited, unquantized model.

model	batch	fp	ES fp	full	K/F	even split	QAE recovered	ppl
GPT-2-XL	0	7.33	0.84	-4.51 $[-5.22, -3.83]$	93.7%	93.4%	68% [58, 81]	21.1
GPT-2-XL	1	5.54	0.74	-3.74 $[-4.40, -3.10]$	94.8%	93.9%	78% [68, 90]	21.1
GPT-2-XL	2	6.44	0.86	-4.22 $[-4.89, -3.58]$	94.0%	94.0%	83% [70, 98]	21.1
GPT-J-6B	0	14.64	1.00	-4.08 $[-4.60, -3.58]$	96.1%	97.5%	76% [64, 88]	11.9
GPT-J-6B	1	13.53	0.98	-3.82 $[-4.43, -3.24]$	92.8%	93.3%	86% [72, 101]	11.9
GPT-J-6B	2	13.77	1.00	-4.09 $[-4.73, -3.50]$	91.7%	93.3%	78% [68, 89]	11.9
Llama-3.1-8B	0	13.22	1.00	-7.63 $[-8.52, -6.79]$	96.1%	97.0%	94% [83, 106]	13.5
Llama-3.1-8B	1	13.21	1.00	-7.69 $[-8.53, -6.89]$	94.4%	95.6%	84% [74, 93]	13.5
Llama-3.1-8B	2	11.35	1.00	-6.33 $[-7.08, -5.60]$	95.4%	97.1%	98% [87, 109]	13.5

U PROPOSITION 2 TESTED

For a single edit the identity $\|\Delta W\,\delta k\|/\|\Delta W k^*\| = \mathrm{KSNR}^{-1}$ is algebra; the content is whether $1/\mathrm{KSNR}$ predicts the keys-only damage. For a batch we compute it per fact with that fact's own u_i and δk_i, ignoring cross-talk from the other facts. $1/\mathrm{KSNR}$ describes the key displacement only (the edit's own output at the displaced key); the keys-only losses it is scored against include the read-out half of Appendix G. Within a model, sweeping the quantizer at fixed architecture: mean $1/\mathrm{KSNR}$ against mean keys-only nats lost is $0.0012 \to 0.00$ (int8), $0.0365 \to 0.51$ (NF4), $0.0505 \to 1.13$ (int4), $0.2906 \to 6.52$ (int3) on GPT-2-XL at $n = 1$; $0.0034 \to 0.01$, $0.1092 \to 0.94$, $0.1517 \to 1.54$, $0.4420 \to 6.72$ on Llama-3.2-3B; $0.0012 \to 0.00$, $0.0464 \to 0.42$, $0.0677 \to 0.65$, $0.2882 \to 4.67$ on GPT-J; $0.0027 \to 0.00$, $0.1173 \to 0.99$, $0.1429 \to 1.88$, $0.5124 \to 8.15$ on Llama-3.1-8B. Thirteen of fourteen model×batch-size blocks (GPT-2, GPT-2-XL, Llama-3.2-3B, GPT-J-6B and Llama-3.1-8B at every batch size run, 124M to 8B; GPT-2-medium has only two quantizers and is excluded) are ordered perfectly; the fourteenth (GPT-J, $n = 1000$, Spearman $+0.800$) misses because its NF4 loss (0.74 nats) exceeds its int4 loss (0.61). Bit width alone orders int8 (not run on GPT-2), the 4-bit pair and int3 in every block, so the only informative comparison inside a block is NF4 against int4. Adding Qwen2.5-3B's two blocks, $1/\mathrm{KSNR}$ gets it right in 15 of 16; so do the key displacement $\|\delta k\|/\|k^*\|$ alone and the constant guess that NF4 is gentler than int4, each in 15 of 16 (GPT-J at $n = 1000$ is the miss for all three). The quantizer ordering is therefore not evidence for Proposition 2 beyond the key displacement it contains. Across architectures, the twelve int3 cells of the four main models (three batch sizes each) give Spearman $r_s = +0.874$; since $1/\mathrm{KSNR}$ barely moves with batch size this is the between-model ordering, mean $1/\mathrm{KSNR}$ over the three batch sizes of 0.288 (GPT-J), 0.291 (GPT-2-XL), 0.444 (Llama-3.2-3B) and 0.519 (Llama-3.1-8B) against keys-only losses at $n = 1$ of 4.7, 6.5, 6.7 and 8.2 nats, and four points carry little weight on their own.

The displacement is coherent with the update. Proposition 2's identity separates $1/\mathrm{KSNR}$ into $|\cos(u, \delta k)|/|\cos(u, k^*)|$ and $\|\delta k\|/\|k^*\|$. Table 50 gives the median $|\cos(u, \delta k)|$ over the 200 MEMIT edits at $n = 1$: at int3 it is 0.285–0.416 on the four main models, 28–50× the $1/\sqrt{d_{\mathrm{in}}}$ of a random direction and 42–74× its median $|\cos|$ ($0.6745/\sqrt{d_{\mathrm{in}}}$), and grows with coarseness (NF4 10–26× and int8 4–5× the $1/\sqrt{d_{\mathrm{in}}}$; at int8 the median $|\cos(u, \delta k)|$ is 0.049, 0.055, 0.041 and 0.039 on GPT-2-XL, Llama-3.2-3B, GPT-J-6B and Llama-3.1-8B). The incoherent approximation $\cos(u, \delta k) \approx 1/\sqrt{d_{\mathrm{in}}}$ under-predicts $1/\mathrm{KSNR}$ by 28–49× at int3 on the four main models (14× on Qwen2.5-3B) and is not used anywhere. The ratio of the median $|\cos(u, \delta k)|$ to the mean $\cos(u, k^*)$ at int3 is 0.39–0.52 for every rank-one editor on GPT-2-XL (ridge $0.428/0.917$, MEMIT $0.355/0.718$, AlphaEdit $0.264/0.509$, KAE against the fp key $0.206/0.531$), so the displacement has a large component along the key itself, and since that component is negative, the directions aligned with k^* that editors produce are, empirically, *anti*-aligned with δk; raising $\cos(u, k^*)$ therefore raises $|\cos(u, \delta k)|$ almost in proportion and leaves $1/\mathrm{KSNR}$ nearly unchanged at $n = 1$ (0.277 against 0.291), which is why alignment is a weak lever on the keys-only channel for single edits; the ridge's larger keys-only trims are at $n \geq 100$ (Appendix N). Reading the keys directly (`code/key_geometry.py`, 200 probes per cell) gives the sign, and at int3 it is the same on every model we measured: the mean $\cos(k^*, \delta k)$ is -0.399 (GPT-2-XL), -0.456 (Llama-3.2-3B), -0.377 (GPT-J-6B), -0.487 (Llama-3.1-8B) and -0.639 (Qwen2.5-3B), shrinking towards zero as the grid gets finer (-0.02 to -0.05 at int8 on four models, $+0.02$ on Qwen2.5-3B, where the key moves by 2.5% of its norm). The quantized key's *component along* k^* shrinks on the four main models and reverses on Qwen2.5-3B $(1 + \cos(k^*, \delta k)\,\|\delta k\|/\|k^*\|$, from the tabulated means, is 0.76, 0.62, 0.76, 0.56 and -0.08 at int3), so the delivered edit $a\,(1 + u^\top \delta k / u^\top k^*)$ is attenuated rather than randomized (annihilated on Qwen2.5-3B); the key's norm itself, from the tabulated means, barely changes on four models (0.94–0.97×) and grows on Qwen2.5-3B (1.30×). On GPT-2-XL the displacement puts 0.366 of its energy in the lowest-eigenvalue half of C against 0.269 for k^* and 0.807 for u, so most of the coherence is the anti-alignment with the key ($0.718 \times 0.399 = 0.29$ of the mean $-\cos(u, \delta k) = 0.350$, Table 49) and the rest comes from the low-variance directions that C^{-1} up-weights.

