CURVPROBE: DAMAGE HAS TWO FACTORS, AND THE CHECKPOINT SETS ONE

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training compression first has to decide *where*: which layers to delete, factorise or quantise. The criterion the field deploys ranks layers by displacement, how far an operation moves a layer's output, and never asks how much that move raises the loss. Expanding the loss where these operations inject their change splits the damage at a layer into that displacement times the loss's curvature along it. Second-order salience already estimates this curvature, one operation at a time. That curvature belongs to the frozen checkpoint, not to the operation, so it can be read once, before any operation is named. **CurvProbe** reads it as a **curvature profile**, one number per layer from $2L$ forward passes, with no backward pass and no Hessian. The factorisation predicts that displacement ranks layers below chance on checkpoints whose curvature is highest where displacement is lowest, as on Llama-3-8B, where the attention sublayer ranked safest to delete costs the most; the profile flags such failures before any damage is measured. One profile, never shown the operation it is judged on, repairs the damage ranking across eighteen checkpoints and eleven operations. On Llama-3-8B its layer sets keep that gain once a quarter of the layers are compressed. After low-rank approximation they hold 65.8% LAMBADA accuracy, against Block Influence's 55.8%. A checkpoint can therefore ship with its profile, and a compressor that arrives later ranks its layers with one forward pass where measuring their damage would take L.

1 INTRODUCTION

To make a pretrained transformer cheaper, a practitioner replaces a module with an approximation, such as a quantised weight or a deleted block (Katharopoulos et al., 2020; Wang et al., 2024; Frantar et al., 2023; Lin et al., 2024; Gromov et al., 2025; Men et al., 2025). The first decision is *where*. The prevailing criterion across post-training compression is displacement: change the modules that move least. Block Influence measures that movement as a change in hidden representations (Men et al., 2025), Gromov et al. (2025) as an angular distance, and AWQ (Lin et al., 2024) as an activation-weighted weight norm.

Displacement is what the operation does to the layer; damage is what the network then does with that displacement. Every one of these operations injects its change at the sublayer output, so expanding the loss there separates the two: the damage a substitution does at layer ℓ is how far it moves that layer's output, multiplied by how sharply the loss curves in the direction it moves (Proposition 1). The operation sets the first factor; the checkpoint sets the second. A criterion built on the first alone measures one factor of two, which is why it can rank layers below chance rather than merely imprecisely: it cannot tell a layer where the network absorbs a change from one where it amplifies it.

The gap is not theoretical. On Llama-3-8B the criterion names eight of the thirty-two attention sublayers as the safest to delete, and deleting those eight costs 5.4 nats against the intact model, within 1.01 nats of deleting every attention sublayer. Figure 1 shows the failure layer by layer. It is not peculiar to deletion: on Llama-3-8B and on Mistral-7B the criterion ranks worse than chance on all five operations we test, the quantisers included.

The second factor, the curvature, is a property of the frozen checkpoint at layer ℓ under a fixed loss and data distribution. An operation only sets the direction in which that curvature is taken. Averaging over directions gives $\kappa_\ell^{\mathrm{iso}} = \mathrm{tr}(H_\ell)/d$, one number per layer, and a symmetric second difference

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

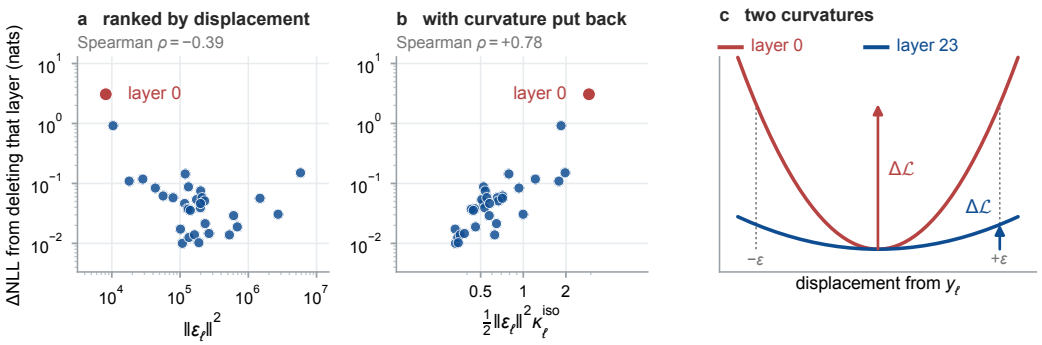


Figure 1: **One checkpoint, thirty-two layers, and why one move costs different amounts.** Llama-3-8B, one point per attention sublayer, the vertical axis the ΔNLL of deleting that sublayer alone. **(a)** Against displacement, the quantity the field ranks on. **(b)** Against the same displacement multiplied by the measured curvature. Layer 0 is the leftmost point of (a) and its highest: it moves least of the thirty-two and costs most. **(c)** The cause, schematic: every operation here acts at one site, and what it costs there is how far it moves the output times how sharply the loss curves — the two curves sharing an axis.

of the loss reads it in $2L$ forward passes of the unmodified model over the calibration set, with no backward pass, no Hessian formed and no operation chosen. We call the reading a **curvature profile**.

Second-order salience has estimated this curvature since optimal brain damage (LeCun et al., 1989), and HAWQ-V2 (Dong et al., 2020) reads its trace in the activations, but each such score is computed for the operation it serves, and neither explains why the displacement criteria the field deploys fail. Separating the two factors addresses both limits. It places the failure on the checkpoint: the checkpoint, more than the operation, decides where displacement fails, accounting for 57% of the variance in how well displacement ranks damage against the operation's 15%, and the profile forecasts those failures before any damage is measured. It also lets the curvature be read once, where every operation injects its change and before any operation is chosen. Whether that isotropic reading keeps enough to rank damage for operations it was not built for is an empirical question, and it is ours.

It does. One reading of the unmodified model repairs the ranking across eighteen checkpoints and eleven operations, and the base model's profile, applied unchanged to its instruction-tuned derivative, still ranks that model's damage at $\rho = +0.60$ on Llama-3-8B, against -0.21 for displacement. The nearest prior score does not carry over to operations it was not built for: on the three pruning operations, where HAWQ-V2's weight-space score is defined, it ranks damage at a mean $\rho = 0.005$ and loses to the profile on all 15 cells.

Throughout, each checkpoint's profile is read once and never shown the operation it is judged on, and results are reported as layer sets, the form in which a practitioner uses them. Our contributions are three.

- **A mechanism** (§4). Expanding the loss at the sublayer output shows why a displacement criterion can rank below chance: damage is a displacement, set by the operation, multiplied by a curvature, set by the checkpoint, and the criterion measures only the first. We prove when it ranks layers correctly, for any operation. Neither factor ranks damage alone: across 89 model–operation pairs each is negatively correlated with damage on 31, their product on one.
- **A prediction that holds** (§4, §5). The factorisation says where a displacement criterion drops below chance: on checkpoints whose curvature is highest where the displacement is lowest. It drops there, and so does every published salience score we can recompute, each worse than chance on some of the very checkpoints and operations it was built for. The profile flags those failures before any damage is measured: every cell where the magnitude rule is negatively correlated with the corrected score is below chance.
- **An instrument: CurvProbe** (§4–§7). It reads the curvature once per checkpoint, before any operation is chosen, and then ranks the layers for each operation with one more forward pass. That reading improves the magnitude rule's damage ranking on **121** of **128** model–

operation pairs, across eleven operations it was never shown, four of which, KV-cache and activation quantisation among them, touch no weight, so no weight-space score can even define them. When up to a quarter of the layers are compressed, its layer sets come within 0.03 nats of the sets chosen by *measuring* the damage, in three of the four settings where both were run (0.13 with the step set from the clean output alone). A second read, along each layer's clean output, needs no operation either and is exactly deletion's direction: it ranks deletion damage better than Block Influence even on the checkpoints where the profile alone falls behind it.

2 Related work

Second-order salience scores what to compress by loss curvature: per weight in optimal brain damage and optimal brain surgeon (LeCun et al., 1989; Hassibi & Stork, 1992), per structure by a weight-space Hessian trace (Yu et al., 2022), per layer by HAWQ-V2's Hutchinson trace in the activations (Dong et al., 2020). CurvProbe reads that trace forward-only, once per checkpoint, at the sublayer output every operation shares, and one reading serves eleven operations. Criteria that score a layer from its own weights and activations (Men et al., 2025; Gromov et al., 2025; Lin et al., 2024) measure displacement, never what the loss does with it; the alternative evaluates the model with each candidate applied (Kim et al., 2024; Song et al., 2024; Yuan et al., 2023; Zhong et al., 2025; Zhang et al., 2024b): direct measurement paid again for every operation. Lu et al. (2026) find that deleting the final quarter of the layers beats, on average, every layer-selection metric they test, perplexity included. The nearest recent objects each serve one setting: an output-space second-order score for MoE experts (Li et al., 2026), a forward-only KL sensitivity for hybrid SSMs (Kong et al., 2026), a layer-graph Hessian decomposition (Bolshim & Kugaevskikh, 2026). Appendix B covers layer merging (Yang et al., 2024b) and the rest.

3 Setup

A decoder-only transformer maintains a residual stream $h_0, \dots, h_L$ with $h_{t+1} = h_t + f_t(h_t)$. A **substitution** at layer ℓ replaces a module, changing that layer's attention output from y_ℓ to $y_\ell + \epsilon_\ell$; each operation below makes one such substitution at whichever layer it is applied to. All act at the same point, which is what makes them comparable, and §4 repeats the test at the MLP output. `linear` swaps softmax for ReLU linear attention on the same projections, `quant4`/`quant3` quantise those projections per-output-channel absmax, `prune` removes the attention sublayer, and `lowrank` truncates the output projection to 25% rank by SVD. Six further operations, registered before they were run, appear in Appendix D.2. **Damage** $D(\mathcal{R})$ is the increase in mean token NLL when the substitution is applied to layer set $\mathcal{R}$, and $a_\ell := D(\{\ell\})$.

Names used throughout. A *displacement criterion* ranks layers by how far the operation moves them; Block Influence, angular distance and AWQ's activation-weighted norm (reduced here to one number per layer) are the deployed ones. The *magnitude rule*, $\|\epsilon_\ell\|^2$ at the sublayer output, is the family's simplest member and our baseline throughout. The *profile* is $\kappa_\ell^{\mathrm{iso}}$, the curvature CurvProbe reads at the same point, and the *corrected score* is their product, $\frac{1}{2}\|\epsilon_\ell\|^2\kappa_\ell^{\mathrm{iso}}$, so it differs from the magnitude rule in one factor only. *Deepest-k* applies the operation to the last k layers. *Direct measurement* applies the operation to each layer in turn and records a_ℓ, the damage every score is judged against. The table that opens the appendix maps every number in the main text to the section that expands it and the file that holds it.

What is held fixed. The claim is about reuse, so the checkpoints have to be varied and everything else held still. We run the five operations on eighteen checkpoints, the *grid*, from Pythia-160M to LLaMA-13B and across both normalisers and a mixture-of-experts: Pythia-160M/410M/1.4B (Biderman et al., 2023), GPT-2-medium (Radford et al., 2019), GPT-J-6B (Wang & Komatsuzaki, 2021), Qwen2.5-0.5B/7B (Qwen Team, 2024), Qwen3-8B (Qwen Team, 2025), TinyLlama-1.1B (Zhang et al., 2024a), Llama-3-8B (Grattafiori et al., 2024), LLaMA-13B (Touvron et al., 2023), OLMoE-1B-7B (Muennighoff et al., 2025), OLMo-1-1B, Gemma-2-9B (Gemma Team, 2024), Mistral-7B-v0.3 (Jiang et al., 2023), Bloom-7B1 (BigScience Workshop, 2022), OPT-6.7B and OLMo-1-7B. Four more (OPT-1.3B, GPT-NeoX-20B, Yi-1.5-9B, Gemma-2-2B) have only the profile read, twenty-two checkpoints in all. Everything else is fixed: one corpus (WikiText-103 (Merity

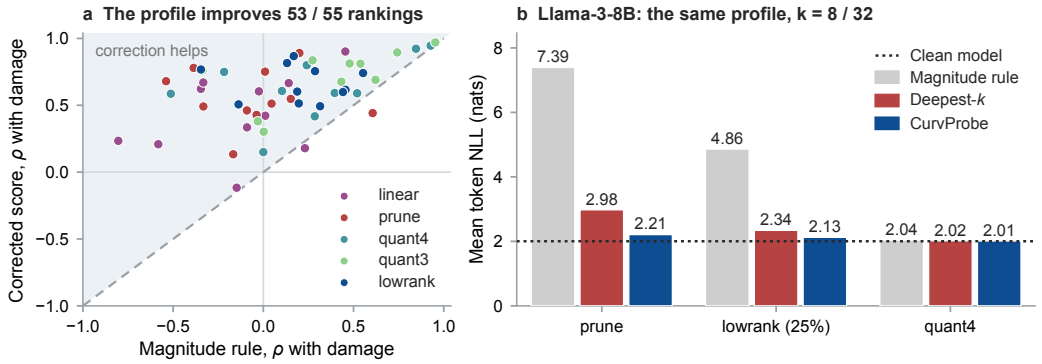


Figure 2: **One curvature profile, read before any operation is chosen, supplies the factor the magnitude rule is missing.** **(a)** Each point is one of the 55 core cells: the magnitude rule's rank correlation with true per-layer damage on the horizontal axis, the corrected score's on the vertical. Left of the vertical zero line the magnitude rule ranks layers worse than chance; 53 points lie above the equality line. **(b)** Mean token NLL after applying each operation to eight of Llama-3-8B's 32 attention sublayers; the dotted line is the unmodified model. The profile is reused across operations, while each operation's magnitudes can select a different set. These losses are on the WikiText-103 stream of §3, not Table 1's WikiText-2 protocol.

et al., 2017), held out, with C4 for the cross-corpus check), one budget (32 sequences of 1024 tokens per layer, up to 256 for selection), and fp32 for every curvature and per-layer damage measurement, because the curvature is a difference of nearly equal losses; each deployment test states its own precision.

How results are counted. A *cell* is one model–operation pair; its result is a rank correlation over a few dozen layers. Cells share checkpoints, so significance is claimed over checkpoints, not cells. Not every checkpoint has every operation, so each count names one of three nested sets: the *core*, eleven checkpoints with all five operations (55 cells); the *grid*, all eighteen (89); and *all cells*, the six further operations included (128). The loss after a layer set is compressed comes from one of two held-out streams: the WikiText-103 evaluation stream (Figure 2b) or the WikiText-2 protocol that Table 1 shares with its downstream tasks.

4 Damage is a magnitude times a curvature, and the curvature can be read first

Take the layer from the opening example. Deleting the attention sublayer at position 0 of Llama-3-8B costs 3.07 nats against 0.92 for the next worst, while moving the layer's output less than any of the other thirty-one (Figure 1a). Layer 0's output sits where the loss curves most sharply: its curvature is 162× that of layer 23, where deletion moves the output 20× more in $\|\epsilon_\ell\|^2$ yet costs less than a hundredth as much (Figure 1c). Figure 1b is the same thirty-two layers once that factor is in: the rank correlation with damage moves from -0.39 to $+0.78$.

That is the factorisation at work, and it is not confined to one layer. Asked for the eight attention sublayers of Llama-3-8B that are safest to delete, the magnitude rule names layers 0–6 and 19, nearly the entire bottom of the network, and gets **one of eight** right against the eight that measurement finds safest. The curvature profile, read from the same calibration articles and never shown the operation, gets **six of eight**; read along the deletion's own direction, it gets **all eight** (Table 1). The rest of the paper states that factorisation, measures it and puts it to use: we write the damage as two factors, show that one of them belongs to the checkpoint rather than the operation, and read it.

Proposition 1. *With all parameters fixed,* $a_\ell = \|g_\ell\|\|\epsilon_\ell\| \cos\theta_\ell + \frac{1}{2}\epsilon_\ell^\top H_\ell \epsilon_\ell + O(\|\epsilon_\ell\|^3)$*, so* $a_\ell = \|\epsilon_\ell\|^2 c_\ell$ *with an* effective coefficient c_ℓ *whose leading term is* $\frac{1}{2}\kappa_\ell$*. A monotone score* $\psi(\|\epsilon_\ell\|)$ *reproduces the ranking of* a_ℓ *only while* c_ℓ/c_m *stays below the squared ratio of the two magnitudes. That is a condition on the curvature alone exactly when the linear term and the remainder are negligible against it, which §7 measures rather than assumes. (Appendix C.1, Proposition 3.)*

The linear term does not rank damage. A first-order score reaches only $\rho = +0.06$ with damage at Pythia-410M. We drop the term not because of that correlation, but because the score built without it ranks damage across the full grid (§5). The same standard covers the third-order remainder: a ranking needs it to be order-preserving rather than small, so the factorisation survives even when a whole sublayer is removed (§7). Dropping both,

$$a_\ell \;\approx\; \tfrac{1}{2}\|\epsilon_\ell\|^2\,\kappa_\ell, \qquad \kappa_\ell = \hat{\epsilon}_\ell^\top H_\ell \hat{\epsilon}_\ell, \tag{1}$$

we are left with two factors that are measured at different times.

The magnitude depends on the operation but on nothing downstream of the layer, so one forward pass returns it at every layer at once, matching per-layer runs exactly on every operation tested. The curvature need not depend on the operation at all. Its expectation over random directions, $\kappa_\ell^{\mathrm{iso}} := \mathrm{tr}(H_\ell)/d$, is a property of the network at layer ℓ. What §5 tests is the transfer claim: that κ^{iso} stands in for the operation-dependent κ_ℓ well enough to rank damage.

Reading it with forward passes. Injecting $t\epsilon$ at the output of layer ℓ gives $\Delta\mathcal{L}(t) = At + Bt^2 + O(t^3)$. Symmetric offsets cancel the linear term, $\frac{1}{2t^2}[\Delta\mathcal{L}(t) + \Delta\mathcal{L}(-t)] = B + O(t^2)$, so $\kappa = 2B/\|\epsilon\|^2$. The probe direction is Gaussian; its length, the step of the difference, is set per layer from one fixed reference operation, linearisation, regardless of which operation is scored, and (i) below sets it from the clean output alone. Nothing else is required: no Hessian is formed, no backward graph is built, and the whole reading costs $2L$ forward passes per calibration batch. The estimator is Hutchinson's (Hutchinson, 1989) with the backward pass removed, the second difference that CR-SAM (Wu et al., 2024) takes in weight space, read here at the sublayer output. On the same layers, batches and probe vectors it agrees with backpropagated Hessian-vector products on two checkpoints ($\rho = 0.918$ and 0.993) at half the wall time and under half the peak memory (Appendix D.5). Where an operation's direction is known before it runs, the same difference can be taken along it. Deletion returns zero, so $\epsilon_\ell = -y_\ell$, and the read along the clean output, a second $2L$ passes with no operation applied, is the directional curvature itself: the *directional read*. Like the profile, it needs no operation and is read once. It is exact for deletion and lifts linearisation and low-rank approximation on nine of their ten cells (§7); the profile serves the quantisers, which round elementwise and leave no direction to read.