The attenuation factor, computed before deployment. Write

$$\rho = 1 + \frac{u^\top \delta k}{u^\top k^*} = 1 + \frac{\cos(u, \delta k)\,\|\delta k\|}{\cos(u, k^*)\,\|k^*\|}$$

for the fraction of the edit that reaches the read-out under the deployed backbone. It needs one forward pass through the quantized model and no editing, no evaluation and no labels. Table 49 gives it per model and quantizer (mean over the 200 edits of the per-edit ρ) against the fraction of its fp margin MEMIT actually keeps in deployment at int3 (ratio of means, full channel). Across the five models the two track each other (Pearson $r = +0.962$, Fisher 95% interval $[0.53, 1.00]$; Spearman $+0.900$, exact two-sided $p = 0.08$), but Qwen2.5-3B carries the correlation: without it Pearson is $+0.655$ ($[-0.83, 0.99]$) and Spearman $+0.800$ over four points ($p = 0.33$; four points cannot give an exact two-sided p below $2/24 = 0.083$), and we claim no more from it; the sign is the more informative result. On Qwen2.5-3B at int3 the key moves by $1.68\times$ its own norm and $\rho = -0.014$ (per edit: median 0.001, interquartile range -0.010 to $+0.012$): the geometry says the delivered edit all but vanishes, and the measured deployed margin is -0.22 against $+14.56$ in fp and -0.25 for the *unedited* model with its backbone at int3 (keys-only channel) on the same prompts (Table 26). The edit is annihilated, not reversed, and ρ was computable before any edit was applied. Within a model ρ also orders the individual edits by the fraction of margin retained (MEMIT, $n = 1$, int3), weakly but significantly, on three of the five models: Spearman $+0.243$ (GPT-2-XL), $+0.245$ (Llama-3.2-3B) and $+0.405$ (Llama-3.1-8B) over 199–200 probes (edits with an fp margin below 1 nat are excluded, since their retained fraction is undefined in practice), all $p < 10^{-3}$ by a two-sided permutation test (20,000 permutations). On GPT-J-6B it is -0.004 ($p = 0.96$), a clean null, and on Qwen, where nothing survives, -0.133 ($p = 0.06$) over a ratio that is noise about zero. Scored instead against the upstream-only retention of Appendix G (the edited matrix reads k_q^* through a full-precision read-out), the quantity the identity strictly describes, the within-model Spearman rises to $+0.271$, $+0.378$ and $+0.627$ on the same three models, with GPT-J at $+0.098$ ($p = 0.17$) and Qwen at -0.151 (the key-geometry run of Table 49 and the split run of Appendix G, whose probes coincide). Between-model, then, the factor tracks the damage; within a model it is a weak signal that fails on two of five, and we report it as such.

Table 49: Key displacement and the attenuation factor $\rho = 1 + u^\top \delta k / u^\top k^*$ (MEMIT, $n = 1$, 200 probes per cell; all columns but the retained fraction are means of per-edit values, so ρ is not the product of the tabulated means), with the fraction of its fp margin MEMIT retains in deployment at int3 (ratio of means, from the deployment-precision run of Table 2); $\cos(u, k^*)$ is in Table 50.

model	quant	$\cos(k^*, \delta k)$	$\|\delta k\|/\|k^*\|$	$\cos(u, \delta k)$	ρ	margin retained
GPT-2-XL	int8	-0.024	0.015	-0.014	$+1.000$	—
GPT-2-XL	NF4	-0.132	0.218	-0.116	$+0.965$	—
GPT-2-XL	int4	-0.158	0.266	-0.135	$+0.950$	—
GPT-2-XL	int3	-0.399	0.593	-0.350	$+0.710$	0.495
Llama-3.2-3B	int8	-0.053	0.038	-0.043	$+0.998$	—
Llama-3.2-3B	NF4	-0.253	0.348	-0.231	$+0.891$	—
Llama-3.2-3B	int4	-0.311	0.421	-0.268	$+0.848$	—
Llama-3.2-3B	int3	-0.456	0.832	-0.394	$+0.558$	0.516
GPT-J-6B	int8	-0.027	0.015	-0.022	$+0.999$	—
GPT-J-6B	NF4	-0.151	0.220	-0.131	$+0.954$	—
GPT-J-6B	int4	-0.206	0.272	-0.156	$+0.933$	—
GPT-J-6B	int3	-0.377	0.643	-0.287	$+0.712$	0.684
Llama-3.1-8B	int8	-0.020	0.042	-0.014	$+0.999$	—
Llama-3.1-8B	NF4	-0.247	0.378	-0.218	$+0.883$	—
Llama-3.1-8B	int4	-0.266	0.457	-0.217	$+0.857$	—
Llama-3.1-8B	int3	-0.487	0.911	-0.403	$+0.488$	0.449
Qwen2.5-3B	int8	$+0.020$	0.025	$+0.000$	$+0.998$	—
Qwen2.5-3B	NF4	-0.124	0.250	-0.034	$+0.978$	—
Qwen2.5-3B	int4	-0.214	0.434	-0.049	$+0.900$	—
Qwen2.5-3B	int3	-0.639	1.684	-0.142	-0.014	-0.015

What sets the size of the displacement across models. $\|\delta k\|/\|k^*\|$ at int3 runs from 0.593 on GPT-2-XL to 1.684 on Qwen2.5-3B, a factor of three the mechanism does not itself explain. The outlier-feature account of quantization (Dettmers et al., 2022; Xiao et al., 2023; Sun et al., 2024) is a candidate: a small number of input channels carry magnitudes far above the rest, and a grid fitted to a group containing one of them is coarse for the others. The covariance C that every editor in this family already estimates carries that structure on its diagonal, $\mathrm{diag}(C)_j = \mathbb{E}[k_j^2]$, so no new measurement is needed (`code/act_outliers.py`; it describes the edited layer's own input, not

the inputs of the upstream matrices whose rounding moves the key, so it is a proxy). Taking the mean root-mean-square magnitude $\sqrt{\mathrm{diag}(C)_j}$ of the top 1% of channels over the median channel's gives 2.45 (GPT-2-XL), 2.56 (GPT-J-6B), 2.86 (Llama-3.2-3B), 3.03 (Llama-3.1-8B) and 8.41 (Qwen2.5-3B), which orders $\|\delta k\|/\|k^*\|$ $(0.593, 0.643, 0.832, 0.911, 1.684)$ exactly: Spearman $+1.000$ on five models, one of 120 orderings. Two caveats keep this an observation rather than a test. We chose the statistic after seeing the data: the cruder $\max/\mathrm{median}$ gives $+0.700$ and the kurtosis of the channel magnitudes -0.300, so two of three candidates do not order it. With only five models the ordering is weak evidence; the four main models are also ordered by how much the main grid's int3 degrades their unedited perplexity (1.18, 1.37, 1.94 and 2.67×, Table 53; Qwen2.5-3B's, measured separately, is 1,967×, Appendix V, so perplexity damage orders all five models as the key displacement does) and, for the three Instruct checkpoints, by chat tuning, and the three largest displacements are also the three gated (SwiGLU) MLPs, whose key is the product of two quantized projections; the statistic is confounded with everything else that differs between them, and separating these needs within-model interventions (outlier smoothing, base against Instruct) that we have not run (the W8A8 runs of Appendix E smooth outliers but also round activations, so they do not separate them). It does, however, make concrete the two results we could not otherwise account for. Qwen2.5-3B, whose predictor calibration and whose fix both fail, is the outlier-dominated model by a factor of nearly three, and the four models split by the same statistic into the two where GPTQ helps the weight channel at batch scale (2.45, 2.56) and the two where it hurts or is neutral (2.86, 3.03), which is also the split of the $n = 1000$ overload (single-layer updates 11.6 and 7.5 times the matrix's norm against 2.2 and 1.7). It also says where to look: a quantizer that protects the salient input channels should displace the key less, which is what AWQ does to the key on the two models measured (Appendix Q: $\|\delta k\|/\|k^*\|$ $0.593 \to 0.548$ on GPT-2-XL and $0.832 \to 0.748$ on Llama-3.2-3B at int3).