Three changes to the recipe leave the reading intact. **(i)** Setting the step from the clean output alone, with no operation involved, leaves the layers almost identically ordered ($\rho \geq 0.98$), and the sets it selects cost at most 0.15 nats more when compressed together (Appendix Table E.11). **(ii)** One probe per layer already gives split-half reliability 0.878. **(iii)** Moving the injection point from the attention output to the MLP output leaves the reading intact: the two sites' profiles have the same shape across layers, and at the MLP output the correction improves 23 of 25 cells, five operations on five checkpoints (Appendix E.2). The approximation holds at all three; §7 measures how far it reaches.

Which operations one profile can serve. For a substitution whose perturbation has second moment Σ at layer ℓ, Proposition 9 gives $\mathbb{E}[\kappa_\ell/\kappa_\ell^{\mathrm{iso}}] = d\ \mathrm{tr}(H_\ell\Sigma)/\,\mathrm{tr}(H_\ell)$. An operation is *curvature-generic*, $\kappa_\ell \approx \kappa_\ell^{\mathrm{iso}}$, when its perturbation is spread across directions. Rounding stays generic, at a median $\kappa/\kappa^{\mathrm{iso}}$ of 1.40 and 1.46 for the two quantisers. Removal does not, because it sets ϵ_ℓ to one learned vector: the median rises to 1.75 for deletion and 2.56 for linearisation. Proposition 3 turns the difference into a test a practitioner can run: a displacement score is faithful while $a_\ell/\|\epsilon_\ell\|^2$ stays flat across layers. The profile is therefore readable, forward-only, and operation-free. Whether it is *useful* is a claim about operations it has never seen, and §5 tests it on eleven of them.

5 One profile ranks layers for operations it never saw

What a practitioner needs from the profile is a layer set. Take the $k{=}L/4$ layers that direct measurement finds cheapest: the corrected score picks $\mathbf{0.60}$ of them on average across the grid, the magnitude rule 0.36. On the seventeen checkpoints where Block Influence is defined, the corrected score picks $\mathbf{0.63}$ and Block Influence 0.45 (sign test $p = 0.013$), although the two selections share fewer than half their layers.

The sets are chosen from the ranking, and Figure 3 tests that ranking against Eq. 1. The correction improves 53 of the 55 core cells (Figure 2a) and $\mathbf{121}$ of all $\mathbf{128}$ cells, and every one of the eighteen checkpoints gains ($p = 3.8 \times 10^{-6}$, the floor of the exact sign-flip test). It changes the sign, not just



Figure 3: **One profile, eleven operations it was never shown.** Each row is one operation, the six registered ones pooled into one row, running from the magnitude rule's mean Spearman correlation with true per-layer damage (hollow) to the corrected score's (filled). Two rows start left of zero, `prune` and `linear`, the two operations that replace the sublayer's function outright. Both cross into positive territory once the curvature factor is restored. Eleven models per row, five to eight for the registered six.

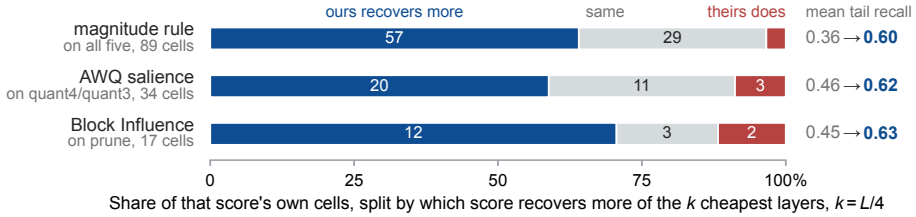


Figure 4: **The scores the field deploys, against ours on their own ground.** Tail recall at $k{=}L/4$ is the fraction of the k cheapest layers by measured damage that a score also puts in its own cheapest k, which is the set that gets deleted. Each row splits one published score's own cells three ways: ours recovers more of that tail, the two recover the same, the published score recovers more — the last on 3 of 89 for the magnitude rule, 3 of 34 for AWQ and 2 of 17 for Block Influence. A third of the magnitude rule's are ties, where both scores already name sets of equal value, so the means at the right understate the margin where the scores differ. Each row compares one score with ours; rows are not compared with one another, because the three scores cover different slices of the grid. One run per checkpoint of §3, recomputed from the released files.

the size: of the nineteen core cells where the magnitude rule is worse than chance, eighteen turn positive, the Llama-3-8B deletion of §4 among them.

Where the rule fails is decided by the checkpoint more than by the operation. Over the eighteen checkpoints of the grid, the checkpoint accounts for 57% of the variance in how well $\|\epsilon_\ell\|^2$ ranks damage and the operation for 15%. On Llama-3-8B and Mistral-7B the rule is worse than chance on all five operations, the quantisers included. Within a checkpoint the operation still matters: on deletion and linearisation, the two that replace the sublayer's function outright, the rule's ρ is on average 0.33 lower than on the other three, and lower on all but two of the eighteen checkpoints.

The profile forecasts the failure. It says where the rule fails before any damage is measured. The forecast is the magnitude rule's rank correlation with the corrected score, which needs no damage: it orders the eighteen checkpoints almost as the rule's measured failure does ($\rho = 0.83$), and each of the 14 cells where it is negative is below chance when measured (Appendix E.1).

This is not an artefact of any one score. Figure 4 sets every published salience score we can recompute against ours, on the operations each was built for, and counts the cells each side wins. The magnitude rule recovers fewer of the cheapest layers than ours on 57 of its 89 cells and more on 3; on 41 it does no better than k layers drawn at random. The blind spot is the family's, not one member's.

A blind spot in a ranking matters only if it shows when the layers are actually removed, and it does. The `prune` bars of Figure 2b continue the opening example: the profile's eight sublayers hold the loss near the intact model's, and the eight the magnitude rule names do not. GPT-J-6B, a structurally different checkpoint, repeats it, and low-rank approximation orders the selectors the same way. Quantisation leaves less to separate: on Llama-3-8B its per-layer damage spans 0.09 nats

against 3.1 for deletion, and the one reading helps most where displacement fails worst. Under three further seeds, the profile's margin over deepest-k varies by at most 0.22 nats across five checkpoints.

The result holds on held-out articles. The results above rank layers on the same corpus they were scored on. To separate the profile from the text it was fit on, we split WikiText-103 by article, so no article is used for both calibration and evaluation. We measure the curvature profile, the perturbation magnitudes, and Block Influence on the calibration articles alone, and use the held-out articles only to measure damage. On Llama-3-8B the profile lifts the magnitude rule's rank correlation from -0.37 to at least $+0.63$, and on Gemma-2-9B from -0.03 to at least $+0.49$; permuting the profile across layers takes the gain away (Appendix Table E.9). All three were among six predictions registered before the test, and all six held. The ordering also carries across corpora: read on WikiText and scored on a held-out C4 subset, it improves 39 of 40 cells.

The frozen profile also picks good sets. The frozen split's three calibration segments name different layers, and it does not matter. Deleting any of the three sets costs about 2.2 nats, against 2.98 for deepest-k and 7.39 for the magnitude rule. With the same frozen profile, the corrected score also correlates positively with low-rank damage on both checkpoints, where the magnitude rule's correlation is negative on both. Under a looser split that carries over only the profile, the corrected score still ranks deletion damage above the magnitude rule on all four checkpoints.

Four cheaper explanations, and why none of them fits. (i) Would any per-layer quantity do? No: permuting κ^{iso} across layers destroys the gain; the corrected score then improves 24 of the 55 core cells on average, against the measured profile's 53. **(ii)** Would depth do? No: the free depth prior $\exp(-c\,\ell/(L-1))$, calibrated leave-one-out, wins four checkpoints, by at most 0.08, and loses the other seven. Nor is it layer 0 alone, the costliest layer in 21 of the 55: dropping it lifts the magnitude rule's mean ρ only from 0.152 to 0.196, against the corrected score's 0.606. **(iii)** Would a scale correction do? No: dividing $\|\epsilon_\ell\|^2$ by each layer's residual-stream or attention-output RMS^2 reaches 0.396 and 0.266, against the corrected score's 0.627 on the 40 cells where all three are measured.

(iv) Would one factor do? No: over the 89 grid cells displacement alone and curvature alone are each negatively correlated with damage on 31, the product on 1, and the product beats whichever factor is better on 16 of the 18 checkpoints (sign test $p = 0.001$). The measured profile beats both substitutes a practitioner might reach for instead, depth and scale, and neither factor works alone.

What the profile adds over the nearest prior score. HAWQ-V2 (Dong et al., 2020) estimates the same trace in the activations, by backpropagated Hutchinson, and the forward read agrees with that estimate at half its wall time (§4). What the profile adds is the site and the reuse: read at the sublayer output, where every one of the eleven operations injects its change, the trace gives a corrected score that ranks damage on all of them. HAWQ-V2's weight-space score, the one it ranks layers with, has no object on four of the eleven, which displace no weights, and on three more it ranks damage at $\rho = 0.005$ against the profile's 0.639. On the quantisation it was built for it draws level with the profile only through a backward pass, and it falls behind on `quant3` and well behind on `lowrank`, 0.48 against 0.69 (Appendix D.11). One forward-only reading, taken before the operation is named, serves all eleven.

6 Deploying the chosen layers

§5 shows which layers CurvProbe picks; deployment asks what the model can still do once they are compressed. We take $k{=}L/4$, the budget of the layer sets in §5. There, removing eight of Llama-3-8B's thirty-two attention sublayers cuts the KV cache by $k/L = 25\%$ and decodes 10.5% faster at 4k (deletion, one-shot). A cache saving is a claim about long sequences, and the ordering holds when the evaluation is repeated on 16k-token sequences: the profile's cost moves by 0.005 nats. Every selector gets exactly this saving; what differs is whether the model still works (Table 1, full grid in Appendix H.4).

The strongest published selector. Block Influence (Men et al., 2025) is the published selector that comes closest, and it is the one to beat here. Over twenty-four layer-set tests on four checkpoints the corrected score is ahead on twenty-one and Block Influence on three, all `prune` (Appendix Table G.3), and on four further checkpoints it matches or beats Block Influence on deletion. End to end on Llama-3-8B the picture is the same (Table 1): the profile's set is ahead in loss and on LAMBADA, and the directional read widens the lead (LAMBADA paired $p < 10^{-30}$). On GSM8K

Table 1: **Same operation, same k, same measured cost, different selectors.** Llama-3-8B, deletion at k=8 of 32, one-shot, bf16, batch 1; LAMBADA is 2000 samples in the base column and 500 in the Instruct column, other tasks 500 throughout (200 GSM8K). Loss is mean token NLL (nats) under the WikiText-2 protocol. Instruct reuses the *base* layer sets, Block Influence's included. Every arm drops the same 25% of the KV cache; the two arms marked — under tok/s were not run through the 4k decode timer. Bold: best; underline: second. †Directional read (§4): exactly the layers direct measurement names. ‡Separate run (clean 0.740), not ranked.

		cost at 4k		base			Instruct	
arm	layers deleted	KV MiB	tok/s	NLL	LAMB.	GSM8K	LAMB.	GSM8K
clean	—	512.0	41.65	2.51	0.686	0.450	0.652	0.715
CurvProbe, isotropic	11, 18–20, 22–25	384.0	46.01	<u>2.73</u>	<u>0.511</u>	0.240	0.444	0.505
CurvProbe, directional†	19, 20, 22–26, 28	384.0	—	**2.66**	**0.601**	**0.405**	**0.540**	0.665‡
Block Influence	22–29	384.0	—	2.79	0.479	<u>0.370</u>	<u>0.462</u>	**0.650**
deepest-k	24–31	384.0	45.95	3.37	0.350	0.220	0.332	<u>0.510</u>
free depth prior	0–2, 19, 20, 22–24	384.0	46.03	6.55	0.045	0.055	0.044	0.035

the directional set draws level with Block Influence ($p = 0.39$). On Qwen2.5-7B no selector keeps GSM8K: every arm scores at most 2.0% of the clean model's accuracy.

The score holds with the real compressors. A practitioner will not run GPTQ L times to decide where to run GPTQ, so the score is computed under a cheap proxy and the operation performed with the real thing. We registered the claim twice before running it, that the corrected score is never worse than the free depth prior by more than 0.6 nats, and it held both times. Scored under round-to-nearest and quantised with GPTQ and AWQ it is at-or-better than deepest-k in all twelve settings. Scored under a naive 2:4 mask and pruned with a compensated 2:4 it holds across sixteen sparse settings (Appendices H.2, H.3). Under the real quantisers every arm, the free depth prior included, lands in one narrow band, as the factorisation predicts: GPTQ and AWQ minimise the displacement itself, and with it the stake of the choice.

A second operation with downstream consequences. The profile was read before any operation was named, so the test that matters is whether it serves a second. For low-rank approximation on Llama-3-8B, the profile's layer set gives 65.8% LAMBADA accuracy, against 57.2% for deepest-k, 55.8% for Block Influence and 6.4% for the magnitude rule on the same 500 examples. Paired over those items the margin over deepest-k is 8.6 points (95% CI $[5.0, 12.2]$), and it is already there at $L/8$. Measuring the damage sets the ceiling, and the profile reaches it at one pass where measuring takes L: the two arms sit a hundredth of a nat apart in mean token NLL, no paired interval separates them on three of the four tasks, and the fourth lies in the profile's favour (Appendix Table E.12).

The selection transfers to the model people serve. Nothing is re-derived: the base profile's eight layers are deleted from the Instruct variant, and the ordering survives. It beats the free depth prior by the same margin it did on base, and beats deepest-k on LAMBADA and ARC-Easy while matching it on MMLU, HellaSwag and GSM8K. Block Influence, on the same base layer sets, is the harder comparison. Across five Instruct tasks the isotropic set and Block Influence are separated on two: the isotropic set leads on ARC-Easy (0.641 against 0.568) and Block Influence on GSM8K ($p = 2 \times 10^{-4}$). On LAMBADA the two are level, and the directional set leads Block Influence ($p = 1.3 \times 10^{-5}$). GSM8K is the task §7 traces to the isotropic read's layers. What carries across the fine-tune is the order of the layers, not the size of the damage, and it carries beyond this one layer set: applied unchanged to the Instruct derivative, the base profile ranks the derivative's per-layer damage at $\rho = +0.60$ on Llama-3-8B and $+0.35$ on Qwen2.5-7B, against -0.21 and $+0.08$ for displacement alone (Appendix H.7).

7 THE OPERATING ENVELOPE

The factorisation of §4 scores one layer at a time, and CurvProbe reads the isotropic average of its curvature. Each premise sets a range; each range is measured here, together with what takes over at its edge.

Where deleted layers interact, search takes over. A per-layer score is enough while removals add. At $k{=}L/4$ they do, and the profile's set beats deepest-k in eight of the twelve downstream deletion settings; at $k{=}L/2$, in two. The interaction is measured: on Qwen2.5-7B and Gemma-2-9B, deleting direct measurement's own set costs 18–$20\times$ the sum of its single-layer damages, and Block Influence's contiguous blocks beat every per-layer selector, direct measurement included. There search pays for its cost. SLEB (Song et al., 2024) re-scores after every removal and wins all four settings we ran it in, at 129 to 301 evaluations; a greedy search of the same kind, restricted to $2k$ candidate layers, beats Block Influence on all five checkpoints we ran it on, by 0.35 to 1.03 nats, at 41–47% of a full search's evaluations. Within the range, taking the k individually cheapest layers is not the weak link: with per-layer damage measured, it picks the best triple when every triple is scored.

What reuse buys, and what it costs. Direct measurement is the quantity the profile approximates: it orders Qwen2.5-7B's layers at $\rho = 0.975$ where the profile reaches 0.569 (Appendix D.9). It pays for that in passes, running each operation through the network once per layer, while one pass returns the displacement at every layer. The profile must be read in fp32: in bf16 the read correlates with the released profile at $\rho = -0.56$. Against measurement in deployment's bf16, the profile route is ahead from the eighth operation: all eleven cost 744 s against 1097 s on Qwen2.5-7B. Once the profile, or the directional read, ships with the checkpoint, each costs $28\times$ less (Appendix D.10). What reuse gives up, with the chosen layers compressed together, is at most 0.03 nats on three of four tests and 0.66 on the fourth, Gemma-2-9B deletion; with the step set from the clean output alone, 0.13 and 0.81 (Appendix Table E.11).

On GSM8K the directional read keeps what the isotropic read loses. The profile is built for next-token loss, and the GSM8K column of Table 1 is where its two reads part. The next-token objective is not the cause: the directional set, chosen under the same loss, keeps the reasoning the isotropic set loses, ahead of it in the same run ($p = 3 \times 10^{-5}$). The difference sits in two layers, 11 and 18, that only the isotropic read picks.

What remains is isotropy, and the directional read targets it. The correction substitutes an isotropic curvature for the one along the operation's actual direction. With the per-layer damage written as

$$a_{\ell,o} - \widehat{a}_{\ell,o} = g_\ell^\top \epsilon_{\ell,o} + \tfrac{1}{2}\|\epsilon_{\ell,o}\|^2 \left(\kappa_{\ell,o}^{\mathrm{dir}} - \widehat{\kappa}_\ell^{\mathrm{iso}}\right) + R_{\ell,o}, \tag{2}$$

the linear term dominates in at most eleven layers across the corpus (Appendix C.2). The residual beyond the fitted linear and quadratic terms concentrates on the two operations that replace the sublayer's function (median $|R_3|/a_\ell$ 0.22 and 0.26, against at most 0.04 for the quantisers), where §4 asks it only to preserve order and §5 finds that it does. The directional term is the large one: the measured κ_ℓ in place of κ^{iso} is worth 0.235 in mean ρ where both are defined. Isotropy, more than the second-order truncation, limits the correction. Where the operation moves the sublayer along a fixed direction, the directional read of §4 recovers it exactly on `prune` and lifts ρ at all five checkpoints on `linear` and at four on `lowrank`, while the quantisers leave no direction to recover. It lifts the two removal operations' mean ρ from 0.492 to 0.782 and leaves no cell below chance (Appendix F.10). It wins where the profile is weakest: on the five checkpoints where Block Influence out-ranks the profile on deletion, the directional read out-ranks Block Influence on all five.