Table 50: MEMIT, $n = 1$, 200 edits per cell: mean $1/\mathrm{KSNR}$, mean $\cos(u, k^*)$, mean $\|\delta k\|/\|k^*\|$, median $|\cos(u, \delta k)|$ (a magnitude; Table 49 gives the signed mean), its ratio to $1/\sqrt{d_{\mathrm{in}}}$ (the root-mean-square $|\cos|$ of a random direction; its median is $0.6745/\sqrt{d_{\mathrm{in}}}$), and the $1/\mathrm{KSNR}$ the incoherent approximation would give. Means and medians do not compose exactly through the identity. From the main grid; Table 49 is the separate key-geometry run, so a shared cell can differ in the last digit.

model	quant	$1/\mathrm{KSNR}$	$\cos(u, k^*)$	$\|\delta k\|/\|k^*\|$	$\lvert\cos(u, \delta k)\rvert$	× random	incoherent $1/\mathrm{KSNR}$
GPT-2-XL	NF4	0.036	0.718	0.218	0.121	10×	0.0038
GPT-2-XL	int4	0.050	0.718	0.266	0.135	11×	0.0046
GPT-2-XL	int3	0.291	0.718	0.593	0.355	28×	0.0103
Llama-3.2-3B	NF4	0.109	0.745	0.347	0.233	21×	0.0051
Llama-3.2-3B	int4	0.152	0.745	0.421	0.263	24×	0.0062
Llama-3.2-3B	int3	0.442	0.745	0.832	0.395	36×	0.0123
GPT-J-6B	NF4	0.046	0.634	0.220	0.129	17×	0.0027
GPT-J-6B	int4	0.068	0.634	0.272	0.158	20×	0.0034
GPT-J-6B	int3	0.288	0.634	0.643	0.285	36×	0.0079
Llama-3.1-8B	NF4	0.117	0.727	0.378	0.220	26×	0.0043
Llama-3.1-8B	int4	0.143	0.727	0.457	0.231	28×	0.0052
Llama-3.1-8B	int3	0.512	0.727	0.911	0.416	50×	0.0105
Qwen2.5-3B	NF4	0.080	0.219	0.250	0.047	5×	0.0109
Qwen2.5-3B	int4	0.152	0.219	0.434	0.053	6×	0.0189
Qwen2.5-3B	int3	1.014	0.219	1.684	0.135	14×	0.0734

Holding model, editor, batch size and quantizer fixed, $1/\mathrm{KSNR}$ ranks which edit loses more margin (in nats) only weakly: median within-cell Spearman $r_s = +0.158$ for MEMIT (the 12 int3 cells of the four main models, range -0.020 to $+0.366$), -0.174 for KAE (the expected sign, since KAE corrects for a large δk); the $+0.24$ to $+0.41$ reported above for three of the same $n = 1$ cells use a different outcome (ρ equals $1 - 1/\mathrm{KSNR}$ to within 0.011 on all but one of these edits, so the predictor is the same up to sign, but the outcome there is the fraction of margin retained, not nats lost), which is why the largest of them lies outside this range.

The ρ-rescaling baseline. If the key displacement only attenuated the delivered edit by ρ, dividing MEMIT's update by each edit's ρ (one extra forward pass through the quantized backbone and no re-solve for single edits; in a batch each fact's residual is divided by its own ρ before a joint re-solve; ρ floored at 0.1, a floor no probed edit reaches on the two models on which it was run, $1/\mathrm{KSNR} \le 0.9$ on every probe of Table 51) would undo it. Table 51 runs that baseline against MEMIT and KAE

in the same run. It returns about what KAE returns: its deployed margin is within 0.41 nats of KAE's in all eight cells; it returns on GPT-2-XL 43–52% of MEMIT's deployed loss against KAE's 46–52%, on Llama-3.2-3B 24–38% against 22–33% (int3 and int4, $n \in \{1, 100\}$). Paired against KAE on the same probes, its deployed margin differs by -0.41 to $+0.20$ nats, significantly in three cells: -0.06 $[-0.12, -0.00]$ on GPT-2-XL at int3, $n = 1$, and $+0.09$ $[+0.03, +0.14]$ and -0.41 $[-0.78, -0.04]$ on Llama-3.2-3B at int4, $n = 1$, and at int3, $n = 100$. The rescaled update is larger ($\|\Delta W\|_F / \|W\|_F$ up to 1.20 at $n = 100$ on Llama-3.2-3B, against MEMIT's 0.64) and costs locality there (ΔNS -0.065 $[-0.100, -0.035]$ at $n = 1$ and -0.110 $[-0.190, -0.030]$ at $n = 100$, int3). The two are not the same update: dividing by ρ restores the increment a at k_q^* but keeps MEMIT's direction u and the baseline shift $W\delta k$, which KAE's residual $v^* - Wk_q^*$ removes, so their agreement says that these two differences matter little for the deployed margin. Proposition 2's attenuation is therefore, on these two models, the key-aware part of the damage, and the rest, the read-out, is what the value re-optimization of QAE returns.

Table 51: MEMIT with each edit's update divided by its attenuation $\rho = u^\top k_q^* / u^\top k^*$ (Proposition 2; floored at 0.1), against MEMIT and KAE in the same run: fp / deployed margin, share of MEMIT's deployed damage returned, and the $1/\rho$ edit's ΔES and ΔNS against MEMIT (deployed, paired 95% CI; bold = CI excludes zero). 200 probes at $n = 1$, the batch's 100 at $n = 100$.