8 Conclusion

Where to compress is a question about two factors, and the criteria the field deploys answer it with one. The missing factor, the curvature, is why the layer that moves least can cost most. It is a property of the checkpoint, so it can be read before the operation exists, and one reading serves operations it never saw. Read first, it also says whether a displacement criterion already in use will fail on the checkpoint in hand (§5). It pays wherever more than one operation is weighed: a pipeline screening candidates, a sweep over ranks or bit-widths, a checkpoint released with its profile (Appendix H.7), and a spot check on a few layers that says when to trust the profile (Appendix G.11). A later compressor can then start from a set fixed before its operation existed.

REPRODUCIBILITY STATEMENT

Data and splits. All experiments use publicly available checkpoints and WikiText-103, with a held-out C4 subset for the cross-corpus check. For the article-held-out test we split WikiText-103 by article, so no article is used for both calibration and evaluation, and a released manifest records the split; the curvature profile, the perturbation magnitudes and Block Influence are all measured on the calibration articles alone, with the held-out articles supplying damage only. End-to-end NLL numbers use one fixed evaluation stream, the second half of the WikiText-103 validation split tokenised without a document-start token. It supplies 123 sequences of 1024 tokens (the pipeline requests 200; the split supplies 123). Table 1 and the appendix tables that share its protocol report WikiText-2 NLL instead.

What is released. The measurement, selection and analysis code and the raw JSON behind every number are in the supplementary material. The table opening the appendix maps every main-text claim to the appendix section that expands it and the file that holds it, and fifty-one of the read-only checking scripts in `analysis/` re-derive the counts they check from those files with no arguments; `results/MANIFEST.md` names those and the seven that need an argument, a companion run, or a model file. Every checkpoint in Appendix Figure I.1 ships the per-layer curvature profile its rate was fitted to, so each $\hat{c}$ can be refitted from the released file rather than taken on trust. Nothing in Figure 1 is a new measurement: both correlations are the Llama-3-8B `prune` entries of Appendix Table E.2, re-derived by `analysis/verify_layer0.py`.

What the code checks on itself. A numerical self-test compares chunk-parallel linear attention against a naive recurrent reference (relative error 2×10^{-7}), and an adapter round-trip checks, for each of the seven distinct attention-reconstruction paths, that it reproduces its model's own attention output; that check caught a rotary error in the GPT-J adapter; the error is reported and corrected in Appendix D.3. The two stage-two files for GPT-J-6B record different identifier strings for the same checkpoint (a local mirror path and the Hub id); the mirror's weights and `config.json` are byte-identical to the Hub copy (SHA-256, checked by the released `verify_gptj_hash.py`, which runs on the box holding the mirror).

Numerical conditions. Seeds control batch order, alignment sampling and initialisation (0, 1, 2). Curvature is measured in fp32, because B is extracted from a sum of nearly equal losses and cancels in bf16; efficiency measurements are in bf16, the deployment configuration. Proofs are in Appendix C.1 onward.

How many runs each number is. The tables that train or re-evaluate say so. The linearisation sweep, the deployment baselines and the B2 re-adjudication are means over three seeds, with the within-run standard deviation in parentheses where there is room for it, and the probe-seed stability table re-runs selection under four. The measurement grids are one run per cell. Re-deriving a cell from the released files is exact, and that is what the checking scripts in `analysis/` do. Re-measuring it on a different box need not be. Where a re-measurement moved a published cell we report the move rather than quietly replacing it (Appendix E.1).

What was registered before it was run. Every registered prediction in this paper is committed in the released registration files with its falsification criterion before the corresponding run, and is reported as it came out (Appendix A.1). The directional endpoint of §6 is Llama-3-8B's; its registered repeat on OLMoE-1B-7B improves on the isotropic set but not on Block Influence (Appendix F.10).

STATEMENT ON THE USE OF AI TOOLS

A large language model (Claude, Anthropic) was used as a research assistant throughout this work: it assisted in writing and debugging the experimental scripts, polishing the manuscript, and searching the literature. The author designed the research programme, formulated the claims and predictions, ran all experiments, verified every reported number against the raw outputs, checked every reference against the published record, and takes full responsibility for the contents of this submission.

ETHICS AND BROADER IMPACT

The measurements in this paper use public checkpoints and public corpora (WikiText-103, C4, LAMBADA, HellaSwag, ARC-Easy, MMLU, GSM8K). No human subjects, private data or newly

collected text are involved, and no model is trained from scratch: the only training is the short alignment and distillation runs of Appendices D.4 and H.5.

The social risk of the method is the risk of compression generally. Cheaper inference for a fixed checkpoint also lowers the cost of deploying it, and a layer-selection rule cannot tell a benign deployment from a harmful one; the same criterion that removes eight attention sublayers to fit a device also removes them from a model we would rather saw in half. Against that, the claim here is bounded by measurement. The method chooses *which* layers to remove for an operation that has already been chosen, and its value is the value of that choice (§7). It does not make a model servable that could not otherwise be served; it makes the already-available choice of layers a measured one rather than a guessed one.

What the work does spend is compute, and we would rather state it than imply it. Reading one profile per checkpoint is $2L$ forward passes per calibration batch, read once and reused across the eleven operations of §5; each operation then adds one pass, which returns its displacement at every layer. The alternative, measuring each operation's per-layer damage directly, is L forward passes per operation, the cheaper route only for a single operation on a checkpoint whose profile has not been read. The evaluation budgets behind every number are in Appendix D.1, and the per-operation wall clock on the one checkpoint where we timed it is in Appendix D.10.

REFERENCES

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. SliceGPT: Compress large language models by deleting rows and columns. In *International Conference on Learning Representations (ICLR)*, 2024.

Stella Biderman, Hailey Schoelkopf, Quentin Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, Aviya Skowron, Lintang Sutawika, and Oskar van der Wal. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning (ICML)*, 2023.

BigScience Workshop. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint*, 2022.

Johann Birnick. The lattice geometry of neural network quantization: A short equivalence proof of GPTQ and babai's algorithm. In *International Conference on Learning Representations (ICLR)*, 2026.

Maxim Bolshim and Alexander Kugaevskikh. Inter-layer Hessian analysis of neural networks with DAG architectures. *arXiv preprint arXiv:2604.11639*, 2026. URL `https://arxiv.org/abs/2604.11639`.

Xiaodong Chen, Yuxuan Hu, Jing Zhang, Yanling Wang, Cuiping Li, and Hong Chen. Streamlining redundant layers to compress large language models. In *International Conference on Learning Representations (ICLR)*, 2025.

Krzysztof Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamás Sarlós, Peter Hawkins, Jared Davis, Afroz Mohiuddin, Lukasz Kaiser, David Belanger, Lucy Colwell, and Adrian Weller. Rethinking attention with performers. In *International Conference on Learning Representations (ICLR)*, 2021.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. HAWQ: Hessian aware quantization of neural networks with mixed-precision. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Zhen Dong, Zhewei Yao, Daiyaan Arfeen, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. HAWQ-V2: Hessian aware trace-weighted quantization of neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

Elias Frantar and Dan Alistarh. SparseGPT: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning (ICML)*, 2023.

Elias Frantar, Sidak Pal Singh, and Dan Alistarh. Optimal brain compression: A framework for accurate post-training quantization and pruning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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 (ICLR)*, 2023.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. Model tells you what to discard: Adaptive KV cache compression for LLMs. In *International Conference on Learning Representations (ICLR)*, 2024.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. https://arxiv.org/abs/2407.21783, 2024.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. The unreasonable ineffectiveness of the deeper layers. In *International Conference on Learning Representations (ICLR)*, 2025.

Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *Conference on Language Modeling (COLM)*, 2024.

Babak Hassibi and David G. Stork. Second order derivatives for network pruning: Optimal brain surgeon. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1992.

Weizhong Huang, Yuxin Zhang, Xiawu Zheng, Fei Chao, and Rongrong Ji. Determining layer-wise sparsity for large language models through a theoretical perspective. In *International Conference on Machine Learning (ICML)*, 2025. URL https://openreview.net/forum?id=otNB7BzsiR.

Michael F. Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics – Simulation and Computation*, 18(3):1059–1076, 1989.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint*, 2023.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning (ICML)*, 2020.

Bo-Kyeong Kim, Geonmin Kim, Tae-Ho Kim, Thibault Castells, Shinkook Choi, Junho Shin, and Hyoung-Kyu Song. Shortened LLaMA: Depth pruning for large language models with comparison of retraining methods. *arXiv preprint arXiv:2402.02834*, 2024.

Jason Kong, Nilesh Prasad Pandey, Flavio Ponzina, and Tajana Rosing. A KL lens on quantization: Fast, forward-only sensitivity for mixed-precision SSM-transformer models. *arXiv preprint arXiv:2604.13440*, 2026. URL https://arxiv.org/abs/2604.13440.

Tanishq Kumar, Zachary Ankner, Benjamin F. Spector, Blake Bordelon, Niklas Muennighoff, Mansheej Paul, Cengiz Pehlevan, Christopher Ré, and Aditi Raghunathan. Scaling laws for precision. In *International Conference on Learning Representations (ICLR)*, 2025.

Eldar Kurtic, Elias Frantar, and Dan Alistarh. ZipLM: Inference-aware structured pruning of language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Disen Lan, Jianbin Zheng, Yuxi Ren, Xin Xia, Xuanda Wang, Xuefeng Xiao, Xipeng Qiu, and Yu Cheng. Morphing into hybrid attention models. *arXiv preprint*, 2026.

Yann LeCun, John S. Denker, and Sara A. Solla. Optimal brain damage. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1989.

Ke Li, Zheng Yang, Zhongbin Zhou, Feng Xue, Zhonglin Jiang, and Wenxiao Wang. HEAPr: Hessian-based efficient atomic expert pruning in output space. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=JAbMgS7gl6.

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, Omri Abend, Raz Alon, Tomer Asida, Amir Bergman, Roman Glozman, Michael Gokhman, Avashalom Manevich, Nir Ratner, Noam Rozen, Erez Shwartz, Mor Zusman, and Yoav Shoham. Jamba: A hybrid transformer-mamba language model. *arXiv preprint*, 2024.

Chi-Heng Lin, Shangqian Gao, James Seale Smith, Abhishek Patel, Shikhar Tuli, Yilin Shen, Hongxia Jin, and Yen-Chang Hsu. Modegpt: Modular decomposition for large language model compression. In *International Conference on Learning Representations (ICLR)*, 2025.

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 (MLSys)*, 2024.

Hong Liu, Zhiyuan Li, David Hall, Percy Liang, and Tengyu Ma. Sophia: A scalable stochastic second-order optimizer for language model pre-training. In *International Conference on Learning Representations (ICLR)*, 2024.

Yao Lu, Hao Cheng, Yujie Fang, Zeyu Wang, Jiaheng Wei, Dongwei Xu, Qi Xuan, and Zhaowei Zhu. Reassessing layer pruning in LLMs: New insights and methods. In *International Conference on Learning Representations (ICLR)*, 2026. URL https://openreview.net/forum?id=04Tfwy3LLC.

Ryan Lucas and Rahul Mazumder. Preserving deep representations in one-shot pruning: A hessian-free second-order optimization framework. In *International Conference on Learning Representations (ICLR)*, 2025.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. LLM-pruner: On the structural pruning of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Sadhika Malladi, Tianyu Gao, Eshaan Nichani, Alex Damian, Jason D. Lee, Danqi Chen, and Sanjeev Arora. Fine-tuning language models with just forward passes. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International Conference on Machine Learning (ICML)*, 2015.

Gabriel Mel. Beyond spectra: Eigenvector overlaps in loss geometry. In *International Conference on Learning Representations (ICLR)*, 2026.

Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in large language models are more redundant than you expect. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20192–20204, 2025.

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

Niklas Muennighoff, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Jacob Morrison, Sewon Min, Weijia Shi, Pete Walsh, Oyvind Tafjord, Nathan Lambert, et al. OLMoE: Open mixture-of-experts language models. In *International Conference on Learning Representations (ICLR)*, 2025.

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

Qwen Team. Qwen3 technical report. *arXiv preprint*, 2025.

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

Liliang Ren, Yang Liu, Yadong Lu, Yelong Shen, Chen Liang, and Weizhu Chen. Samba: Simple hybrid state space models for efficient unlimited context language modeling. In *International Conference on Learning Representations (ICLR)*, 2025.

Jiwon Song, Kyungseok Oh, Taesu Kim, Hyungjun Kim, Yulhwa Kim, and Jae-Joon Kim. SLEB: Streamlining LLMs through redundancy verification and elimination of transformer blocks. In *International Conference on Machine Learning (ICML)*, 2024. arXiv:2402.09025.

Ke Sun. Deterministic bounds and random estimates of metric tensors on neuromanifolds. In *International Conference on Learning Representations (ICLR)*, 2026.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Wenfang Sun, Xinyuan Song, Pengxiang Li, Lu Yin, Yefeng Zheng, and Shiwei Liu. The curse of depth in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint*, 2023.

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

Junxiong Wang, Daniele Paliotta, Avner May, Alexander M. Rush, and Tri Dao. The mamba in the llama: Distilling and accelerating hybrid models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Erik Wijmans, Brody Huval, Alexander Hertzberg, Vladlen Koltun, and Philipp Krähenbühl. Cut your losses in large-vocabulary language models. In *International Conference on Learning Representations (ICLR)*, 2025.

Tao Wu, Tie Luo, and Donald C. Wunsch, II. CR-SAM: Curvature regularized sharpness-aware minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 2024.

Songlin Yang, Bailin Wang, Yikang Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. In *International Conference on Machine Learning (ICML)*, 2024a.

Yifei Yang, Zouying Cao, and Hai Zhao. LaCo: Large language model pruning via layer collapse. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 6401–6417, 2024b. doi: 10.18653/v1/2024.findings-emnlp.372. URL https://aclanthology.org/2024.findings-emnlp.372/.

Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Gen Li, Ajay Jaiswal, Mykola Pechenizkiy, Yi Liang, Michael Bendersky, Zhangyang Wang, and Shiwei Liu. Outlier weighed layerwise sparsity (OWL): A missing secret sauce for pruning llms to high sparsity. In *International Conference on Machine Learning (ICML)*, 2024.

Shixing Yu, Zhewei Yao, Amir Gholami, Zhen Dong, Sehoon Kim, Michael W. Mahoney, and Kurt Keutzer. Hessian-aware pruning and optimal neural implant. In *IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2022.

Zhihang Yuan, Yuzhang Shang, Yue Song, Dawei Yang, Qiang Wu, Yan Yan, and Guangyu Sun. ASVD: Activation-aware singular value decomposition for compressing large language models. *arXiv preprint arXiv:2312.05821*, 2023. URL https://arxiv.org/abs/2312.05821.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. TinyLlama: An open-source small language model. https://arxiv.org/abs/2401.02385, 2024a.

Yang Zhang, Yawei Li, Xinpeng Wang, Qianli Shen, Barbara Plank, Bernd Bischl, Mina Rezaei, and Kenji Kawaguchi. Finercut: Finer-grained interpretable layer pruning for large language models. *arXiv preprint arXiv:2405.18218*, 2024b. URL https://arxiv.org/abs/2405.18218.

Longguang Zhong, Fanqi Wan, Ruijun Chen, Xiaojun Quan, and Liangzhi Li. BlockPruner: Fine-grained pruning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 5065–5080, 2025.

Zhongzhu Zhou, Qingyang Wu, Junxiong Wang, Mayank Mishra, Shuaiwen Leon Song, Ben Athiwaratkun, and Chenfeng Xu. Taylor-calibrate: Principled initialization for hybrid linear attention distillation. *arXiv preprint*, 2026.

WHERE EACH CLAIM IS SUPPORTED

Every number in the main text is produced by a released script and stored in a released JSON file. This table maps each main-text claim to the appendix section that expands it and to the file that holds it. Paths are relative to `results/` in the supplementary material (the submission copy of the `gpu_experiments/results/` tree, kept in sync at release).

main-text claim	appendix	source
§4: the measurement		
Proposition 1; first-order term at chance	C.1	`score_*.json`; `analysis/verify_firstorder.py`
Symmetric differencing; the R_3 remainder	C.2	`transfer_*.json`
Estimator vs. backprop HVP, $\rho = 0.918$	D.5	`hvp_410m.json`, `hvp_qwen05.json`
Probe budget, split-half reliability	D.6	`probebudget_*.json`
Norm invariance of the profile	D.7	`substitutions_*.json`
Anisotropy of H_ℓ beyond the scalar	F.9	`metric_lowrank_410m.json`
Wall-clock and pass counts	D.10	`wallclock_410m.json`
Definitions of the six further operations	D.2	`newsubs_*.json`, `moresubs_*.json`
§5: damage prediction		
Table E.7, per-cell 11×5 grid	E.1	`analysis/permutation_test.json`
Where the magnitude rule fails, and the profile's forecast of it (57% against 15%; $\rho = 0.83$; 14 of 14)	E.1	`analysis/verify_failure_forecast.py`
Extended grid, 121 of 128 cells, eighteen checkpoints	E.1	`analysis/verify_extended_grid.py`; `analysis/extended_grid.json`
Tail recall and Jaccard at $k{=}L/4$	F.1	`analysis/tail_metrics.py`; `analysis/tail_metrics.json`
Profile controls: permuted, reversed, free priors	E.1	`profile_ablation_9model.json`
Layer-scale competitor, 40 cells	E.7	`profile_ablation_8model.json`; `stream_norms_*.json`
Curvature against residual-stream norm, eight checkpoints	E.7	`stream_norms_*.json`; `substitutions_*.json`; `analysis/verify_curvature_vs_norm.py`
Sequence-budget accuracy (0.966 / 0.569)	D.9	`p1b_410m.json`, `p1b_qwen7b.json`
MLP-site transfer, 23/25 cells	E.2	`mlp_permutation.json`
Each field's own published score	E.9	`realbaselines_*.json`
Two-factor ablation: each factor alone against the product, 89 core cells, one run per checkpoint, Table E.4	§5	`substitutions_*.json`; `analysis/verify_twofactor.py`
Block Influence scoreboard, five Instruct tasks, paired McNemar	§6	`downstream_llama38b_instruct{,_bi}_k8.json`; `gsm8k_llama38b_instruct{,_bi}_k8.json`; `analysis/verify_bi_scoreboard.py`
Profile shipped with a checkpoint, priced on its instruction-tuned derivative	H.7	`transfer_{llama38b,llama38b_recheck,llama38b_instruct,qwen7b,qwen7b_instruct}.json`; `substitutions_{llama38b,qwen7b}_instruct_{base,own}profile.json`; `analysis/verify_profileship.py`
Field-wide: every published score on its own operation, one run per checkpoint, eighteen checkpoints (Figure 4)	E.9	`substitutions_*.json` (read-only); `../make_field_house.py`; `analysis/verify_fieldscores.py` re-derives every number in the figure and in Table E.18
The layer-0 worked example, Llama-3-8B `prune` (Figure 1)	E.1	`substitutions_llama38b.json`, `transfer_llama38b.json` (read-only); `../make_fig_layer0.py`; `analysis/verify_layer0.py` re-derives every number the paragraph and the figure state
Article-held-out test, both checkpoints, and the permuted control	E.5	`p1a_strict_{llama38b,gemma29b}.*.json`
Held-out sets deleted together; direct measurement's sets (0.03 and 0.66 nats)	E.5	`p1a_joint_{llama38b,gemma29b}_prune.json`; `p1a_lowrank_{llama38b,gemma29b}/p1a_joint_*_lowrank.json`
Low-rank downstream tasks against direct measurement and Block Influence	E.5	`e2e_llama38b_lowrank_k8.json`, `e2e_llama38b_lowrank_k8_bi.json`
Cross-corpus: profile on WikiText, damage on C4, 40 cells	E.1	`substitutions_*_c4.json`; `analysis/verify_crosscorpus.py` also re-derives the Qwen2.5-7B GSM8K collapse of §7

Curvature-generic split, nine checkpoints	F.1	`aniso_summary_9model.json`; `73_family_predictor.py`; `analysis/verify_generic.py` re-derives the per-operation medians, the family spreads and Table F.2's two counts
Figure 2, both panels	E.1	`substitutions_*.json` (11×5 core); `e2e_llama38b_*.json` (the four arms); `analysis/verify_fig1.py` re-derives the $55/53/19$ counts and all nine bars
HAWQ-V2 weight-space baseline; its activation route (0.48 against 0.69 on low-rank)	E.10, D.11	`36_weightspace_baseline.py`; `weightspace_{410m,1.4b}_dw.json` (corrected $\|\Delta W\|^2$); `weightspace_*.json`
Injection vs. output, $\rho = -0.050$ vs. $+0.987$	E.8	`propagation_410m.json`
Split-half reliability of a_ℓ	D.8	`substitutions_*.json`
§6: selection		
$28/35$ scoreboard (pre-Qwen3-8B)	G.2	`analysis/reproduce_35settings.py`
Deployment baselines, three seeds	G.2	`scoreboard_*.json`
Noise-aware full-35 recount	G.2	`analysis/b3_full35_rejudication.py`
Profile vs. depth prior overlap	G.2	`analysis/free_prior_divergence.py`
Real GPTQ and AWQ selection	H.2	`realptq_select_*.json` (`_qwen7b_l412.json`: $L/4$, $L/2$ on 28 layers)
Real SparseGPT-style 2:4 selection	H.3	`sparseselect_*.json` (`_{qwen7b,tinyllama,gptj6b}_l412.json`: $L/4$, $L/2$ at every depth; the GPT-J files were re-measured, see the manifest); `analysis/verify_realsparse.py` re-derives the table, the budgets and the registered rule
Linearisation sweep, per model	G.8	`ksweep_profile_*.json`
Restricting a greedy search	G.9	`profile_seeded_search.json`
OLMoE one-shot selection, six cells	G.4	`substitutions_olmoe.json`, `selection_by_profile_olmoe.json`
The spot-check gate	G.11	`spotcheck_gate*.json`
Pairwise interactions, 403 subsets	C.5	`pairs_160m.json`, `oracle_160m_k6.json`
LoRA healing	H.8	`healing_report.json`
Distillation dissociation	H.5	`distill_obj_410m.json`
Curvature as a training objective	H.5	`curvature_distill_*.json`
§6: downstream tasks		
Downstream task rows	H.6	`downstream_*.json`
Table G.1, all three families	H.6, F.4	`envelope_table.json`, `envelope_table_third.json`
Position-resolved damage	H.11	`longctx_410m.json`
Serving footprint, $896 \to 678$ MiB measured	H.10	`serving_*.json`, `analysis/exp70_reconcile.py`
The free prior priced, gate $\hat{c} > 3$	I.1	`73_prior_gate_scoreboard.py`; `prior_gate_scoreboard.json`
Alignment reproducibility, ± 0.3–0.4 nats	D.4	`74_longctx_aligned.py`; `longctx_aligned_qwen7b_k7*.json`
Deletion-by-skip serving, equivalence checked	H.10	`72_deploy_prune_serving.py`; `deploy_prune_*.json`
Deletion damage flat in position	H.9	`74_longctx_aligned.py --substitution prune`; `longctx_prune_llama38b_k8.json`
Instruct transfer, both selectors	H.6	`downstream_llama38b_instruct_*.json`; `gsm8k_llama38b_instruct_*.json` (the `_k8` files are CurvProbe, the `_bi_k8` files Block Influence)
Eight-shot GSM8K, retention and paired test	H.11	`gsm8k_*.json`; `analysis/gsm8k_paired.py`
Generational split in $\hat{c}$, $p = 0.00033$	I.1	`73_prior_gate_scoreboard.py`; `substitutions_*.json`
BOS control on a tokenizer that has one (GPT-J)	I.1	`transfer_gptj6b_{bos,nobos}.json`
GPT-J grid on three protocol variants	I.8	`09_transfer.py --no-bos`; `transfer_gptj6b_fixed_nobos.json`; `substitutions_gptj6b{,_fixed,_unified}.json`; `analysis/gptj_protocol_variants.py`
Adapter round-trip, seven paths	D.3	`76_adapter_roundtrip.py`; `roundtrip_baseline.txt`
Grid run both ways under the GPT-J adapter fix	D.3	`gptj_adapter_sensitivity.py`; `substitutions_gptj6b_fixed.json`
EXP89, frozen before measurement	A.1	`prereg_generation_split.json`
EXP90, gated MLP vs. normaliser	A.1	`prereg_olmo_gated_vs_norm.json`; `transfer_olmo7b.json`
Split under four summary statistics	I.16	`summary_robustness.json`
Centring dose-response, three checkpoints	I.5	`77_centering_doseresponse.py`; `centering_{gpt2m,p160m}.json`, `centering_qwen05_reverse.json`
Training trajectory, both normalisers	I.15	`80_training_trajectory.py`; `trajectory_tinyllama.json`
Norm placement across the corpus	I.6	`81_norm_placement.py`; `norm_placement.json`
EXP102, OLMo-2, frozen before measurement	I.7	`prereg_exp102_olmo2.json`; `transfer_olmo2{11247b,04251b}.json`, `roundtrip_olmo2*.json`, `curvature_index_with_olmo2.json`
EXP104, operation-free validity check	I.8	`09_transfer.py --probe-resid`; `transfer_*_proberesid.json`
EXP105, rate gate, frozen before measurement	I.13	`prereg_exp105_isogate.json`; `transfer_{bloom7b,olmo2_1b,olmo2_7b}_iso.json`; `analysis/iso_gate.py`
EXP106, weight-space score on three displacements	E.11	`37_weightspace_coverage.py`; `weightspace_coverage_*.json`

EXP107, matched normaliser training pair	I.9	prereg_exp107_matched_pair.json; 38_matched_norm_pair.py; matched_norm_pair.json; transfer_pair_*.json
EXP108, three unseen organisations	I.11	prereg_exp108_unseen_families.json; transfer_{redpajama3b,falcon3_7b,smollm2_1p7b}_iso.json; analysis/exp108_verdict.py
EXP109, genericity of a deployed quantiser	I.10	prereg_exp109_quantiser_genericity.json; aniso_realq_{qwen05,1.4b,7b}.json
EXP110b, normaliser dose-response in one checkpoint	I.12	prereg_exp110{,b}_audit_doseresponse.json; 83_audit_doseresponse.py; audit_dose_{opt13b,opt13b_full,p160m}.json; analysis/exp110b_verdict.py
Inversion window, predicted vs realised	I.14	82_window_prediction.py; window_prediction.json
DC fraction ρ at every norm input	I.5	79_dc_fraction.py; dc_fraction.json
DC-accumulation mechanism, refuted	I.4	78_dc_component.py; dc_{gpt2m,tinyllama}.json
Deployment operating point, Table G.10	G.8	ksweep_profile_qwen7b.json
Table H.5, end-to-end quantisation	H.4	e2e_*.json, e2e_extra_*.json
Family rule as a threshold predictor, LOO 0.867	F.3	family_predictor.json
§7: scope and registered predictions		
Anisotropy, $\rho = +0.697$ over 1090 pairs	F.1	aniso_summary_9model.json
Curvature-generic: variance identity, $r \geq d/(\lambda_{\max}/\bar{\lambda})$	F.1	derivation; metric_lowrank_410m.json
Curvature-generic ladder, 9 checkpoints	F.1	analysis/curvature_generic.py
Profile inherited by lineage, not scale	I.17	analysis/lineage_residual.py
Effective rank r_ℓ, and a refuted prediction	F.8	effrank_*.json, analysis/exp71_verdict.py
Two-family partition recovered	F.2	family_recovery.json
Cross-substitution correlations	F.1	crossmodel_summary.json
Family rule under two continuous knobs	F.3	swasweep_*.json, tempsweep_*.json
Registered checks (B)–(E)	A.2	newsubs_*.json, moresubs_*.json
fifteen-checkpoint extension: 82/90, n=55 min ρ=+0.1498	F.4	verify_lowerbound.py; substitutions_*.json, newsubs_*.json
CGT boundary on two external operations (exp79)	F.5	79_cgt_extrapolation.py; exp79_*_v2.json
Registered frozen test on six never-seen deployment cells (exp80)	F.6	80_cgt_frozen_triage.py; exp80_410m.json, exp80_gpt2m.json
Registered fourth-family envelope probe, bloom-7b1 (exp82), fixed harness	F.7	82_envelope_fourth_family.py; 82_preregistered_verdict.json, envelope_table_fourth.json; first run in superseded/bloom_residual_bug/
The fp32 read timed against measurement, and the bf16 read's failure (exp123)	D.10	fp32_reads_qwen7b.json, decision_cost_onepass_qwen7b_{bf16,fp32}_e123.json; verify_fp32_cost.py
Displacement in one pass, and what a decision costs (exp115, exp116)	D.10	115_onepass_eps.py, onepass_*.json; 116_decision_cost_onepass.py, decision_cost_onepass_qwen7b.json
The directional read against Block Influence, seven checkpoints, and its endpoints (exp117)	F.10	117_clean_probe.py; kclean_*.json, selection_kclean_*.json; downstream_llama38b_{k8_kclean_lambada,instruct_k8_kclean}.json; gsm8k_llama38b_k8_samerun.json; e2e_gemma29b_prune_sets_postnorm_B.json
The restricted greedy search, five checkpoints (exp118, exp120)	G.9	118_seeded_greedy.py, 120_additivity_screen.py; seeded_{qwen7b,gemma29b,bloom7b}.json, additivity_screen_{qwen3_8b,olmo7b}.json
Profile sets on four further checkpoints; the additivity screen (exp120)	G.9	additivity_screen_*.json; verify_additivity_screen.py
The search's candidate pool against Block Influence's and random pools (exp121)	G.9	121_pool_control.py; pool_control_{qwen7b,qwen3_8b,olmo7b}.json; verify_pool_control.py
Where the criterion does not pay, with the size of each boundary	J	the section named in each row of Table J.1
appendix only: the named allocator		
Ranking against OWL's D^ℓ, 11 of 12 checkpoints	G.12	84_owl_fix_table.py; owl_sweep_*.json
The 30-cell grid and its two registered refutations (exp86)	G.12	86_owl_prediction_grid.py; owl_grid_summary.json, owl_pred_locked.json
Water-filled budget, 4 of 5 against the linear map (exp87)	G.12	87_waterfill.py; owl_wf_*.json, waterfill_summary.json

A Registered predictions and checks

A.1 What was registered before measurement, and what happened

Each row is a prediction written into a file, hashed and frozen *before* the measurement that judges it. The criterion column is the frozen wording, and the registration files ship with the supplementary material.

Eighty predictions across thirty-one registrations. Forty held. Forty did not, and four of the forty are the paper's central reading. The first two are EXP107 (I.9): one architecture trained twice under the two normalisers, all three outcomes registered at equal specificity, the causal reading refuted at that scale, and §I.1, the abstract and Table I.2 changed to say so. The third, EXP108's Falcon3-7B (I.11), the counterexample that took the out-of-sample record to eight of eleven. The fourth, EXP110b (I.12), turned the audit's own quantity inside one trained checkpoint without recovering the

between-checkpoint association, so §I.1 is stated as the between-checkpoint reading it is scored on, not as within-model causation. The other thirty-six map boundaries rather than testing the reading. The largest groups: five where the curvature-genericity threshold stops sorting operations, three the checkpoint family whose quadratic probe collapses (Bloom-7B1, on a harness that dropped its residual stream; Appendix F.7), eight the budget (score corollaries, two mechanisms for a baseline's failure, one evaluation-protocol window, two the centring asymmetry, one safe-side prediction, two an extrapolation between departure and direction landing in its pre-declared inconclusive band). Two the safe side of the audit, registered twice and refuted twice. The factorisation itself (121 of 128 model–operation cells over eighteen checkpoints, never negative on the weight-noise family, n=63) was never placed at risk. The seven registrations added last (EXP115, EXP117–EXP121, EXP123) put the one-pass displacement, the directional read, the restricted search, the read's second endpoint, an additivity screen, the search's candidate pool and the timed fp32 read at risk: fourteen held; the eight that did not are a reproduction tolerance on TinyLlama, the Gemma-2-9B endpoint, the restricted search's distance to the full one on Gemma-2-9B, the two OLMoE-1B-7B endpoint criteria against Block Influence (Appendix F.10), the screen's routing, which flagged three sets already within its margin, and two pool criteria: the read's pool beats random pools of the same size on one checkpoint of three (Appendix G.9). The two claims that *were* open are the prior audit of §I.1. EXP88 froze its rule, constants, criterion and snapshot hash before LLaMA-13B was measured, and held. EXP92 set the audit against the architecture rule on the one checkpoint where they disagree. EXP90 the normaliser reading against the gated-MLP reading, and whether the normaliser *sets* $\hat{c}$ rather than predicting it, undecidable in the corpus, decided against us by EXP107 and tested by EXP108 on the label rule itself, under a scope condition fixed in advance, on three organisations the paper had never measured. EXP80's registration committed the paper, if its main criterion failed, to report the failure and change neither the threshold, the operations, nor the statistic, and we did none of the three.

registration	prediction, as frozen	outcome
EXP77	H1∧H2: κ-correction lower bound conserved on a seventh lineage, Mistral-7B	**confirmed**
EXP110b	H0: $\lvert\mathrm{Spearman}(\hat{c}, \overline{\Delta\rho})\rvert < 0.4$ over five normaliser settings in one checkpoint; §I.1 narrows to the between-checkpoint association it is scored on	**confirmed** — $+0.300$; narrowing made (I.12)
	H1: within a model it holds at $\geq +0.7$, endpoints past the registered margin; H2: reverses at ≤ -0.4	**both refuted**; 84% of the movement falls past the registered clean-loss guard, uncounted as support
EXP109	H1: a deployed quantiser's perturbation is more curvature-generic than round-to-nearest's — our explanation for selection stopping under good quantisers	**confirmed**, too small to mean it — medians -0.016 and -0.026 against the 0.603 separating the regimes (I.10)
	H0/H2: no consistent ordering, or the deployed quantisers are *less* generic	**both refuted**; replaced by a measured explanation — int3 rounding is already generic
EXP108	H1/H3: under a scope condition and label rule frozen in advance, an unseen organisation reads on the side its normaliser predicts (RedPajama-INCITE-3B < 3; SmolLM2-1.7B > 3)	**both confirmed** — 1.64 and 6.74
	H2: Falcon3-7B, uncentred, reads $\hat{c} > 3$	**refuted** — 2.67, the first uncentred checkpoint below the constant (I.11)
EXP107	H1: trained under the two normalisers with everything a training run can hold fixed held fixed, the uncentred run's $\hat{c}$ exceeds the centred run's by more than 1.0 at the final checkpoint; H2: > 1.0 the other way	**both refuted** — $\Delta = -0.12$
	H0: they differ by less than 1.0, and we report the normaliser as predicting $\hat{c}$, not setting it	**confirmed**; causal reading withdrawn (I.9)