model	quant	n	MEMIT	KAE	KAE rec.	$1/\rho$	$1/\rho$ rec.	ΔES	ΔNS
GPT-2-XL	int3	1	12.84 / 6.35	9.91 / 9.76	52% [48, 57]	14.70 / 9.70	52% [47, 56]	**+0.070** [+0.035, +0.110]	+0.000 [+0.000, +0.000]
GPT-2-XL	int4	1	12.84 / 11.71	12.33 / 12.29	51% [44, 59]	13.35 / 12.26	49% [42, 57]	+0.000 [+0.000, +0.000]	−0.010 [−0.025, +0.000]
GPT-2-XL	int3	100	13.13 / 6.69	9.50 / 9.76	48% [43, 53]	14.80 / 9.96	51% [46, 56]	**+0.060** [+0.020, +0.110]	−0.020 [−0.050, +0.000]
GPT-2-XL	int4	100	13.13 / 11.99	12.46 / 12.52	46% [38, 57]	13.58 / 12.48	43% [35, 52]	+0.000 [+0.000, +0.000]	+0.000 [+0.000, +0.000]
Llama-3.2-3B	int3	1	13.73 / 6.98	9.55 / 9.20	33% [28, 38]	12.67 / 9.18	33% [28, 38]	+0.010 [−0.010, +0.030]	**−0.065** [−0.100, −0.035]
Llama-3.2-3B	int4	1	13.73 / 12.18	13.27 / 12.69	33% [26, 41]	13.90 / 12.78	38% [31, 48]	+0.000 [+0.000, +0.000]	−0.015 [−0.035, +0.000]
Llama-3.2-3B	int3	100	13.76 / 6.77	9.05 / 8.83	29% [22, 37]	12.05 / 8.42	24% [16, 31]	+0.020 [−0.030, +0.070]	**−0.110** [−0.190, −0.030]
Llama-3.2-3B	int4	100	13.76 / 12.28	12.86 / 12.61	22% [8, 38]	13.77 / 12.73	30% [19, 44]	+0.000 [+0.000, +0.000]	−0.020 [−0.050, +0.000]

V PERPLEXITY

Does the deployment-precision solve break the language model? Table 52 answers it in one run per model, so that all four editors share C, the cached values and the test split. For single edits, one edited fact per model, QAE's perplexity is 0.9944–$0.9994\times$ MEMIT's; the two perplexity runs' shared single-edit cells differ by about 0.1% on Llama-3.2-3B and agree to rounding on GPT-2-XL. At $n = 100$ it is 1.0040 and $1.0501\times$ MEMIT's, against 1.0060 and $1.1325\times$ for the key-aware solve alone: solving for the deployment value adds almost none of the fluency cost ($\leq 0.8\%$) that solving for the deployment key does, and adding it to the key partly offsets the key's cost, which is what one would expect of a value fitted with ROME's essence-drift KL term against the deployed network.

What the key-aware solve costs at batch scale. Table 53 gives the int3 rows of the grid and its fp rows at $n = 1000$, and the column a reader should look at is KAE's at $n = 1000$. There the edit alone has already damaged the model: MEMIT's own $n = 1000$ int3 edit takes held-out perplexity to 1.8–$22\times$ the unedited value, and at full precision to $5.4\times$ and $6.5\times$ on GPT-2-XL and GPT-J (the unweighted-covariance overload of Appendix A). Re-solving at the deployed key makes that worse by a further $1.05\times$ (GPT-2-XL), $1.28\times$ (Llama-3.1-8B), $1.74\times$ (GPT-J-6B) and $2.54\times$ (Llama-3.2-3B). A likely mechanism is the alignment trade of Corollary 2, extended past its rank-one setting: KAE trades alignment with k^* for alignment with k_q^*, and a less aligned update of the same delivered size is a larger perturbation, which a thousand of them make visible. At $n = 100$ the same effect is 1.01–$1.14\times$ MEMIT (1.01–$1.21\times$ the unedited model) and at $n = 1$ it is nil. Durability and fluency therefore respond to different interventions: on GPT-2-XL and GPT-J at $n = 1000$/int3 the alignment ridge reaches $0.22\times$ and $0.52\times$ MEMIT's perplexity, whereas on the two Llamas it is $1.20\times$ and $1.01\times$ MEMIT's, so the ridge is what preserves fluency on the two models where the weight channel is large and not elsewhere. Quantization's own perplexity cost is large (int3 takes Llama-3.1-8B from 14.52 to 38.72 before any edit exists; 1.2–$2.7\times$ over the four models; on Qwen2.5-3B, measured separately on the first 99,840 test tokens under its own tokenizer with `code/wa_ppl.py`, int8, NF4 and int4 give 1.00, 1.07 and $1.20\times$ its unquantized 11.01, and symmetric int3 $21{,}654$, so its int3 model is broken before any edit) and is not the edit's fault, so the edit is charged only for what it adds on top of the deployed baseline. The fp rows at $n = 1000$ involve no quantizer and are the editor's own damage, which separates the two overloaded models from the two Llamas as $\|\Delta W\|_F / \|W\|_F$

Table 52: Held-out WikiText-103 perplexity of the deployment-precision solve at int3, one run per model ($n = 0$ is the unedited model; an $n = 1$ row is one edited model, the first probe fact). +key, +value and +both (QAE) as in Table 2; "× unedited" and "× MEMIT" are ratios to the unedited model and to MEMIT *at the same precision*.

model	n	editor	ppl	× unedited	× MEMIT
GPT-2-XL	0	unedited	24.615	1.0000	—
GPT-2-XL	1	MEMIT	24.617	1.0001	1.0000
GPT-2-XL	1	+key	24.613	0.9999	0.9998
GPT-2-XL	1	+value	24.606	0.9996	0.9995
GPT-2-XL	1	**+both (QAE)**	24.601	0.9995	0.9994
GPT-2-XL	100	MEMIT	24.765	1.0061	1.0000
GPT-2-XL	100	+key	24.914	1.0122	1.0060
GPT-2-XL	100	+value	24.694	1.0032	0.9972
GPT-2-XL	100	**+both (QAE)**	24.863	1.0101	1.0040
Llama-3.2-3B	0	unedited	37.314	1.0000	—
Llama-3.2-3B	1	MEMIT	37.218	0.9974	1.0000
Llama-3.2-3B	1	+key	37.092	0.9941	0.9966
Llama-3.2-3B	1	+value	37.220	0.9975	1.0000
Llama-3.2-3B	1	**+both (QAE)**	37.008	0.9918	0.9944
Llama-3.2-3B	100	MEMIT	39.677	1.0633	1.0000
Llama-3.2-3B	100	+key	44.934	1.2042	1.1325
Llama-3.2-3B	100	+value	39.993	1.0718	1.0080
Llama-3.2-3B	100	**+both (QAE)**	41.666	1.1166	1.0501

Table 53: WikiText-103 test perplexity (99,840 tokens, identical for every row within a model; an $n = 1$ row edits the first probe fact only); in brackets, the ratio to the unedited model at the same precision. At fp, KAE coincides with MEMIT by construction; "ridge" is the alignment ridge.

model	quant	n	unedited	MEMIT	KAE	ridge
GPT-2-XL	int3	1	24.61	24.62 (1.00×)	24.61 (1.00×)	24.63 (1.00×)
GPT-2-XL	int3	100	24.61	24.76 (1.01×)	24.91 (1.01×)	24.81 (1.01×)
GPT-2-XL	fp	1000	20.80	111.90 (5.38×)	111.90 (5.38×)	39.21 (1.88×)
GPT-2-XL	int3	1000	24.61	286.01 (11.62×)	300.50 (12.21×)	62.70 (2.55×)
Llama-3.2-3B	int3	1	37.31	37.25 (1.00×)	37.05 (0.99×)	37.23 (1.00×)
Llama-3.2-3B	int3	100	37.31	39.72 (1.06×)	45.21 (1.21×)	39.87 (1.07×)
Llama-3.2-3B	fp	1000	19.21	24.39 (1.27×)	24.39 (1.27×)	27.82 (1.45×)
Llama-3.2-3B	int3	1000	37.31	133.44 (3.58×)	338.54 (9.07×)	159.71 (4.28×)
GPT-J-6B	int3	1	16.34	16.33 (1.00×)	16.35 (1.00×)	16.34 (1.00×)
GPT-J-6B	int3	100	16.34	16.56 (1.01×)	17.47 (1.07×)	16.60 (1.02×)
GPT-J-6B	fp	1000	11.97	77.50 (6.47×)	77.50 (6.47×)	58.18 (4.86×)
GPT-J-6B	int3	1000	16.34	361.14 (22.10×)	627.26 (38.38×)	188.94 (11.56×)
Llama-3.1-8B	int3	1	38.72	38.81 (1.00×)	38.68 (1.00×)	38.80 (1.00×)
Llama-3.1-8B	int3	100	38.72	40.08 (1.03×)	41.90 (1.08×)	40.30 (1.04×)
Llama-3.1-8B	fp	1000	14.52	16.37 (1.13×)	16.37 (1.13×)	17.34 (1.19×)
Llama-3.1-8B	int3	1000	38.72	69.32 (1.79×)	88.79 (2.29×)	70.28 (1.81×)