registration	prediction, as frozen	outcome
EXP95	Two-sided, frozen after the inverse sweep returned a null and before ρ was measured. H1: the DC fraction $\rho > 0.10$ in a LayerNorm-trained checkpoint and < 0.02 in an RMS-trained one, so the null is a power failure; H0: $\rho > 0.10$ in the RMS one too, so the null is real and §I.1's manipulation claim narrows to LayerNorm checkpoints	**both refuted** — $\rho = 1.5$–7.6×10^{-4} on all three, smallest where the response is largest; neither reading claimed
EXP97	$H_{\text{departure}}$: sweeping past the identity point either way moves $\hat{c}$ regardless of sign, i.e. GPT-2-medium moves by more than 0.3 between α=1.0 and α=1.5; frozen with a pre-declared band for neither prediction $H_{\text{direction}}$: $\hat{c}$ tracks only the sign of the DC change, i.e. the same move is < 0.1	**refuted** — moves by 0.182 **refuted** — 0.182 exceeds it; inconclusive per the registration's third band (I.5)
EXP90	H1: OLMo-1-7B (gated MLP, non-parametric LayerNorm) reads $\hat{c} > 3$, i.e. the gated MLP sets the rate and the normaliser is incidental; frozen with checkpoint hash, opposite reading at equal specificity H1-alt: $\hat{c} < 3$, i.e. the normaliser sets the rate	**refuted** — $\hat{c} = 0.69$, CI $[0.37, 1.00]$ **confirmed**; with OLMoE (5.66) a matched pair differing in the normaliser among the five candidate causes, so the claim moved from the block to the normaliser
EXP92	Two rules frozen before OPT-6.7B's damage was measured, on the one checkpoint where they disagree: the architecture rule (LayerNorm $\Rightarrow$ free depth heuristic safe, $\Delta\rho \leq 0.05$) against the prior audit ($\hat{c} = 3.75 > 3 \Rightarrow$ unsafe, $\Delta\rho > 0.05$)	**the audit was right** — $\Delta\rho = +0.091$, the corrected score ahead on all four completed operations
EXP89	H1: GPT-NeoX-20B (2022 recipe, 20B, 44 layers) reads $\hat{c} < 3$; frozen with rule, boundary, operation set and snapshot hash H2: Yi-1.5-9B (2024 recipe, lineage absent from the derivation set) reads $\hat{c} > 3$	**confirmed** — $\hat{c} = 2.569$ **confirmed** — $\hat{c} = 7.030$, reproduced to four decimals on two independent GPUs
EXP88	The two-sided prior audit ($\hat{c} > 3$ or $\hat{c} < 0$, both constants the free prior's own, nothing fitted) classifies LLaMA-13B correctly against the measured $\Delta\rho > 0.05$ criterion; frozen with rule, operation set, margin and snapshot hash	**confirmed** — $\hat{c} = 5.39$, predicted unsafe; measured $\Delta\rho = +0.711$, the free prior at $\rho = -0.135$ against $+0.577$
EXP99	H1: OLMo-1-7B ($\hat{c} = 0.687$, far inside $[0, 3]$) is SAFE, testing the audit's thinnest side (four true negatives out of sixteen); frozen with checkpoint hash, operation set, 0.05 margin H0, at equal specificity: OLMo-1-7B is UNSAFE despite $\hat{c} = 0.687$	**refuted** — mean $\Delta\rho = +0.150$, three times the margin, from a checkpoint that is not a boundary case **confirmed**. With EXP96 the safe side is $4/7$, and §I.1 claims the rule in one direction only
EXP98	H1: an RMS-normalised checkpoint's $\hat{c}$ RISES over training by ≥ 1.0 from earliest to latest public checkpoint, where Pythia-160M's falls; frozen with checkpoint list and threshold H0, at equal specificity: the RMSNorm rate is set early and training does not move it ($\lvert\Delta\rvert < 0.5$)	**confirmed** — TinyLlama reads 3.84 at 105B tokens and 5.58 at 3T, a rise of $+1.74$; nonmonotone, peaking at 6.91 (I.15) **refuted**
EXP96	H1: OLMo-1-1B ($\hat{c} = 0.839$, inside $[0, 3]$) is SAFE: the audit's two-sided rule predicts the corrected score does not beat the heuristic; frozen with checkpoint hash, operation set, 0.05 margin H0, at equal specificity: OLMo-1-1B is UNSAFE despite $\hat{c} = 0.839$	**refuted** — mean $\Delta\rho = +0.055$, clearing the margin by 0.005; the audit's second miss, with GPT-2-medium (§I.1) **confirmed**; the family split holds inside the miss — capability-removal favours the heuristic, weight-noise the corrected score

registration	prediction, as frozen	outcome
EXP79	H1: `ropetrunc` scores above τ, i.e. capability-removal	**refuted** — it behaves like weight noise on both checkpoints
	H2: `headscale` scores at or below τ, i.e. weight-noise	**refuted** — it behaves like capability-removal
	H3 (corollary): salience ρ orders the two accordingly	**refuted**, and in the same direction
EXP80	R1: registered family assignments for three never-seen deployment operations right on at least $2/3$ of resolved cells	**refuted**, 2 of 5
	R2 (main): on resolved cells the *frozen* τ agrees with measured behaviour on at least $2/3$	**refuted**, 2 of 5 — the failures are all within two jackknife s.d. of τ
EXP82	R1: the transfer diagnostic passes on a fourth family (Bloom-7B1)	**refuted** (corrected -0.033 vs raw $+0.233$ on `prune`)
	R2: no significant loss to deepest-k at $L/4$	**refuted**, $4/4$ tasks
	R3: the ordering reverses at $L/2$	confirmed (HellaSwag -0.09)
	R4 (guard): the corrected selection beats the depth prior	**refuted**, $4/4$ tasks
EXP84	R1 (protocol sanity): dense LLaMA-13B perplexity inside 5.09 ± 0.5	**not met as registered** — see the note below
	R2: our budget beats OWL's at the same total sparsity	confirmed (50.58 vs 87.92)
	R3: at least 3 of 4 downstream tasks not below OWL	confirmed, 4 of 4
EXP85	H1: on Qwen2.5-7B the single loss is saturation, so the budget is not worse at the two sparsities where *that* checkpoint still works. (Scope is the checkpoint; EXP86 registers the cross-checkpoint test separately, and it fails.)	**confirmed**
	H2: the loss is alignment-driven and sparsity-independent	not supported (mutually exclusive with H1)
	H3: $\mathrm{sign}(\rho(\text{score}, a) - \rho(D, a))$, computable before pruning, calls the winner	**refuted**, 4 of 6 cells
EXP86	primary: the operating band and the probe-validity gate, both read before compressing, predict which cells gain over uniform, on six held-out checkpoints	**refuted**, 4 of 13 cells as first measured (3 on the fixed harness) — below chance ($p = 0.95$), and the band fails at both edges
	falsifier: Bloom-7B1, whose probe our own diagnostic calls invalid ($\lvert R_3\rvert/a = 0.92$), gains nothing in band	**refuted** — it gains 19.4% at 70% sparsity
EXP87	primary: a budget water-filled from our own damage factorisation beats the linear inversion on ≥ 4 of the 5 cells inversion loses	confirmed, 4 of 5
	secondary: those five losses flip against OWL, ≥ 3 of 5	**refuted**, 1 of 5 — the one flip is decisive (Mistral-7B at 70%: 57.3 against OWL's 68.1)
EXP93[§]	GPT-2-medium (0.35B), matched in scale to inverted Pythia-410M but from another lineage, does not invert	**refuted** — it inverts, $\rho = -0.480$, so the lineage reading fails
	TinyLlama (1.1B), matched to inverted Pythia-1.4B, does not invert	confirmed, $\rho = +0.718$
EXP94[§]	primary: if D^ℓ's allocation is carried by depth, replacing it with its own depth trend costs $\leq 10\%$ on $3/4$ cells	**refuted**, 0 of 4 — it costs 23 to 44%
	contrast: that reduction costs our allocation more than OWL's	**refuted**, 1 of 4
P1A-strict	P1 (Llama-3-8B): the frozen score $\frac{1}{2}\lVert\epsilon\rVert^2\kappa^{\mathrm{iso}}$, profile and perturbation magnitude both estimated on calibration articles, ranks held-out damage above the calibration magnitude rule in $3/3$ segments	**confirmed** — $+0.63$–$+0.86$ against -0.37 (Table E.9)
	P2 (Gemma-2-9B): the same direction in $\geq 2/3$ segments	**confirmed** — $3/3$, $+0.49$–$+0.55$ against -0.03
	P3: a directly measured per-layer damage score ranks $\geq$ the frozen score on both models	**confirmed** — direct $\rho \geq +0.96$

registration	prediction, as frozen	outcome
	P4: the frozen score is positive in $3/3$ segments, on both models	**confirmed**
	P5: permuting the profile across layers drops the correlation in $3/3$ segments	**confirmed** — to -0.24 (Llama-3-8B) and $+0.03$ (Gemma-2-9B)
	P6: freezing the perturbation magnitude onto calibration costs ≤ 0.15 in ρ against the component-transfer score	**confirmed** — at most 0.05
EXP115	exact: one clean pass gives $\|\epsilon_\ell\|$ at every layer equal to the per-layer route, $\leq 10^{-4}$ relative, every operation, five checkpoints	**confirmed** — 0 on all 55 (D.10)
	repro: it reproduces the released $\|\epsilon_\ell\|$ to $\leq 10^{-3}$, Spearman ≥ 0.999	**refuted** on TinyLlama's five core operations — up to 4.2×10^{-3}, Spearman 1.000
EXP117	P1: the directional read out-ranks Block Influence on deletion on ≥ 5 of the six checkpoints where Block Influence out-ranks the profile	**confirmed** — $6/6$ (F.10)
	P2: its $k{=}L/4$ tail recall is $\geq$ Block Influence's on ≥ 5 of the six	**confirmed** — $6/6$
	P3: it does not rank linearisation below the profile on more than one	**confirmed**
	E-Llama: its Llama-3-8B set is at or below Block Influence's NLL and at or above its LAMBADA	**confirmed** — 2.66 against 2.79, 0.601 against 0.479
	E-Gemma: its Gemma-2-9B set is within 0.10 nats of direct measurement's	**refuted** — 4.00 against 3.02
EXP118	G1: a greedy restricted to the clean-output read's $2k$ cheapest layers is at or below Block Influence on every checkpoint run	**confirmed** — $3/3$ (G.9)
	G2: within 0.05 nats of the full greedy	**refuted** on Gemma-2-9B ($+0.156$); held on Qwen2.5-7B and Bloom-7B1
	G3: at most half the full greedy's evaluations	**confirmed** — 41–47%
EXP119	D1: the read's OLMoE-1B-7B set is at or below Block Influence's WikiText-2 NLL	**refuted** — 3.151 against 3.070 (F.10)
	D2: and at or above its LAMBADA	**refuted** — 0.364 against 0.458 ($156{:}342$, $p = 4.9 \times 10^{-17}$)
	D3: and at or below the isotropic set's NLL	**confirmed** — 3.151 against 3.502
EXP120	A1: with $\tau = 3$, $R \leq \tau$ keeps the profile's set within 0.05 nats of the better of Block Influence and deepest-k and $R > \tau$ flags a set outside it, on ≥ 7 of 8 settings	**refuted** — $5/8$; the three misses are flags on sets inside the margin (G.9)
	A2: where $R > \tau$ at $k{=}L/4$, the profile-restricted greedy is below Block Influence and the profile's set	**confirmed** — $2/2$, 0.71 nats below Block Influence on both
	A3: $R > \tau$ at $k{=}L/2$ on ≥ 3 of 4 checkpoints	**confirmed** — $3/4$
EXP121	C1: the read's $2k$ pool leads the greedy to a set within 0.05 nats of Block Influence's pool (Gemma-2-9B registered, not run)	**confirmed** — $3/3$ (G.9)
	C2: and more than 0.05 nats below the mean of three random pools	**refuted** — $1/3$, OLMo-1-7B only
	C3: and more than 0.05 nats below Block Influence's pool, on ≥ 2	**refuted** — $0/3$
EXP123	V1: the timed fp32 profile is the released one, $\rho \geq 0.98$	**confirmed** — 0.993 (D.10)
	V2: the timed directional read is the released one, $\rho \geq 0.98$	**confirmed** — 1.000
	V3: the clean-output step gives the released profile, $\rho \geq 0.98$	**confirmed** — 0.991
	C1/C2 (mapping, not a prediction): break-even of the profile and directional routes	7.3 and 7.2 operations; §7 now says "ahead from the eighth"

§EXP93 and EXP94 ship as `exp88_preregistered.md` and `89_depth_proxy.py`, renumbered here because EXP88 and EXP89 already name the LLaMA-13B audit and the generation-split registrations.

The one entry that needs a note. EXP84's R1 froze a window around LLaMA-13B's published dense perplexity. We measure 6.66, outside it. The cause is the protocol, not the pipeline: this comparison reads the token-loss evaluation over 200 sequences of 256 tokens, not the 2048-token full-test-set convention behind the published figure, and the shorter context raises the absolute number. (The substitution grid of §3 uses 32 sequences of 1024 tokens. The protocols are never mixed inside a comparison.) Every arm and the clean baseline it is divided by is read under that same protocol, so R2, R3 and the ratios of §G.12 are unaffected. Only absolute perplexity against a published table is not comparable. We record R1 as not met, not reinterpret it.

One run was voided rather than reported. EXP79's first execution returned curvatures negative at almost every layer. The registration was untouched. We diagnosed the error (a whole-set baseline differenced against a single-batch expansion, leaving a constant offset a sum does not cancel), fixed it, re-ran the measurement, and added an operation-agnostic consistency check so the same failure cannot pass silently again. The voided run and its diagnosis stay in the registration file. The re-run then refuted the registration.

Two Bloom-7B1 entries were judged on a broken harness. EXP82's R1 and EXP86's falsifier were scored on replacement modules that dropped Bloom's residual stream (Appendix F.7). Re-measured with the residual restored, EXP82's R1 holds ($+0.013 \to +0.350$) and R2–R4 come out as before. EXP86's falsifier loses its premise: Bloom's probe now passes the diagnostic, and the budget no longer gains at 70% sparsity (281 against uniform's 277), moving the primary criterion from 4 of 13 cells to 3 and leaving it below chance. The count keeps the verdicts as first registered.

Further registered predictions live in the sections that use them, uncounted above: the two continuous-knob sweeps of §F.3 ($4/4$ confirmed on the window sweep. On the temperature sweep the registered direction holds and the single crossing does not), the two commitments the variance identity makes in §F.8 (one confirmed, one refuted), the six out-of-sample operations of Table E.7, and checks (B)–(E) of §A.2.

A.2 Registered checks (B)–(E) in full

(E) In-block compensation: architecture-level, not layer-level. A sequential residual lets a layer's MLP partially cancel an attention perturbation. A parallel one cannot (§C.3). With $c_\ell = \|\delta_{\ell+1}\|/\|\epsilon_\ell\|$ the fraction of injected ϵ_ℓ surviving to $h_{\ell+1}$, a parallel residual must give exactly 1: measured to 10^{-8} on all three Pythias (one layer per model excluded, HuggingFace applying the final layer norm), mean c 1.000000 on every Pythia, 1.12–2.23 elsewhere. Robust signal: signed reaction $\cos(\text{reaction}, \epsilon)$, -3×10^{-4} on the Pythias, -0.21 to -0.38 on the sequential models, tracking $\rho(\texttt{linear}, \texttt{quant4})$ across all six at $\rho = -0.886$ (exact permutation $p = 0.033$).

Magnitude fragile, the layer-level test unsupportive: $\Delta_c := \bar{c}(\texttt{linear}) - \bar{c}(\texttt{quant4})$ against $\rho(\texttt{linear}, \texttt{quant4})$ gives $+0.886$ ($p = 0.033$) without the final layer, $+0.714$ ($p = 0.136$) all layers, $+0.771$ ($p = 0.103$) without first and last, all three reported, the choice changing the conclusion. $|c_\ell(\texttt{linear}) - c_\ell(\texttt{quant4})|$ against how far apart the families rank ℓ gives $\rho = -0.35$, CI $[-0.47, -0.09]$, *against* the registered direction, uninformative within the sequential group ($[-0.30, +0.56]$). Direction tracks the family sign. Magnitude and layer structure do not mediate the reversal, a partially identified signal, not a mechanism.

(B) The quantiser is not an artefact of round-to-nearest (within a model, not across the grid. GPTQ reconstructs against the layer-input Hessian, AWQ searches per-input-channel scales, both pointing ϵ differently from RTN) and direction is what §4 claims matters. All six quantisers (GPTQ, AWQ, RTN $\times$ int4/int3) behave like RTN on Pythia-410M (cross-correlation with `quant4` $+0.90$ to $+1.00$, against -0.46 to -0.60 with `linear`), but weaken on Qwen2.5-0.5B to $+0.275$–$+0.527$, GPTQ's correlation to `linear` turning slightly negative. Direction depends on algorithm and model, not the operation's name, reported, not asserted model-independent.

(C) The taxonomy predicts out of sample, 2/3. Registered before measurement, three substitutions used nowhere else: sliding-window attention at $w = 128$, head-pruning (both predicted capability-removal), activation quantisation (predicted weight-noise). On Pythia-160M and 410M the last two land in their families (separation 0.50–1.37). Sliding window lands with weight-noise on both ($\rho(\text{position}) = +0.22/+0.06$, $\rho(\|\epsilon\|^2, a) = 0.56/0.81$), a miss §F.1 takes up.

(D) The reversal is not measurement noise. Split-half re-estimation of a_ℓ gives Spearman–Brown reliability 0.94–0.99 on GPT-2, matching every other model (0.93–0.99). Disattenuating moves GPT-2's `linear`–`quant4` from $+0.437$ to $+0.454$ (Appendix D.8).

Related work we build on. Closest recent work shares our ingredients, Lucas & Mazumder (2025), Liu et al. (2024) (second-order curvature), Chen et al. (2025), Lin et al. (2025), Gromov et al. (2025) (layer importance), Ge et al. (2024), Wijmans et al. (2025), Kumar et al. (2025) (serving cost), but none reads the curvature as a forward-only per-checkpoint profile pricing unseen operations.

B Extended related work

Second-order salience has stayed inside one method at a time: OBD/OBS (LeCun et al., 1989; Hassibi & Stork, 1992) for pruning, HAWQ (Dong et al., 2019) ranking layers by the top Hessian eigenvalue and HAWQ-V2 (Dong et al., 2020) arguing for the *average* trace instead, HAP (Yu et al., 2022) carrying it to structured pruning, still, six years on, derived once per method with a backward pass per implementation. Fisher and K-FAC (Martens & Grosse, 2015) approximate H structurally where we need a directional Rayleigh quotient at one site. Neuromanifold trace estimates (Sun, 2026) share our primitive but expand in weight space. Mel (2026) shows that the test loss a small training perturbation costs depends on how the eigenvectors of the two losses overlap, not on their spectra alone; the directional term of Eq. 2 is a dependence of the same kind, and one isotropic scalar gives it up. Practice is local by construction ((Sun et al., 2024; Lin et al., 2024; Frantar et al., 2022; 2023; Frantar & Alistarh, 2023; Men et al., 2025; Ashkboos et al., 2024; Kurtic et al., 2023)), each $\|\epsilon\|$ in a different coordinate system, now an explicit one for quantisation (Birnick, 2026); LLM-Pruner (Ma et al., 2023) is the exception, a first-order importance from the loss gradient. We reproduce Lu et al.'s verdict (§5): they fail: the quantity they share goes negative on the operations that remove capability. Deletion-specific selectors measure the loss instead, once per block (Shortened-LLaMA (Kim et al., 2024)) or again after every removal (SLEB (Song et al., 2024)). §G.1 prices both.