does (11.62, 2.24, 7.46, 1.71 for GPT-2-XL, Llama-3.2-3B, GPT-J and Llama-3.1-8B against 5.38, 1.27, 6.47 and 1.13×; four points, descriptive only; GPT-2-XL's norm ratio, 11.623, equals its int3 perplexity ratio in Table 53, 11.620, to two decimals by coincidence). Tables 52 and 53 are separate runs, which is why their shared Llama-3.2-3B cells differ by up to 0.3 perplexity (about 0.6%); over four models, three editors (MEMIT, KAE, the ridge) and int8/int4/int3 in the run of Table 53, of which the table prints the int3 rows, the single edit (one fact per model) moves perplexity by no more than 0.70% from the unedited baseline (each single-edit row is one edited model, the edit of the lowest-identifier probe); QAE's single-edit change in Table 52 is −0.06 to −0.56% against MEMIT.

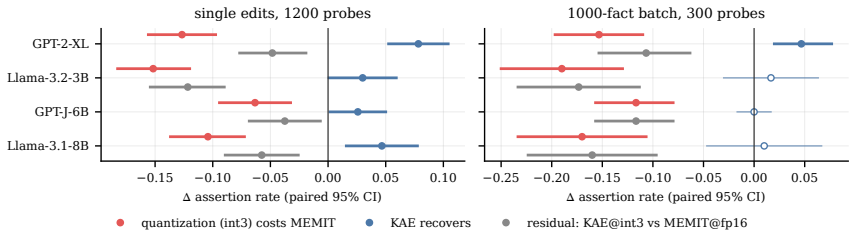


Figure 5: **Effect on generated text.** Free continuations scored by an LLM judge (`gpt-4o-mini`) for whether they assert or imply the new object, paired per probe; left, single edits (1,200 probes per model, a size chosen after a 400-probe look); right, one 1000-fact batch (300 probes from inside it; KAE's recovery covers zero on three of four models there); open markers, 95% CI includes zero (at the 97.5% that allows for the look, KAE's single-edit recoveries on Llama-3.2-3B and GPT-J-6B include zero too). A second judge reproduces all twelve single-edit contrasts. "fp16" in the legend and in this appendix is the unquantized 16-bit model (bf16 on the Llamas). The QAE generation run (200 probes, GPT-2-XL and Llama-3.2-3B) is reported in §4.5 and below.

W BEHAVIOURAL ROBUSTNESS

A second judge. Every stored generation was re-scored by `deepseek-chat` with the same judge prompt (`code/gen_judge.py`; Appendix A gives its content). For the twelve single-edit contrasts (four models × {quantization costs MEMIT, KAE recovers, residual gap}) the second judge gives, as signed changes in the assertion rate (a cost is negative), $-0.250/+0.165/-0.085$ (GPT-2-XL), $-0.202/+0.052/-0.149$ (Llama-3.2-3B), $-0.107/+0.047/-0.060$ (GPT-J-6B) and $-0.162/+0.065/-0.097$ (Llama-3.1-8B) against the first judge's $-0.127/+0.078/-0.048$, $-0.152/+0.030/-0.122$, $-0.063/+0.026/-0.037$ and $-0.104/+0.047/-0.058$: every contrast has the same sign and is significant under both at 95%, and the judge we report gives the smaller magnitude in all twelve contrasts (the smaller cost, recovery and residual gap of the two judges; the human audit below does not settle which judge's magnitudes are right). Raw agreement is 0.578–0.665 (Cohen's κ 0.400–0.488 on the three-way verdict and 0.545–0.571 for the new object against the rest; Cohen, 1960), but the structure of the disagreement is that the stricter judge's (`gpt-4o-mini`'s) set of "asserts the new object" verdicts sits inside the looser one's (`deepseek-chat`'s; 0.927–0.960 containment): the stricter judge's positives are nested in the looser judge's, and the two differ on where they put the line, which a paired within-probe contrast is largely insensitive to. The single-edit runs were extended from 400 to 1,200 probes after two contrasts (GPT-J's cost, Llama-3.1-8B's recovery) came out significant under one judge only; at 1,200 both judges agree on all twelve. The larger sample was chosen after seeing the smaller one, so it is not pre-specified. Testing each contrast at 0.025 to allow for the two looks (97.5% intervals, first judge), the four costs and the recoveries on GPT-2-XL and Llama-3.1-8B still exclude zero; the recoveries on Llama-3.2-3B ($+0.030$ $[-0.003, +0.063]$) and GPT-J ($+0.026$ $[-0.002, +0.053]$) do not. The 800 probes the first look never scored (401–1,200) give the same verdicts on their own, at 95% under the first judge: the four costs (0.083–0.164) and the recoveries on GPT-2-XL ($+0.084$ $[+0.051, +0.116]$) and Llama-3.1-8B ($+0.044$ $[+0.006, +0.081]$) exclude zero, those on Llama-3.2-3B ($+0.018$ $[-0.018, +0.052]$) and GPT-J ($+0.024$ $[-0.005, +0.054]$) do not; under the second judge all eight exclude zero. The 400-probe look scored the 400 lowest identifiers, the first 400 rows of the result files. In the batch panel of Figure 5 (one 1000-fact batch, 300 probes, first judge; on GPT-2-XL and GPT-J a batch that already takes the unquantized perplexity to $5.4\times$ and $6.5\times$, Table 53) int3 costs MEMIT more than for single edits, 0.153 $[0.110, 0.197]$, 0.190 $[0.130, 0.250]$, 0.117 $[0.080, 0.157]$ and 0.170 $[0.107, 0.233]$ of the assertion rate on GPT-2-XL, Llama-3.2-3B, GPT-J-6B and Llama-3.1-8B, and KAE's recovery excludes zero only on GPT-2-XL ($+0.047$ $[+0.020, +0.077]$; $+0.000$ $[-0.017, +0.017]$ on GPT-J-6B).

A human audit. Two annotators labelled 240 stored continuations independently and blind (`code/human_audit.py`, seed 2027): on each of the four main models, 40 drawn uniformly from the judged (probe, condition) pairs over the unedited, MEMIT fp16, MEMIT int3 and KAE int3 continuations, and 20 from the pairs on which the two judges disagree. Each item shows the subject, the relation, the two candidate objects in random order and the continuation, and neither the condition nor any verdict. Each annotator labelled from a written guide with worked examples.