Neighbouring lines, not baselines here: each answers a different question from which layer to apply a fixed operation to. Low-rank decomposition that budgets rank per layer by measuring perplexity at each candidate truncation (Yuan et al., 2023) is an allocator. The profile gives an ordering, not a budget (the fixed-rank variant is what §G.1 prices. A better ordering does not make a better allocator, Appendix G.12). Depth reduction folding adjacent blocks into one (Yang et al., 2024b) replaces the deletion cliff with a different parameterisation of the collapsed block, not the quantity we rank. Sparsity allocation from a reconstruction-error bound (Huang et al., 2025) gives a continuous budget, the boundary §G.1 already reports. BlockPruner (Zhong et al., 2025) and FinerCut (Zhang et al., 2024b), pruning sub-block units and below-whole-layer granularity, answer *what inside a layer* to remove, whereas every selector priced here answers *which layer* to apply a fixed operation to (the deployment arm of §6 holds the operation fixed for exactly that reason), a different comparison, not a missing one. Closest, a Hessian-free second-order pruning framework solving for retained weights at a fixed pattern (Lucas & Mazumder, 2025) takes Newton steps without forming or storing the Hessian, through Hessian-vector products; its object is a weight-space reconstruction objective and its decision which weights to keep, ours the loss curvature at the sublayer output, read with forward passes alone, and which layers to convert.

Forward-only estimation, and what is being converted. MeZO (Malladi et al., 2023) replaced backpropagated gradients with finite differences along random directions, for *optimisation*: the same primitive we build on, read for a different coefficient, its estimate first-order, driving an update. Ours the symmetric difference, cancelling the first-order term, reading a quadratic coefficient nothing is updated with (the checkpoint measured, not moved). The operations come from the efficient-attention literature (Katharopoulos et al., 2020; Choromanski et al., 2021; Yang et al., 2024a; Gu & Dao, 2024; Lieber et al., 2024; Ren et al., 2025; Lan et al., 2026; Zhou et al., 2026).

C Proofs

Throughout $\mathcal{L} : \mathbb{R}^D \to \mathbb{R}$ is the evaluation loss as a function of layer ℓ's attention output y_ℓ, every parameter and every other layer's *function* fixed, $D = B \cdot L \cdot d_{\text{model}}$. $g_\ell = \nabla\mathcal{L}(y_\ell)$, $H_\ell = \nabla^2\mathcal{L}(y_\ell)$, $a_\ell = \mathcal{L}(y_\ell + \epsilon_\ell) - \mathcal{L}(y_\ell)$. Assume $\mathcal{L} \in C^{4,1}$ (four times differentiable, locally Lipschitz fourth derivative) on a neighbourhood of $[y_\ell,\, y_\ell + \epsilon_\ell]$, true for cross-entropy on a smooth-activation transformer, where at our sequence lengths and perturbation sizes the softmax denominators stay bounded away from zero. ReLU checkpoints are piecewise smooth only, so the expansion holds at generic points and we rely on the remainder check of Appendix C.2.

C.1 When a magnitude score is faithful

Proposition 2 (Exact expansion). *With the assumptions above,*

$$a_\ell \;=\; \underbrace{\|g_\ell\|\,\|\epsilon_\ell\|\cos\theta_\ell}_{\text{first order}} \;+\; \underbrace{\tfrac{1}{2}\,\epsilon_\ell^\top H_\ell\,\epsilon_\ell}_{\text{second order}} \;+\; R_3(\epsilon_\ell), \qquad \cos\theta_\ell = \frac{\langle g_\ell, \epsilon_\ell\rangle}{\|g_\ell\|\|\epsilon_\ell\|}, \tag{3}$$

with the exact integral remainder $R_3(\epsilon) = \frac{1}{2}\int_0^1 (1-t)^2\, D^3\mathcal{L}(y_\ell + t\epsilon)[\epsilon,\epsilon,\epsilon]\,dt$, *so that* $|R_3(\epsilon)| \le \frac{1}{6}\,M_3\|\epsilon\|^3$ *where* $M_3 = \sup_{t\in[0,1]}\|D^3\mathcal{L}(y_\ell + t\epsilon)\|$.

Proof. Taylor's theorem with integral remainder for $\phi(t) = \mathcal{L}(y_\ell + t\epsilon)$ on $[0,1]$, using $\phi'(0) = \langle g_\ell, \epsilon\rangle$, $\phi''(0) = \epsilon^\top H_\ell \epsilon$, $\phi'''(t) = D^3\mathcal{L}(y_\ell + t\epsilon)[\epsilon,\epsilon,\epsilon]$. The bound follows from $\int_0^1 (1-t)^2 dt = \frac{1}{3}$. □

Writing $\kappa_\ell = \hat\epsilon_\ell^\top H_\ell \hat\epsilon_\ell$ with $\hat\epsilon_\ell = \epsilon_\ell/\|\epsilon_\ell\|$, equation 3 factors as

$$a_\ell \;=\; \|\epsilon_\ell\|^2\, c_\ell, \qquad c_\ell \;:=\; \tfrac{1}{2}\kappa_\ell \;+\; \frac{\|g_\ell\|\cos\theta_\ell}{\|\epsilon_\ell\|} \;+\; \frac{R_3(\epsilon_\ell)}{\|\epsilon_\ell\|^2}. \tag{4}$$

We call c_ℓ the *effective coefficient*: what a magnitude score cannot see.

Proposition 3 (Fidelity condition). *Let* $s_\ell = \psi(\|\epsilon_\ell\|)$ *with* ψ *strictly increasing, and suppose* $a_\ell > 0$ *for all* ℓ. *The ranking induced by* s *coincides with the ranking induced by* a *if and only if*

$$\frac{c_\ell}{c_m} \;<\; \Big(\frac{\|\epsilon_m\|}{\|\epsilon_\ell\|}\Big)^2 \qquad \textit{for every pair with } \|\epsilon_\ell\| < \|\epsilon_m\|. \tag{5}$$

In particular the condition holds whenever c_ℓ *is constant in* ℓ, *and it fails whenever* c *varies across layers by more than the squared ratio of perturbation magnitudes.*

Proof. ψ strictly increasing gives $s_\ell < s_m \iff \|\epsilon_\ell\| < \|\epsilon_m\|$. By equation 4, $a_\ell < a_m \iff c_\ell\|\epsilon_\ell\|^2 < c_m\|\epsilon_m\|^2$, i.e. equation 5 for positive c, and constant c suffices. Conversely, for any $\|\epsilon_\ell\| < \|\epsilon_m\|$ one may choose c_ℓ/c_m above the stated ratio, reversing the two orderings. □

So a magnitude score's adequacy is a measurement of the spread of c_ℓ against that of $\|\epsilon_\ell\|^2$: on Pythia-410M $\|\epsilon_\ell\|^2$ spans $58\times$ across layers while a_ℓ spans $37\times$ in the *opposite* rank order, violating equation 5 by a wide margin, by c_ℓ rather than by the magnitudes.

Which term of c_ℓ dominates. The ratio of the first-order to the second-order contribution in equation 4 is

$$\frac{|\langle g_\ell, \epsilon_\ell\rangle|}{\big|\tfrac{1}{2}\epsilon_\ell^\top H_\ell \epsilon_\ell\big|} \;=\; \frac{2\,\|g_\ell\|\,|\cos\theta_\ell|}{\|\epsilon_\ell\|\,|\kappa_\ell|}. \tag{6}$$

For independent isotropic directions in $\mathbb{R}^D$, $\mathbb{E}|\cos| = \sqrt{2/(\pi D)}\,(1 + O(D^{-1}))$, so $D^{-1/2}$ is the reference scale. The measured median $|\cos\theta_\ell|$ is $0.86\times$ that at Pythia-410M ($D = 2.10\times10^6$) and $1.69\times$ at Pythia-160M ($D = 1.57\times10^6$). Table C.1 evaluates equation 6 by two routes sharing no first-order measurement: the closed form takes $\|g_\ell\|$, $\cos\theta_\ell$ from backpropagation, the differencing estimate $\hat A$, $\hat B$ from forward passes only (§C.2). They rank layers consistently ($\rho = +0.760$ over 107 layers) and 90% (closed form), 93% (differencing) of layers fall below 0.5.

Table C.1: First-order over second-order contribution, equation 6, from separately measured components and, in the last two columns, from the symmetric-differencing fit. Layers with $|\kappa|$ or $|B|$ below 10^{-9} are dropped rather than winsorised (one layer of GPT-2-medium). The linear term is a minority contribution on every model and the two routes rank layers consistently.

	closed form, eq. 6			differencing	
model	layers	median	IQR	median	ρ between routes
Pythia-160M	12	0.167	[0.071, 0.266]	0.256	+0.776
Pythia-410M	24	0.074	[0.046, 0.147]	0.183	+0.788
Pythia-1.4B	24	0.087	[0.038, 0.144]	0.125	+0.550
GPT-2-medium	23	0.371	[0.148, 0.559]	0.458	+0.891
Qwen2.5-0.5B	24	0.067	[0.036, 0.120]	0.112	+0.600
pooled	107	0.100	—	0.181	+0.760

Damage is second-order here not because $\langle g_\ell, \epsilon_\ell\rangle$ vanishes identically but because ϵ_ℓ is no more aligned with g_ℓ than a random direction: its term is a minority of a_ℓ and does not vary with ℓ in a

way that would rescue a magnitude score. GPT-2-medium, where the linear term matters most (0.37 against 0.07–0.17 elsewhere), is also where the correction of §5 gains least.

C.2 The symmetric-differencing estimator

Proposition 4 (Second-order accuracy). *Let $\phi(t) = \mathcal{L}(y_\ell + t\epsilon) - \mathcal{L}(y_\ell)$ with $\mathcal{L} \in C^{4,1}$, and write $A = \langle g_\ell, \epsilon\rangle$, $B = \frac{1}{2}\epsilon^\top H_\ell \epsilon$, $C = \frac{1}{6}D^3\mathcal{L}(y_\ell)[\epsilon,\epsilon,\epsilon]$. Then for $t \neq 0$*

$$\frac{\phi(t) + \phi(-t)}{2t^2} = B \;+\; \tfrac{1}{48}\big(D^4\mathcal{L}(\xi_+) + D^4\mathcal{L}(\xi_-)\big)[\epsilon^{\otimes 4}]\, t^2, \tag{7}$$

$$\frac{\phi(t) - \phi(-t)}{2t} = A \;+\; C\,t^2 \;+\; O(t^4), \tag{8}$$

for some $\xi_\pm$ on the segment. Both estimators are second-order accurate in t, whereas the one-sided quotients are not: $\phi(t)/t$ is first-order accurate, and $\phi(t)/t^2$ does not converge to B when $A \neq 0$.

Proof. Expand $\phi(\pm t) = \pm At + Bt^2 \pm Ct^3 + \frac{1}{24}D^4\mathcal{L}(\xi_\pm)[\epsilon^{\otimes 4}]t^4$ by Taylor's theorem with Lagrange remainder. Adding cancels the odd powers, leaving $2Bt^2$ plus a fourth-order term. Subtracting cancels the even powers, leaving $2At + 2Ct^3$ plus $\frac{1}{24}\big(D^4\mathcal{L}(\xi_+) - D^4\mathcal{L}(\xi_-)\big)[\epsilon^{\otimes 4}]t^4$, which is $O(t^5)$ since the two points are $O(t)$ apart and $D^4\mathcal{L}$ is locally Lipschitz. Dividing gives equation 7–equation 8. One-sided, $\phi(t)/t = A + Bt + O(t^2)$ has leading error $O(t)$. □

(i) The $t = 1$ identity. ϵ_ℓ is by construction the change the substitution induces in y_ℓ, so $\phi(1) = a_\ell$ with no approximation: comparing the two tests pins the injection point exactly, not heuristically, and returns $\rho = +1.000$ for all five substitutions.
(ii) Why the A estimate is the noisier one. The leading error of equation 8 is Ct^2, controlled by the third-order tensor driving R_3, so $\hat{A}$ agrees with the backpropagated $\langle g_\ell, \epsilon_\ell\rangle$ where the measured third-order remainder is small and disagrees where it is large ($\rho = 0.90, 0.96$ on two models against $0.39, 0.46$ on the two with the larger remainders). B's estimator has no third-order error term, which is why we build on B.
(iii) We report the remainder, not assumed away. $a_\ell - (\hat{A} + \hat{B})$ is the residual after the fitted terms, R_3 at $t = 1$ together with the fit's own error, whose leading part is (ii)'s Ct^2: over the nine checkpoints with full error terms the pooled median $|R_3|/a_\ell$ is 1.4% (int4), 3.3% (int3), 10.9% (low-rank), 21.7% (linearisation), 25.8% (deletion), the per-model deletion median ranging 15–38%. The second-order term dominates in $\geq 82\%$ of layers for every operation, the first-order almost never (≤ 11 layers across the corpus): the ground for dropping it in §4. Order preservation matters more than size: on deletion $\rho(R_3, a_\ell)$ is $+0.82$ to $+0.95$ on six of the nine checkpoints and $+0.73$ and $+0.76$ on two more, negative only on GPT-2-medium (-0.17). It is at zero on Llama-3-8B linearisation ($+0.05$), and those two are the cells where the corrected score trails.

C.3 The parallel-residual identity

Proposition 5. *Let a block with* parallel *residual compute $h_{\ell+1} = h_\ell + A(\mathrm{LN}_1(h_\ell)) + M(\mathrm{LN}_2(h_\ell))$, and let $\tilde{h}_{\ell+1}$ be the result of replacing $A(\mathrm{LN}_1(h_\ell))$ by $A(\mathrm{LN}_1(h_\ell)) + \epsilon$. Then $\delta_{\ell+1} := \tilde{h}_{\ell+1} - h_{\ell+1} = \epsilon$ identically, for every ϵ and every M, so $c_\ell := \|\delta_{\ell+1}\|/\|\epsilon\| = 1$ exactly.*
Let a block with sequential *residual compute $h' = h_\ell + A(\mathrm{LN}_1(h_\ell))$ and $h_{\ell+1} = h' + M(\mathrm{LN}_2(h'))$. Then under the same replacement*

$$\delta_{\ell+1} = \epsilon + \Delta_M(\epsilon), \qquad \Delta_M(\epsilon) := M(\mathrm{LN}_2(h' + \epsilon)) - M(\mathrm{LN}_2(h')), \tag{9}$$

and $c_\ell = 1$ if and only if $2\langle \epsilon, \Delta_M\rangle + \|\Delta_M\|^2 = 0$.

Proof. Parallel: both LN arguments are h_ℓ, untouched by the replacement, so only the attention summand changes and the difference is ϵ. Sequential: $\tilde{h}' = h' + \epsilon$, and substituting into the second line gives equation 9, whence $\|\delta_{\ell+1}\|^2 = \|\epsilon\|^2 + 2\langle \epsilon, \Delta_M\rangle + \|\Delta_M\|^2$. □

Proposition 5 makes the correctness check of §A.2(E) a theorem, not a convention, and supplies a self-test: any deviation from $c_\ell = 1$ on a parallel-residual model means the injection misses the attention output, so we require it before reading anything else (measured deviation 3.3×10^{-9} on all three Pythia models). The sequential signed reaction $\cos(\Delta_M, \epsilon)$ is negative when the MLP pushes back inside the block and $O(10^{-4})$ when a parallel block has nothing to push against. By equation 9 the perturbation still grows when $\cos(\Delta_M, \epsilon) > -\|\Delta_M\|/(2\|\epsilon\|)$, as it does here ($c_\ell \in [1.12, 2.23]$, Appendix A.2(E)), measured -0.21 to -0.38 on the three sequential models against -3×10^{-4} on the three parallel ones.

A boundary case that must be excluded. HuggingFace applies the final layer norm to the last entry of `output_hidden_states`, so for $\ell = L-1$ the computed quantity is $\|\mathrm{LN}(h+\epsilon)-\mathrm{LN}(h)\|/\|\epsilon\|$, which Proposition 6 shows is not 1 for any architecture. That one layer per model is excluded from every statistic in §A.2(E).

C.4 Why perturbations are reported relative to the stream norm

Proposition 6. *Let* $\mathrm{LN}(h) = \sqrt{d}\,\bar{h}/\|\bar{h}\|$, *where* $\bar{h} = h - \frac{1}{d}(\mathbf{1}^\top h)\mathbf{1}$ *(the learned diagonal gain and bias are absorbed into* W_U*). Then for* $\|\delta\| \ll \|\bar{h}\|$,

$$W_U\mathrm{LN}(h+\delta) - W_U\mathrm{LN}(h) = \frac{\sqrt{d}}{\|\bar{h}\|}\, W_U\Big(I - \frac{\bar{h}\bar{h}^\top}{\|\bar{h}\|^2}\Big)\bar{\delta} \;+\; \eta, \qquad \|\eta\| \;=\; O\Big(\|W_U\|_{\mathrm{op}}\sqrt{d}\,\frac{\|\delta\|^2}{\|\bar{h}\|^2}\Big). \tag{10}$$

Proof. Centring is linear, so $\overline{h+\delta} = \bar{h} + \bar{\delta}$. The map $u(x) = \sqrt{d}\,x/\|x\|$ has Jacobian $Du(x) = \frac{\sqrt{d}}{\|x\|}\big(I - xx^\top/\|x\|^2\big)$, from differentiating $x/\|x\|$ as $I/\|x\| - xx^\top/\|x\|^3$. u is C^∞ away from the origin with $\|D^2u(x)\| = O(\|x\|^{-2})$, so the first-order Taylor expansion at $x = \bar{h}$ carries the stated remainder, and composing with the linear map W_U multiplies the bound by $\|W_U\|_{\mathrm{op}}$, the factor in equation 10. □

(i) The Jacobian annihilates $\bar{h}$, so a perturbation that only rescales the residual stream is invisible at first order. *(ii)* With prefactor $\sqrt{d}/\|\bar{h}\|$ the loss at first order sees δ only through the *relative* perturbation $\|\delta\|/\|\bar{h}\|$ and its direction. Since $\|h_t\|$ grows monotonically with depth in trained transformers, absolute perturbation norms at different depths are not comparable, so Appendix E.8 reports $\|\delta_t\|/\|h_t\|$ throughout. *(iii)* Setting $\delta = \epsilon$ and $h = h_L$ gives the boundary case excluded in §C.3.