The annotators agree with each other at Cohen's $\kappa = 0.726$ on the 160 uniform items (0.738 for the new object against the rest). The judges agree with them much less. On the 139 uniform items where the annotators agree, κ on the three-way verdict is 0.221 for `gpt-4o-mini` and 0.428 for `deepseek-chat`, mostly because `gpt-4o-mini` calls 61 of the 97 continuations that assert neither object the true object; on the new object against the rest, the distinction the assertion rate uses, it is 0.523 and 0.569. There the judges err differently, `gpt-4o-mini` both ways (12 assertions missed, 12 invented) and `deepseek-chat` one way (none missed, 28 invented): `gpt-4o-mini` has sensitivity 0.64 and specificity 0.89 and reproduces the annotators' assertion rate (0.237 against 0.237), while `deepseek-chat` has sensitivity 1.00 and specificity 0.74 and overstates it (0.439). No result in the paper uses the true-object verdicts. Against each annotator separately on all 160 uniform items, including the 21 on which the annotators disagree, κ is 0.200 and 0.234 for `gpt-4o-mini` and 0.386 and 0.438 for `deepseek-chat` on the three-way verdict, and 0.452 and 0.499, and 0.506 and 0.557, on the new object against the rest. On the uniform items (pooled over models; each annotator's labels averaged) the annotators see MEMIT's assertion rate fall from 0.486 at fp16 to 0.223 at int3 (a drop of 0.262 [0.067, 0.451]; the items differ between conditions, of which the unedited, MEMIT fp16, MEMIT int3 and KAE int3 have 31, 35, 47 and 47 of the 160, so the contrast is unpaired), and both judges see a drop too (0.316 under `gpt-4o-mini`, 0.296 under `deepseek-chat`). KAE's +0.064 [−0.096, +0.223] over MEMIT at int3 is too small for the 47 items in each int3 condition to resolve under the annotators or `gpt-4o-mini` (+0.021 [−0.149, +0.191]); `deepseek-chat` gives +0.213 [+0.021, +0.404]. On this pooled sample `gpt-4o-mini`'s assertion rate matches the annotators', which with sensitivity 0.64 and specificity 0.89 is a property of the base rate, not of the rater, and its individual verdict is right about four times in five. With condition-independent errors its contrasts would be attenuated (by sensitivity + specificity $-1 \approx 0.5$, so the annotators' 0.262 would read as about 0.14); its drop is larger than the annotators', which the audit's intervals cannot distinguish from attenuation or inflation, so 240 items do not settle it.

QAE generations under both judges. The 200-probe QAE generation run of §4.5 was re-scored by the second judge on the stored text. On GPT-2-XL quantization costs MEMIT 0.200 [0.130, 0.270] of the assertion rate and QAE returns +0.235 [+0.170, +0.300] (first judge 0.175 and +0.200); on Llama-3.2-3B QAE again returns nothing significant (+0.040 [−0.040, +0.120]; first judge +0.035 [−0.040, +0.110] of a 0.145 cost), and its deployed assertion rate stays below the unquantized edit's under both judges (−0.110 [−0.195, −0.025] and −0.120 [−0.200, −0.040]). Under neither judge is QAE shown equivalent to the unquantized edit within ±0.05 (a margin we chose, not one of the logged predictions of Appendix Z) on GPT-2-XL (90% intervals [−0.040, +0.090] under the first judge and [−0.020, +0.090] under the second): the upper bounds exceed +0.05, but the lower bounds lie above −0.05, so at that margin QAE is non-inferior to the unquantized edit under both judges (one-sided, 5%).

A prediction logged in advance failed here: that the surviving margin would order the behavioural damage across models (MEMIT's deployed int3 margin at $n = 1$, Table 11, orders them GPT-2-XL, Llama-3.1-8B, Llama-3.2-3B, GPT-J-6B, lowest first; the first judge's costs order them Llama-3.2-3B, GPT-2-XL, Llama-3.1-8B, GPT-J-6B, largest first), with Llama-3.1-8B predicted at −0.233 to −0.150; it came in at −0.092 [−0.145, −0.040] at 400 probes and −0.104 at 1,200 under the first judge, against which it was logged (−0.162 under the second, inside the band).

Calibration budget. Llama-3.2-3B was run at 1,200 probes under two covariance budgets (100k and 500k tokens) whose edits differ by roughly 30% in relative Frobenius distance; every behavioural contrast moves by less than 0.006, inside its own half-width of ±0.03.

Margin to behaviour. With the margin recorded in the same model state that produced each continuation, the margin lost to quantization predicts the behavioural flip (the new object asserted at fp16 and not at int3) at AUROC 0.656 [0.623, 0.691] on GPT-2-XL against a control (the fp16 margin predicting the same verdict) of 0.517; on Llama-3.1-8B 0.567 vs 0.533, Llama-3.2-3B 0.561 vs 0.535, GPT-J 0.538 vs 0.532. The standardized separations of margin lost between flipped and surviving probes (0.527, 0.246, 0.215, 0.138 sd) imply AUROCs within 0.011 of those measured under the equal-variance binormal model, so the models differ in effect size and not in what 1,200 probes can see; the link is item-level on GPT-2-XL and weak on the other three.

ZsRE generations (400 probes, int3, first judge; GPT-2-XL's numbers first): the assertion rate is 0.033 unedited, 0.147 after MEMIT at fp16, 0.085 at int3 and 0.138 for KAE at int3; the paired contrasts (the edit's effect at fp16, the quantization cost, KAE's recovery, and the residual gap)

are $+0.115$ $[+0.085, +0.147]$, -0.062 $[-0.092, -0.033]$, $+0.052$ $[+0.025, +0.083]$ and a residual of -0.010 $[-0.048, +0.025]$ (n.s.). On Llama-3.2-3B: $0.058/0.185/0.110/0.135$, contrasts $+0.128$ $[+0.092, +0.163]$, -0.075 $[-0.117, -0.035]$, $+0.025$ $[-0.013, +0.062]$ (n.s.) and -0.050 $[-0.092, -0.010]$.

X PER-EDIT PREDICTION

Does ESNR say *which* edit will fail? Pooled over $n = 1, 100, 1000$ within int3 on GPT-2-XL it scores AUROC 0.744, but batch size moves both ESNR and the failure rate enormously, so that number ranks cells, not edits. Holding quantizer and batch size fixed on GPT-2-XL but still pooling the six editors (full channel), ESNR^2's AUROC (score $-\mathrm{ESNR}^2$, so below 0.5 is backwards) is 0.371 (int3, $n = 1$, 72 failures of 1,200), 0.393 (int3, $n = 100$), 0.578 (int3, $n = 1000$), 0.478, 0.496 and 0.458 (int4, NF4, int8 at $n = 1000$), and pooling editors is itself confounded (on Llama-3.1-8B at int3, $n = 1$, KAE has a third of MEMIT's median ESNR^2 and a quarter of its failure rate; for weights-only failures, all of them KAE or KAE+KRE edits, whose weights-only channel reads the fp key, $-\mathrm{ESNR}^2$ pooled over the six editors of each (quantizer, n) cell reaches, on Llama-3.1-8B, AUROCs of 0.909, 0.946 and 0.918 at int3, $n = 1$, 100 and 1000). The clean test scores MEMIT alone with editor, quantizer and batch size fixed, a stratified bootstrap, and only cells with at least ten deployed failures and ten successes: the eleven such cells on the four main models have intervals spanning 0.5 in nine; the two that do not are int3 at $n = 1000$ on GPT-2-XL (0.597 $[0.510, 0.674]$) and GPT-J (0.636 $[0.550, 0.722]$), scored with higher ESNR meaning less likely to fail. This is the expected null where the failures are full-channel and almost entirely in the conditioning, which ESNR does not model; the two exceptions are the overloaded cells, where the weight channel alone costs 42% and 37% of the loss (29% and 26% by the even split; Table 11). The bin-width criterion scored in its own direction, pooled over editors, batch sizes and quantizers, has AUROC 0.409 on GPT-2-XL and 0.512 on Llama-3.2-3B, and 0.12–0.27 within quantizers on three models (Table 34); within a fixed editor, quantizer and batch size it cannot rank a batch's edits and is not a consistent predictor of single edits (Appendix L). What survives is what Proposition 1 asserts. ESNR predicts the readout error it is a theory of, in level (Appendix B) and in rank: within each of the 288 editor×quantizer×batch-size cells of the four main models, the Spearman correlation of $1/\mathrm{ESNR}$ with the measured readout error has median $+0.904$ and minimum $+0.534$ (MEMIT cells $+0.548$ to $+0.990$). Both sides are divided by $\|\Delta W k^*\|$, which could correlate them by itself; it does not drive this correlation: over MEMIT's 16 single-edit cells the rank correlation of the absolute predicted and measured errors is $+0.974$ to $+0.991$, and $+0.705$ to $+0.962$ after partialling out the residual norm $\|a\|$. It does not predict behaviour beyond trivial baselines. Pooled over the 12,000 rows of a model, where it mostly separates configurations, its AUROC for efficacy failure is 0.771–0.827; bit width with batch size as a tie-break scores 0.721–0.878 and the edit's own fp margin 0.721–0.912, and the in-sample failure rate of the row's (quantizer, batch size) cell, which sees nothing of the edit, 0.831–0.881, above $-\mathrm{ESNR}^2$ on every model (Table 54), so that number is not evidence for the predictor.

Table 54: Pooled AUROC for deployed efficacy failure over all rows of a model's grid (six editors, four quantizers, three batch sizes), for $-\mathrm{ESNR}^2$ and three baselines: bit width with batch size as tie-break, the edit's fp margin (lower = weaker), and the in-sample failure rate of the row's (quantizer, batch size) cell, an upper bound for any predictor that sees only the quantizer and batch size. Failures include edits that already fail unquantized; scores are oriented so that above 0.5 ranks failures first.

model	rows	failures	$-\mathrm{ESNR}^2$	bits, then n	fp margin	cell rate
GPT-2-XL	12,000	1044	0.771	0.721	0.900	0.831
Llama-3.2-3B	12,000	360	0.822	0.805	0.828	0.843
GPT-J-6B	12,000	467	0.827	0.781	0.912	0.869
Llama-3.1-8B	12,000	296	0.825	0.878	0.721	0.881

Y GPTQ ON THE EDITED MATRIX AND AS A BACKBONE

The edited matrix is quantized with GPTQ (Hessian $= C$) or round-to-nearest, backbone in fp; entries are the paired GPTQ$-$RTN readout-error difference as a fraction of the RTN baseline, negative

meaning GPTQ reproduces the edit better, bold where the 95% CI excludes zero (percentile intervals, approximate at these sizes; Appendix A), on 12 (GPT-2), 25 (GPT-2-XL) and 15 (the other three models) probes per cell (batch is $n = 150$ for GPT-2 and 1000 otherwise). GPT-2 (124M): $\mathbf{-14.8}\%$, $\mathbf{-21.4}\%$, $\mathbf{-31.2}\%$, $\mathbf{-29.4}\%$ ($n = 1$ int3, $n = 1$ int4, batch int3, batch int4); GPT-2-XL: $+1.2\%$ (n.s.), $\mathbf{-2.2}\%$, $\mathbf{-7.8}\%$, $\mathbf{-6.5}\%$; Llama-3.2-3B: $\mathbf{+19.4}\%$, $\mathbf{+10.0}\%$, $\mathbf{+6.5}\%$, $\mathbf{+6.1}\%$; Llama-3.1-8B: $\mathbf{+15.0}\%$, $\mathbf{+5.1}\%$, $+1.5\%$ (n.s.), $+0.7\%$ (n.s.); GPT-J-6B: $\mathbf{+6.7}\%$, -0.3% (n.s.), $\mathbf{-9.3}\%$, $\mathbf{-9.7}\%$. In this configuration (natural column order, no act-order) GPTQ never helps on either Llama, and at batch scale the split follows the MLP type (it helps the GPT-2 family and GPT-J, whose MLPs are not gated, hurts Llama-3.2-3B and leaves Llama-3.1-8B unchanged); it is not d_{in} (Llama-3.2-3B at 8192 is the worst cell and GPT-J at 16384 among the best) and at $n = 1$ not the size of the RTN error being compensated (the int3, $n = 1$ readout errors of this sweep's own RTN arm on the four main models are 0.059–0.082), though at batch scale the models GPTQ helps are also those with the larger RTN errors (0.24–0.92 at int3, against 0.14 and 0.18 on the Llamas). The outlier statistic of Appendix U separates the two groups after the fact and Qwen2.5-3B, the natural test of that account, is not in this sweep. All of this is the weight channel, which carries almost none of the deployed damage. The sweep's RTN arm independently recomputes the grid's readout error and agrees to 1.2% at $n = 1$ on GPT-2-XL.

A GPTQ backbone. Serving stacks apply GPTQ to every linear layer, and its error feedback changes the whole backbone an edit is conditioned on, so we built GPTQ backbones for GPT-2-XL and Llama-3.2-3B from the unedited model (`code/gptq_backbone.py`; the edit is solved and inserted afterwards, not re-calibrated): block by block as GPTQ does, each block calibrated on activations that have already passed through the quantized blocks below it (WikiText-103 train, 128×512 tokens, damping 1%, symmetric absmax groups of 64 as in the main grid, 3 and 4 bits, columns in their natural order with each group's scale set when the loop reaches it, i.e. without act-order or static groups); in every channel that quantizes the edited matrix it is GPTQ-quantized with its own second moment C as Hessian; QAE's share is taken against the MEMIT of its own run. Rebuilt with the same code and data in a separate perplexity run, the backbones' WikiText-103 test perplexity (99,840 tokens) is 21.10 and 22.93 at 4 and 3 bits on GPT-2-XL, against 20.80 unquantized, but 78.4 and 2,843 on Llama-3.2-3B, against 19.21: without act-order, GPTQ breaks Llama-3.2-3B, as Appendix E describes, so the Llama-3.2-3B rows describe a damaged model, and the served GPTQ of Appendix E, with act-order, is the evidence about GPTQ as deployed. In Table 55, MEMIT's conditioning share is 98.8–101.1% (even split 98.8–100.6%) in all eight cells and the weights-only channel is at most 0.05 nats, so under a Hessian-aware backbone the damage is again the conditioning. GPTQ's error feedback minimizes each layer's output error on calibration text, yet it moves the edited layer's key at the subject about as far as round-to-nearest does (MEMIT, $n = 1$, means over the 100 probes of Table 55, not tabulated: $\|\delta k\|/\|k^*\|$ 0.23 and 0.54 at 4 and 3 bits on GPT-2-XL against 0.27 and 0.59; 0.42 and 0.86 on Llama-3.2-3B against 0.42 and 0.83), and the edits lose 1.0–1.4 nats at 4 bits and 4.2–7.0 at 3. KAE returns 31–43% of MEMIT's deployed damage and QAE, solved on the same backbone, 95–105% at 3 bits on both models (75–111% at 4 bits, where the loss is small and the intervals wide), as on the round-to-nearest grid. A broken backbone is therefore not by itself what defeats QAE on Qwen2.5-3B's symmetric int3 model: on this one (perplexity 2,843) the key is displaced about as under round-to-nearest (mean $1/\mathrm{KSNR}$ 0.48 at GPTQ-3 in the same run against 0.44 at int3, Table 50), whereas on Qwen2.5-3B at int3 it is 1.01 and the edit is annihilated ($\rho = -0.014$, Appendix U).