C.5 Per-layer ranking and modularity

For a layer set $\mathcal{R} \subseteq \{0, \ldots, L-1\}$ let $D(\mathcal{R})$ be the damage of converting every layer in $\mathcal{R}$ jointly. Every set function admits the exact Möbius (finite-difference) expansion

$$D(\mathcal{R}) = \sum_{\ell\in\mathcal{R}} a_\ell + \sum_{\ell<m\in\mathcal{R}} c_{\ell m} + \sum_{|T|\geq 3,\, T\subseteq\mathcal{R}} m_T, \qquad a_\ell = D(\{\ell\}), \quad c_{\ell m} = D(\{\ell,m\}) - a_\ell - a_m. \tag{11}$$

This is an identity, not a model: the coefficients are successive finite differences of D itself.

Proposition 7. *If every* $D \in \mathcal{D}$ *is modular with the same coefficient vector* a*, then ranking layers by* $s = a$ *and returning the bottom* k *is the exact minimiser of* D *over all* k*-subsets, for every* k *and every* $D \in \mathcal{D}$*. The score must be order-equivalent to* a*: any strictly increasing transform of* a *induces the same bottom-*k *set, whereas an arbitrary* s *is not exact even against a modular* D*. Conversely, if some* $D \in \mathcal{D}$ *has a nonzero* pair *coefficient and* $\mathcal{D}$ *is closed under* unbounded *pair perturbations (for every such* D *and pair* $\{a,b\}$*, some one-parameter family in* $\mathcal{D}$ *varies* c_{ab} *over all of* $\mathbb{R}$ *with all other coefficients fixed), then no fixed score is exact at* $k = 2$ *across that family.*

Proof. ($\Leftarrow$) $D(\mathcal{R}) = \sum_{\ell\in\mathcal{R}} a_\ell$ makes minimising over k-subsets exactly taking the k smallest a_ℓ. ($\Rightarrow$) Suppose some $D \in \mathcal{D}$ has a pair coefficient $c_{ab} \neq 0$, and take $k = 2$: the rule returns the two lowest-s layers, independent of c_{ab}. Varying c_{ab} by τ along the family in $\mathcal{D}$, all other coefficients fixed, changes $D(\{a,b\})$ by τ and leaves every other 2-subset unchanged, so for $|\tau|$ large enough of the appropriate sign $\{a,b\}$ moves from unique minimiser to unique maximiser while the rule's output is constant: no fixed s is correct for both members of that one-parameter family. □

Unboundedness is the hypothesis the proof uses. A bounded family is a counterexample. An earlier version asked only that the family vary c_{ab} *through zero*, strictly weaker than what the proof consumes ($|\tau|$ large enough to move $\{a,b\}$ from minimiser to maximiser), and under it the converse is false: with $a = (1,2,3)$, every pair coefficient in $[-0.1, 0.1]$, no terms of order ≥ 3, every c_{ab} crosses zero, yet

$$D(\{1,2\}) \leq 3.1 \;<\; 3.9 \leq D(\{1,3\}), \qquad D(\{2,3\}) \geq 4.9,$$

so the three 2-subset values never overlap and ranking by a returns the optimal pair on every member. What defeats a fixed score is interaction mass large enough to close the gaps between self-term sums, not mass that merely changes sign.

The closure condition is necessary, not decorative: $D(\mathcal{R}) = \sum_{\ell \in \mathcal{R}} s_\ell + \lambda \binom{|\mathcal{R}|}{2}$ has a uniform pair term and is not modular, yet bottom-k is exact at every k because that term is constant over k-subsets. Pair-uniform interactions are exactly what the closure assumption rules out, and they are invisible to budget-fixed selection, the regime §G.1 prices separately.

Per-layer ranking is adequate exactly as far as the interaction mass is small, which is measurable: all 66 pairwise coefficients at Pythia-160M and 120 of 276 at Pythia-410M give $\overline{|c|}/\overline{|a|} = 0.215$ and 0.445 over the whole pair set, but 0.066 and 0.120 in the tercile of pairs with the smallest self-term sums $a_\ell + a_m$, the region a bottom-k selector searches, so adding the pairwise term does not improve selection (§G.1). The rule's form was not wrong. *what* was ranked was.

Corollary (the sign of the reversal). Proposition 7 only says a fixed score *can* fail. The observed direction (deepest-k wins at $L/2$) is itself a statement about the measured coefficients. For four blocks with self-terms $a_0 < a_1 < a_2 < a_3$ and no terms of order ≥ 3, the per-layer rule deletes $\{0, 1\}$ and deepest-k deletes $\{2, 3\}$ at $k = 2 = L/2$, and by equation 11

$$D(\{2,3\}) < D(\{0,1\}) \iff c_{23} - c_{01} < (a_0 + a_1) - (a_2 + a_3). \tag{12}$$

The right-hand side is negative because the deep self-terms are larger (per-layer damage grows toward the output end on the removal operations, §F.1), so the reversal needs the deep pair's interaction to fall below the shallow pair's by *more* than that gap, necessary, not sufficient: with self terms $(1, 2, 3, 4)$, c_{01}=0, c_{23}= − 1 the reversal still does not occur. At Pythia-160M the deepest six's pair term is $+0.476$ of its self sum (super-additive, above) yet the envelope still reverses, so the inequality resolves in favour of deepest-k on every family in Table G.1. A fixed per-layer price is exact at $k = 1$ (singletons carry no pair term) and increasingly decided by pair terms as k grows.

The same holds at the aggressive budget, where adjacent layers are forced into the set. Over the 403 measured 6-subsets of Pythia-160M ($k = L/2$, all 66 pairs known) the pairwise term improves *prediction* (Spearman with true damage rises from $+0.926$ to $+0.955$) but not *selection*: both surrogates return the pool optimum, and after removing it the same next-best one. Resampling the pool 500 times, the corrected surrogate picks a better subset 45 times and a worse one 201. The truncation does not converge: damage is super-additive ($1.476\times$ the sum of single-layer damages for the deepest six) but the fifteen pairwise terms over-count it, so $\sum a + \sum c$ over-predicts by 0.948 nats and the over-prediction grows with damage ($\rho = -0.836$ between residual and truth). Third and higher orders would cost more than the 924 subsets being ranked.

Is the remainder order-preserving? §4 argues a ranking needs R_3 co-monotone with damage, not merely small. Table C.2 measures both from the same symmetric-differencing fit ($R_3 = a_\ell - (A_\ell + B_\ell)$) at no extra cost. R_3 is largest on the two capability-removal operations and there most strongly co-monotone: across the twenty cells the induced ranking moves by less than 0.12 in nineteen. Four cells have $\rho(R_3, a) < 0$ (`quant4` and `quant3` on Pythia-1.4B. `prune` and `quant4` on GPT-2-medium) and only one moves by more than 0.12, `prune` on GPT-2-medium, one of the two cells where the correction does not improve (Appendix E.1).

Table C.2: The third-order remainder is large but order-preserving. $|R_3|/a$ is the per-layer median, $\rho(R_3, a)$ its rank correlation with true damage; the last columns rank by the second-order term alone against second order plus the residual $a_\ell - (\hat{A} + \hat{B})$.

model	operation	$\lvert R_3\rvert/a$	$\rho(R_3, a)$	$\rho(B, a)$	$\rho(B{+}R_3, a)$	change
Pythia-1.4B	`linear`	0.22	+0.857	+0.890	+0.992	+0.102
Pythia-1.4B	`prune`	0.28	+0.925	+0.950	+0.990	+0.039
Pythia-1.4B	`quant4`	0.02	−0.062	+0.943	+0.940	−0.003
Pythia-1.4B	`quant3`	0.03	−0.353	+0.974	+0.982	+0.008
GPT-2-medium	`linear`	0.17	+0.295	+0.615	+0.645	+0.030
GPT-2-medium	`prune`	0.15	**−0.169**	+0.557	+0.432	**−0.124**
GPT-2-medium	`quant4`	0.01	−0.257	+0.813	+0.803	−0.010

The complete per-cell grid is in the supplementary appendix (Table S8), with the released JSON/CSV.

C.6 Isotropic curvature concentrates

For a random direction $u \sim \mathrm{Uniform}(\mathcal{S}^{d-1})$ and a symmetric H with $\mathrm{tr}(H) \neq 0$, let $\kappa(u) = u^\top H u$, $\kappa^{\mathrm{iso}} = \mathbb{E}\,\kappa(u) = \mathrm{tr}(H)/d$, and $r = \mathrm{tr}(H)^2/\mathrm{tr}(H^2)$. Then

$$\mathrm{Var}\left(\frac{\kappa(u)}{\kappa^{\mathrm{iso}}}\right) = \frac{2(d-r)}{r(d+2)} \leq \frac{2}{r}, \tag{13}$$

and by Chebyshev, for every $t > 0$,

$$\Pr\left(\left|\frac{\kappa(u)}{\kappa^{\mathrm{iso}}} - 1\right| \geq t\right) \;\leq\; \frac{2(d-r)}{r(d+2)t^2} \;\leq\; \frac{2}{r\,t^2}. \tag{14}$$

Proof. Write $H = \sum_i \lambda_i v_i v_i^\top$ in an orthonormal eigenbasis and $u_i = \langle u, v_i \rangle$. For u uniform on the sphere, $\mathbb{E}[u_i^2] = 1/d$ and $\mathbb{E}[u_i^2 u_j^2] = (1 + 2\delta_{ij})/(d(d+2))$, so $\mathbb{E}[\kappa(u)] = \mathrm{tr}(H)/d$ and $\mathbb{E}[\kappa(u)^2] = (\mathrm{tr}(H)^2 + 2\,\mathrm{tr}(H^2))/(d(d+2))$. Subtracting $(\mathrm{tr}(H)/d)^2$ gives equation 13. The upper bound uses $d - r \leq d + 2$, and Chebyshev gives equation 14. □

This is the mechanism behind the family split: a weight-noise perturbation is near-isotropic, so its directional curvature departs from κ^{iso} by more than a relative margin t with probability at most $2/(r\,t^2)$, a $\pm\sqrt{2/(r\delta)}$ envelope at confidence $1 - \delta$. A capability-removal perturbation is a learned low-dimensional direction and need not be.

Proposition 8. *For independent directions $u_1, \ldots, u_r$ and sample mean $\bar{\kappa} = \frac{1}{r}\sum_{j=1}^{r} u_j^\top H u_j$,*

$$\mathrm{Var}\left(\frac{\bar{\kappa}}{\kappa^{\mathrm{iso}}}\right) \;\leq\; \frac{2}{r_{\mathrm{eff}}\,r}, \qquad \Pr\left(\left|\frac{\bar{\kappa}}{\kappa^{\mathrm{iso}}} - 1\right| \geq t\right) \;\leq\; \frac{2}{r_{\mathrm{eff}}\,r\,t^2}. \tag{15}$$

Proof. Apply equation 13 to each independent direction and divide the variance by r. □

Proposition 9. *Let ϵ be a random perturbation direction with second moment $\Sigma = \mathbb{E}[\epsilon\epsilon^\top]$, and let H be symmetric with $\mathrm{tr}(H) \neq 0$. Then*

$$\mathbb{E}\left[\frac{\epsilon^\top H \epsilon}{\kappa^{\mathrm{iso}}}\right] = \frac{d\;\mathrm{tr}(H\Sigma)}{\mathrm{tr}(H)}, \qquad \kappa^{\mathrm{iso}} = \mathrm{tr}(H)/d. \tag{16}$$

Proof. $\mathbb{E}[\epsilon^\top H \epsilon] = \mathrm{tr}(H\,\mathbb{E}[\epsilon\epsilon^\top]) = \mathrm{tr}(H\Sigma)$, and $\kappa^{\mathrm{iso}} = \mathrm{tr}(H)/d$. □

Proposition 10 (A rate mismatch opens an inversion window)**.** *Write $x_\ell = \ell/(L-1)$ and suppose the curvature profile is exponential in relative depth, $\kappa_\ell^{\mathrm{iso}} = Ae^{-\hat{c}\,x_\ell}$ with $A > 0$. Let a fixed-rate depth prior assume rate c_0, scoring $p_\ell = \frac{1}{2}\|\epsilon_\ell\|^2 e^{-c_0 x_\ell}$ where the corrected score is $s_\ell = \frac{1}{2}\|\epsilon_\ell\|^2 \kappa_\ell^{\mathrm{iso}}$. For any two layers ℓ, m with $\Delta = x_m - x_\ell > 0$ and magnitude ratio $r = \|\epsilon_m\|^2/\|\epsilon_\ell\|^2$, the two scores order the pair* oppositely *if and only if*

$$\min(\hat{c}, c_0)\,\Delta \;<\; \log r \;<\; \max(\hat{c}, c_0)\,\Delta. \tag{17}$$

The set of log-ratios that invert the pair is therefore an interval of length $|\hat{c} - c_0|\,\Delta$: empty if and only if $\hat{c} = c_0$, and widening linearly in both the rate mismatch and the depth separation.

Proof. $s_m > s_\ell \iff \|\epsilon_m\|^2 e^{-\hat{c}x_m} > \|\epsilon_\ell\|^2 e^{-\hat{c}x_\ell} \iff \log r > \hat{c}\,\Delta$, and identically $p_m > p_\ell \iff \log r > c_0\Delta$. The orderings differ exactly when $\log r$ separates $\hat{c}\Delta$ from $c_0\Delta$, which is Eq. 17, of length $|\hat{c}\Delta - c_0\Delta|$. Ties are measure-zero and assigned either way without affecting the length. □

What this says, and what it does not. It licenses measuring a *rate*: the window closes exactly when $\hat{c} = c_0$ and widens in proportion to $|\hat{c} - c_0|$, so whether a fixed-rate depth prior can be trusted on a checkpoint is decided by the rate mismatch, not by the similarity of the two profiles' shapes. This is the theoretical counterpart of an empirical null in §I.1: rank agreement between κ^{iso} and the prior is at chance as an audit ($12/17$ leave-one-out), Spearman being scale- and rate-invariant and blind to the quantity Eq. 17 identifies, while thresholding $\hat{c}$ against the prior's own $c_0 = 3$ reaches $12/15$. It does *not* predict how many pairs invert, which depends on where the realised $\log r$ fall relative to the window. In the stylised case of an exponential magnitude, $\|\epsilon_\ell\|^2 = Be^{b\,x_\ell}$, Eq. 17 becomes $\min(\hat{c}, c_0) < b < \max(\hat{c}, c_0)$, independent of Δ, so the prior reproduces the corrected ranking exactly or reverses it entirely. As a prediction of the *sign* of $\rho(\mathrm{prior}, a)$ that form is right on ten of fifteen checkpoints, the five misses all having $|b - c_0| < 0.5$, where the idealisation is weakest and the predicted reversal marginal.

Table D.1: **What every number in this paper is measured on.** Per checkpoint: depth, the substitution grid's sequence budget, and the operations with a released per-layer measurement. Corpus: the held-out WikiText-103 stream of §3, fp32 for every curvature reading, bf16 for the efficiency measurements, seeds 0, 1, 2. OPT-6.7B carries four operations rather than five (no low-rank cell released), so the extended grid of Table E.5 has 128 cells rather than 129.

checkpoint	layers	seq len	sequences	operations released
Pythia-160M	12	1024	32	five core
Pythia-410M	24	1024	32	five core
Pythia-1.4B	24	1024	32	five core, plus three registered
GPT-2-medium	24	1024	32	five core, plus three registered
Qwen2.5-0.5B	24	1024	32	five core, plus three registered
TinyLlama-1.1B	22	1024	32	five core, plus three registered
GPT-J-6B	28	1024	32	five core
Qwen2.5-7B	28	1024	32	five core, plus three registered
Llama-3-8B	32	1024	32	five core
Qwen3-8B	36	1024	32	five core
OLMoE-1B-7B	16	1024	32	five core
LLaMA-13B	40	1024	32	five core
Gemma-2-9B	42	1024	32	five core
Mistral-7B-v0.3	32	1024	32	five core
OPT-6.7B	32	1024	32	four (no low-rank)
OLMo-1-1B	16	1024	32	five core
OLMo-1-7B	32	1024	32	five core
Bloom-7B1	30	1024	32	five core

Table D.2: **The checkpoint cohorts.** Every analysis is one of these four sets or a subset of one; the core 11×5 grid of Table E.7 and the *every operation* subset of §5 are such subsets.

cohort	n	composition
substitution grid	18	the checkpoints listed in §3, five operations each
audit-scored	18	the grid; Bloom-7B1 joins on the fixed harness, where its probe passes the operation-based check (Appendix F.7)
normaliser split	20	the profile minus the two Gemma-2 checkpoints, whose curvature rises with depth
curvature profile	22	the grid plus the four profile-only checkpoints

D Operations, estimator and protocol

D.1 Evaluation protocol and checkpoint cohorts

D.2 The six further operations

The second block of Table E.7 adds six operations to the five defined in §3. Every one of them acts at the same site (it changes layer ℓ's attention output and nothing else), which is what makes a single profile applicable to all of them. Their family registrations and the outcome of those registrations are in Appendices A.2(C) and F.3. This section gives only the definitions.

`sparse24` 2:4 structured sparsity on the attention projections: in each group of four weights along the input dimension the two smallest by magnitude are zeroed. Half the weights are gone. The functional form (softmax over all positions, all heads) is untouched.

`kvquant` int4 quantisation of the K and V tensors only, per-head absmax, weights untouched. Nothing structural changes. This is the deployment-relevant member of the set: KV-cache quantisation ships in serving stacks.

`headmerge` GQA-style head merging: K and V are averaged across adjacent head pairs and queries are left alone, so every weight is still used and softmax is intact while head-specific retrieval is gone.

`swa` sliding-window attention at $w = 128$: softmax and every weight are kept and only long-range retrieval is removed.

`headprune` the half of the attention heads with the smallest output contribution are zeroed: capability removal at a finer granularity than deleting the whole sublayer.

`actquant` int4 quantisation of the activations entering the attention module, every weight untouched. It is noise, as weight quantisation is, but it enters through activations instead.

`sparse24` and `headmerge` are the discriminating pair. Under a naive "how much capacity is removed" reading they point in the opposite direction from the eliminate-or-perturb rule of §F.1, so registering them is a test of the rule rather than a restatement of it.

D.3 DO THE ADAPTERS REPRODUCE THEIR MODELS' ATTENTION?

Every operation-dependent number here rests on a reconstruction: the measurement hooks a block's attention module, captures its input and output y_{sm}, and rebuilds the attention from the adapter's `qkv` and `out_proj` into the substituted output and the displacement ϵ_ℓ, which is a difference between two different things unless y_{sm} is reproduced under ordinary softmax attention. The Reproducibility Statement asserts that precondition. A per-architecture check came late, when one adapter failed.