Table 55: MEMIT on a GPTQ-quantized backbone (sequential GPTQ over every linear layer, WikiText-103 calibration 128×512 tokens, damping 1%, group 64; the edited matrix is GPTQ-quantized with its own Hessian C), 100 probes; fp is MEMIT's unquantized margin and the channel columns its change in nats (negative = margin lost). KAE and QAE recovered: share of MEMIT's deployed damage removed (KAE from the same run, QAE from the deployment-precision run on the same backbone).

model	quant	n	fp	weights-only	keys-only	full	K/F	even split	KAE rec.	QAE recovered
GPT-2-XL	GPTQ-3	1	13.21	−0.03 [−0.05, −0.00]	−4.20 [−4.78, −3.66]	−4.22 [−4.79, −3.68]	99.5%	99.4%	42%	105% [93, 118]
GPT-2-XL	GPTQ-3	100	13.13	−0.05 [−0.08, −0.01]	−4.22 [−4.78, −3.68]	−4.27 [−4.83, −3.73]	98.8%	98.8%	39%	99% [87, 112]
GPT-2-XL	GPTQ-4	1	13.21	−0.00 [−0.02, +0.01]	−0.99 [−1.22, −0.77]	−0.99 [−1.22, −0.77]	99.5%	99.6%	40%	86% [50, 129]
GPT-2-XL	GPTQ-4	100	13.13	−0.01 [−0.02, +0.01]	−1.00 [−1.23, −0.78]	−1.01 [−1.23, −0.79]	98.9%	99.0%	38%	75% [40, 115]
Llama-3.2-3B	GPTQ-3	1	13.85	−0.03 [−0.08, +0.03]	−7.01 [−7.74, −6.29]	−7.04 [−7.78, −6.32]	99.6%	99.6%	32%	100% [93, 108]
Llama-3.2-3B	GPTQ-3	100	13.55	+0.02 [−0.05, +0.08]	−6.75 [−7.46, −6.07]	−6.70 [−7.42, −6.00]	100.8%	100.5%	31%	95% [87, 104]
Llama-3.2-3B	GPTQ-4	1	13.85	−0.00 [−0.03, +0.03]	−1.41 [−1.74, −1.09]	−1.40 [−1.72, −1.08]	101.1%	100.6%	36%	111% [76, 147]
Llama-3.2-3B	GPTQ-4	100	13.55	−0.01 [−0.05, +0.03]	−1.34 [−1.65, −1.04]	−1.36 [−1.67, −1.05]	98.8%	99.0%	43%	111% [73, 151]

Z PREDICTIONS LOGGED BEFORE THE RUNS

The following predictions were written into the project's experiment log and committed to its version history before the corresponding runs were started: P1–P7 on 13 September 2026, before the first AWQ forward pass and the first ZsRE edit, and P8–P10 on 16 September 2026, before the first FT-L run (an internal, dated log, not an external registry; it will be made public with the repository upon acceptance). Each numbered item counts as one prediction; "passed" means the logged numeric threshold or direction held, "failed" that it did not, and "passed in part" that one logged part held and another did not; "passed as written" (P6) is a pass whose caveat is stated with it.

- **P1** AWQ cuts int3 damage on GPT-2-XL by at least 25%: *failed* (19.0%, 17.2% and 12.1% at $n = 1, 100, 1000$; the direction held in 21 of 24 MEMIT cells).
- **P2** Proposition 1 covers AWQ with no new parameter, its calibration ratio landing where its RTN partner's does: *passed* (four models, every batch size; at $n = 1000$ this means sharing RTN's miscalibration; Figure 3).
- **P3** AWQ's benefit is selective, not a uniform refinement: *passed* (step-map ratio $\Delta_{\mathrm{AWQ}}/\Delta_{\mathrm{RTN}}$ 0.990–1.020 at the median channel, 0.864–0.907 at the top-1% channels, Llama-3.1-8B's at one of its two exponents; Table 41).
- **P4** AWQ leaves KAE less to recover (direction only): *passed* (logged as a direction with no size or cell threshold, on the 12 cells of GPT-2-XL and Llama-3.2-3B then planned: positive in 10 of them; in 22 of all 24 cells, significantly in 21; Appendix Q).
- **P5** ZsRE conditioning share (logged as the "key share") above 0.8 on both models: *passed in part* (94.0–101.6% over the 12 ZsRE cells of the grid, Table 11); its qualitative half, that the weights-only damage be indistinguishable from zero, held in 8 of the 12 cells; in the other four it is at most 0.06 nats.
- **P6** Smaller absolute margin loss on ZsRE than on the matched CounterFact cell: *passed as written* (GPT-2-XL int3, $n = 1$: 5.52 vs 6.49 nats, Table 11), with the caveat that ZsRE's fp margin is also smaller (8.63 vs 12.84), so the full-channel margin kept is smaller (36% vs 49%, ratio of means).
- **P7** KAE's advantage reproduces in sign on ZsRE: *passed* (ΔES $+0.120$ and $+0.090$ at int3, $n = 1$; Table 37).
- **P8** FT-L's $|\cos(u, k^*)|$ below 0.3: *failed* (0.399, 0.380).
- **P9** FT-L's conditioning share below 90%: *failed* (97.0–100.6% at int3, $n = 1$; 95.0–100.6% over every FT-L cell read in Appendix R, 91.7–100.6% with GPT-2-XL's unread $n = 100$ cells).
- **P10** Proposition 1 calibrates on a dense update, inside the 0.90–1.00 band of MEMIT (the band logged on the cells run by then; all 32 MEMIT cells at $n \leq 100$ now span 0.895–1.017, five of them outside it): *passed* (FT-L at int3: 0.993 and 0.991 as mean per-edit ratios, Table 43; 0.993 and 0.990 as ratios of cell means; the registration did not say which average, and both are inside the band).

Two earlier predictions, written into the log before the first multi-layer run and before the Llama-3.1-8B generation run respectively, also failed and are reported in Appendix S (multi-layer spreading, logged as one prediction with three parts before the GPT-2 run, two wrong and one not supported there and on GPT-2-XL, and on Llama-3.2-3B two of the three held at $L = 5$) and Appendix W (that the surviving margin orders the behavioural damage across models). That makes twelve predictions: five failures (P1, P8, P9, the multi-layer prediction and the behavioural-ordering prediction), one partial pass (P5) and six passes; or fourteen with seven failures if the multi-layer prediction's three parts are counted separately. The passes are not equally informative: of the predictions with a numeric threshold (P1, P5, P8, P9, P10 and the behavioural-ordering prediction, whose Llama-3.1-8B band was -0.233 to -0.150), P10 passed, P5 passed in part and the other four failed; P2 set no numeric tolerance, P4, P6 and P7 are directions or signs, and P3 is a property AWQ's step map has by design. None of the paper's headline claims about the mechanism was logged in advance (the transfer claims of the first contribution are P2 and P10); they are confirmatory only in the sense that each was tested on models or datasets added after it was first observed. Analyses reported for completeness on which no positive claim rests: the pooled per-edit AUROC (Appendix X), the sub-bin variance form (Appendix B), the incoherent approximation to $\cos(u, \delta k)$ (Appendix U), and the GPT-2-only NF4 batching result (Appendix P).