The check. One short batch in fp32, three layers per model, the adapter's reconstruction against the module's own output along the code path `09_transfer.py` uses. Aliased adapters (Qwen2, Gemma-2 and Mistral → Llama, Qwen3-MoE → Qwen3) give seven distinct `qkv` paths covering eighteen checkpoints, each validated by the smallest model that exercises it.

path	validated on	pos. emb. reached	max relative error
`llama`	TinyLlama-1.1B	yes	0.000
`gpt_neox`	Pythia-1.4B	yes	0.000
`qwen3`	Qwen3-8B	yes	0.000
`olmoe`	OLMoE-1B-7B	yes	0.000
`gpt2`	GPT-2-medium	n/a (learned)	0.000
`gptj`	GPT-J-6B	yes	1.2×10^{-6}
`bloom`	—	—	exempt: ALiBi is not causal softmax

The failure, and the fix. The `gptj` adapter declared `uses_rope=False` and applied no rotation, though GPT-J uses partial rotary embeddings: round-trip error 0.78 against a tolerance of 10^{-3}. Applying GPT-J's interleaved partial rotary (`rotary_dim` $= 64$ of 128) from the module's own position table takes it to 1.2×10^{-6}.

What the failure reached. κ^{iso} (a normalised Rayleigh quotient: a random probe scaled to $\|\epsilon\|$, divided by $\|\epsilon\|^2$) reaches the reconstruction only through that norm, so norm-invariance predicts it barely moves. It does not, GPT-J reads $\hat{c} = 2.365, 2.373, 2.333, 2.372$ across four code paths including the fixed one, and the era split, the boundary and EXP89 do not turn on the bug. The operation-dependent quantities do: on the corrected adapter GPT-J's five-operation grid moved the linearisation cell's $\Delta\rho$ from -0.535 to -0.544 and the checkpoint's mean from -0.034 to -0.054, all five operations still improving over $\|\epsilon\|^2$, the audit label unchanged at *prior ok*, the linearisation *sweep* stale and withdrawn (Appendix D.4).

What the fix does *not* propagate to. The eleven-model grid of §5, measured between July and August before the evaluation-batching fix of Appendix I.1, carries no document-start token. The corrected GPT-J grid, measured after it, does, its clean NLL moving from 2.4184 to 2.4276, which an attention reconstruction cannot cause and a leading token can. The corrections were separated: `09_transfer.py` now takes `--no-bos`, and GPT-J was measured a third time on the corrected adapter under the grid's own protocol (`substitutions_gptj6b_unified.json`, clean NLL 2.41822, the no-token value, which confirms the protocol).

GPT-J grid	clean NLL	adapter	token	lin. $\Delta\rho$	mean $\Delta\rho$
pre-fix run, file superseded	2.41838	pre-fix	none	-0.535	-0.034
`gptj6b_fixed`	2.42762	corrected	BOS	-0.544	-0.054
`gptj6b_unified`	2.41822	corrected	none	-0.600	-0.082

Across all three variants: all five GPT-J operations improve over $\|\epsilon\|^2$, the 53 of 55 core-grid count is unchanged, the audit label is *prior ok* in every one, and no per-operation mean of Table E.7 moves by more than 0.014 (`lowrank`. The other four by at most 0.006). The corrections compound rather than cancel, so the unified grid sits furthest from the unsafe boundary.

Which variant the paper reports. `substitutions_gptj6b.json`, the file the grid is read from, carries the corrected run, the pre-fix measurement was replaced under that name when the adapter was fixed, so a reproducer reading the released tree gets 2.41822 and the corrected cells, and `analysis/gptj_protocol_variants.py` prints the first and third rows above identically. Shipping both files would have been better. The pre-fix numbers are recorded so the correction can be priced rather than trusted, substituting a checkpoint's cells being safe only because that aggregate is invariant across the three variants, checked rather than asserted.

Why this is reported rather than quietly fixed. A silent reconstruction error is the failure mode this method can least detect on its own, every downstream number staying plausible. The check is cheap, is now a gate on the measurement stack, and caught one architecture.

D.4 How reproducible is a 300-step alignment?

All linearisation numbers follow 300 alignment steps on the replaced layers; we measure that procedure's reproducibility cost.

Measurement. On Qwen2.5-7B at k=7 we repeated the whole pipeline in a fresh process. There the corrected score and the free depth prior select *identical* layers ($\{1, 2, 4, 5, 15, 16, 17\}$, §6); their arms are one artefact measured twice, any difference procedure noise. Four draws gave ΔNLL at 1024 tokens of 0.19, 0.46, 0.57, 0.19, spread 0.38 nats, where the model is compared at margins of 0.04 to 0.16. The published within-run seed spread 0.016–0.026 (Table G.10) misses this: within-process seed variation moves batch order and alignment sampling, not the between-process accumulation order.

Source. Dropout is zero, so the seed is not the source; what remains is bf16 non-determinism in the backward pass, amplified over 300 distillation steps whose KL starts near 10^3: two runs descend one objective, stopping at different points. Raising the step count, lowering the learning rate, fixing the batch set or aligning in fp32 would each shrink it; we report it for the configuration the paper ran, not a chosen-small one.

Invalidated. Against a ± 0.4 nat band, 44 of 48 arm-versus-deepest-k margins fall inside; one survivor is countable, the $L/2$ collapse on Qwen2.5-7B at $+3.16$. GPT-J-6B's $+2.78$ is stale, not counted: its sweep predates Appendix D.3's adapter correction; re-measuring its five-operation grid on the corrected adapter moved rank correlations under 0.02 with the audit label unchanged, yet the corrected profile selects one different layer at each budget (k=7, k=14), so the sweep is stale. Re-running needs two concurrent fp32 copies of a 6B model, beyond the 48 GB available; withdrawn. So the $L/2$ reversal of §7 is robust, the $L/4$ margins are not: §G.1's worst $L/4$ margin, 0.08 nats, reads as "no arm separates at $L/4$", not a ranking. §G.1's headline count $(22/24)$ uses only the three deterministic operations for this reason; attention linearisation is reported separately as unresolved cells.

Unaffected. Deletion, low-rank truncation and quantisation are one-shot: no alignment, no optimiser, deterministic given checkpoint and evaluation batches. That covers §5's per-layer grid, §G.1's results on those three operations, §7's operating envelope, every downstream table including Table 1, §6's quantisation, and the curvature profile itself, two forward passes, no training. The finding narrows one operation's selection claims, not the factorisation, profile, or gate.

D.5 The forward-only estimator against backpropagation

Table D.3: The same curvature quantity computed two ways on Pythia-410M, same layers, batches and probe vectors. The forward-only symmetric-difference route agrees with backpropagation Hessian–vector products (Hutchinson) at $\rho = 0.918$ on κ^{iso} (median relative error 0.037), while using 2.35× less peak memory and 2.01× less wall time. The check was repeated on Qwen2.5-0.5B, an RMSNorm checkpoint, at four probes: $\rho = 0.993$ on κ^{iso} (median relative error 0.014), 2.07× less time and 2.84× less memory (`hvp_qwen05.json`).

route	$\rho(\kappa^{\text{iso}})$	median rel. err	time (s)	peak mem (MB)
forward-only (ours)	0.918	0.037	81.0	1867
backprop HVP	—	—	162.8	4395
ratio	—	—	2.01×	2.35×

D.6 Probe budget and estimator stability

The profile's default is one Gaussian probe per layer, the smallest budget that could work. Table D.4 answers whether that is a measurement or a draw: a pool of probes is measured once and subsampled offline, so every row after the first is free. Split-half reliability is 0.878 at r=1 (0.935 after Spearman–Brown correction) and the correction result is flat from r=2. The residual error is therefore carried by the isotropic approximation, not by the probe budget.

D.7 Norm invariance of the probe profile

§4 fixes the probe norm by the per-layer linearisation error under one reference operation, not the operation under test. Table D.5 checks both consequences. The layer *ordering*, all a scale-invariant rank correlation can use, is norm-independent: profiles at $c = 0.1$ and $c = 0.3$ of the clean output norm, independent probes, agree at $\rho = 0.79$–0.999. The clean-network norm $c\|y_\ell\|$ costs nothing: the 30-cell result is $28/30$ at either c, matching the reference-norm probes behind the reported results.

Table D.4: Estimator variance on Pythia-410M. "probes/layer" is the total per-layer probe evaluations. Split-half ρ is between two disjoint probe subsets (Spearman–Brown corrected). Cells-improved is the curvature-correction result at that budget. The curve is flat from r=2.

probes/layer	split-half ρ	SB-corrected	cells impr.
1	0.878	0.935	4.6/5
2	0.873	0.932	4.9/5
4	0.924	0.961	4.9/5
8	0.958	0.978	5.0/5
16	0.972	0.986	5.0/5
32	0.986	0.993	5.0/5

Llama-3-8B's strict profile re-measured at the clean norm shifts the ordering by $\rho = 0.98$–1.00 and the strict correlation by ≤ 0.06. The reference scale is a convenience, not what drives the result.

The floor is arithmetic, not method: B comes from $\mathcal{L}(t) + \mathcal{L}(-t) - 2\mathcal{L}(0)$, nearly equal fp32 losses whose difference underflows below some norm. We report the B-spread diagnostic of Table D.5 rather than tune c to taste: a run whose spread is large is unusable at any c. $c \in [0.1, 0.3]$ clears the floor on every model measured.

The ordering is not the quantity this paper thresholds. A reviewer noted that everything above tests layer *ordering*: two exponential profiles can order identically while their fitted rates differ by a factor, so the check as run does not license $\hat{c}$. Correct: we fit $\hat{c}$ at each c on the same released profiles:

	c=0.03	c=0.1	c=0.3	released
Pythia-160M	1.74	1.79	1.75	1.82
Pythia-410M	−0.32	0.15	0.39	0.46
Pythia-1.4B	0.73	1.17	1.21	1.06

Inside the usable range the rate moves by ≤ 0.23 (Pythia-410M), 0.04 elsewhere. Released values, whose probe norm is the substitution's $\|\epsilon_\ell\|$, sit at the c=0.3 end, within 0.07, 0.07, 0.15: the substitution-matched norm does not produce the rate. Below the floor the estimate collapses where the B-spread diagnostic says it should: at c=0.03 Pythia-410M reads -0.32, the wrong *sign*, so a rate read there would fail §I.1's sign arm on an arithmetic artefact, the strongest reason we publish the floor alongside the estimator. Pythia-410M's 0.23 spread is half its 0.46: larger than the OLS interval suggests, small against the 3.6 between the two group means, and every checkpoint stays on its own side of c_0. Only the checkpoints nearest c_0 turn on a small change (OPT-1.3B 2.92 ± 0.35, Qwen2.5-7B 3.18 ± 0.21, OPT-6.7B 3.75 ± 0.24), where the 0.23 probe-scale spread is comparable to their own standard errors: probe scale alone does not decide which side a borderline checkpoint lands on, and the audit abstains rather than picks (Appendix I.1).

Table D.5: The probe norm does not set the profile's layer ordering. "invariance" is the Spearman between profiles measured at $c = 0.1$ and $c = 0.3$ with *independent* probes; "vs. matched" is agreement with the substitution-matched profile; "cells" is the correction result at $c = 0.1$. The last column is the median spread of B between the two symmetric offsets at $c = 0.03$, the estimator's own contamination diagnostic: on the three Pythia models it rises by an order of magnitude, where the $c = 0.03$ ordering also degrades (invariance against $c = 0.3$ falls to 0.26 at 410M). The floor is fp32 cancellation in $\mathcal{L}(t) + \mathcal{L}(-t)$, not a dependence of the profile on the norm; $c \in [0.1, 0.3]$ sits above it on every model.

model	invariance (c: 0.1 vs 0.3)	vs. matched	cells	B spread at c=0.03
Pythia-160M	0.993	0.993	5/5	0.79
Pythia-410M	0.791	0.778	4/5	0.87
Pythia-1.4B	0.997	0.942	5/5	1.68
GPT-2-m	0.974	0.932	4/5	0.39
Qwen2.5-0.5B	0.999	0.997	5/5	0.05
TinyLlama	0.999	0.999	5/5	0.05
30-cell result, fixed norm c=0.1			**28**/30	mean $\rho = 0.660$
30-cell result, fixed norm c=0.3			**28**/30	mean $\rho = 0.658$
30-cell result, substitution-matched			**28**/30	mean $\rho = 0.657$

Not an artefact of the calibration corpus. We measure the profile on held-out WikiText-103 throughout. No corpus change we can make moves the layer selection. On a disjoint corpus (Pile

Table D.6: The profile transfers across calibration corpora. Columns (a) and (b) measure and evaluate on the same corpus; (c) measures on WikiText-103 and evaluates damage on the Pile. The transfer gap $b - c$ is what a practitioner pays for reusing a profile measured elsewhere.

model	op.	(a) WT→WT	(b) Pile→Pile	(c) WT→Pile	gap $b - c$
Pythia-410M	`quant4`	$+0.956$	$+0.977$	$+0.974$	$+0.004$
	`lowrank`	$+0.616$	$+0.354$	$+0.354$	0.000
Qwen2.5-0.5B	`quant4`	$+0.800$	$+0.824$	$+0.813$	$+0.011$
	`lowrank`	$+0.868$	$+0.864$	$+0.873$	-0.009
profile shape agreement, WT vs Pile		$\rho = 0.978$ (410M), 0.988 (Qwen2.5-0.5B)			

split, same probe count and norm) the ordering shifts by only $\rho = 0.978$ (Pythia-410M), 0.988 (Qwen2.5-0.5B) against the WikiText-measured profile of the same checkpoint. On GSM8K's own mathematical text, far further from WikiText, Llama-3-8B's ordering agrees at $\rho = 0.88$–0.98 and the selection it feeds moves only by tie-breaking (Appendix E.6). The reference point is not 1 but the 0.91–0.99 spread between WikiText calibration segments of the same checkpoint. Every corpus we tried lands inside it. The competing selector is equally unmoved (Block Influence $\rho = 0.993$–0.994 between corpora against its own 0.996–0.999), which matters, the alternative explanation for its GSM8K win being better text (Appendix E.6). The *transfer gap* in Table D.6, Pile damage predicted from the Pile-measured profile against the WikiText-measured one, is within ± 0.012 in all four model–operation cells and exactly zero in one: a profile measured on whichever corpus is convenient loses nothing, as the factorisation predicts, $\kappa_\ell^{\mathrm{iso}} = \mathrm{tr}(H_\ell)/d$ a network property, calibration data entering only through the Hessian's evaluation point.

The same property on C4. The check above varies the corpus, not the operation. A stronger version holds the profile fixed and sweeps the whole grid. Table D.7 repeats the main text's 8×5 grid on a held-out C4 subset, three of the eight checkpoints printed (6000 documents from one shard. Same 32×1024 probe protocol. The complete grid is Table S4), measuring κ^{iso} once on WikiText and reporting the transfer Spearman $\rho(\frac{1}{2}\|\epsilon_\ell\|^2 \kappa_\ell^{\mathrm{iso}}, \Delta\mathrm{NLL}_\ell)$ on WikiText and on C4. Nothing needs re-fitting: the weighted score beats the magnitude-only score $\rho(\frac{1}{2}\|\epsilon_\ell\|^2, \Delta\mathrm{NLL}_\ell)$ in 39 of 40 cells on C4 against 38 of 40 on WikiText, though slightly weaker on the held-out corpus (mean ρ $0.627 \to 0.575$, mean change -0.052, median -0.049). The loss is concentrated: seven cells carry 94% of it, the one that stops helping entirely is GPT-2-medium `prune` ($+0.44 \to -0.04$), the rest individual cells (TinyLlama `prune` and `lowrank`, Qwen2.5-7B `quant3`, GPT-2-medium `linear`), not a systematic operation or scale.

Table D.7: **The WikiText profile, applied to C4.** Columns (a) and (c′) use the same WikiText-measured profile; (a) evaluates per-layer damage on WikiText, (c′) on a held-out C4 subset. $\Delta = (c') - (a)$. The curvature-weighted score beats the magnitude-only score in $38/40$ cells on WikiText and $39/40$ on C4.

model	op.	(a) WT→WT	(c′) WT→C4	Δ
GPT-2-medium	`linear`	$+0.179$	-0.043	-0.223
	`quant4`	$+0.592$	$+0.510$	-0.082
	`quant3`	$+0.380$	$+0.506$	$+0.126$
	`prune`	$+0.442$	-0.044	-0.486
	`lowrank`	$+0.493$	$+0.350$	-0.143
TinyLlama	`linear`	$+0.623$	$+0.532$	-0.090
	`quant4`	$+0.607$	$+0.578$	-0.029
	`quant3`	$+0.813$	$+0.809$	-0.003
	`prune`	$+0.752$	$+0.394$	-0.358
	`lowrank`	$+0.816$	$+0.549$	-0.267
Qwen2.5-7B	`linear`	$+0.335$	$+0.178$	-0.157
	`quant4`	$+0.590$	$+0.519$	-0.071
	`quant3`	$+0.691$	$+0.389$	-0.302
	`prune`	$+0.548$	$+0.477$	-0.071
	`lowrank`	$+0.514$	$+0.592$	$+0.078$

The complete per-cell grid is in the supplementary appendix (Table S4), with the released JSON/CSV.

D.8 Measurement reliability

Every correlation in this paper is bounded above by how reliably a_ℓ itself is measured, so we measure that first. Splitting the calibration sequences in half, scoring a_ℓ independently on each half and

correcting with Spearman–Brown puts the reliability of the per-layer damage at **0.93–0.99** across the grid, and at 0.94–0.99 on GPT-2-medium, the checkpoint whose cross-operation structure is the odd one out (Appendix G.3), so its anomaly is not a noisier measurement. Disattenuating for that reliability moves GPT-2-medium's `linear–quant4` correlation from $+0.437$ to $+0.454$, i.e. by less than the rounding of the claims that rest on it. At the 32-sequence budget the same figure is what licenses the cost comparison of Appendix D.10: direct measurement is not paying for its accuracy with a larger corpus. Per-checkpoint halves are released in `substitutions_*.json`.

D.9 Accuracy against the calibration budget

§7 states that direct measurement is the quantity the profile approximates. This is that comparison. Both routes get the same budget B for one operation (deletion) and are scored against the full-budget oracle: the direct route measures each layer's damage on B sequences, the profile route reads κ^{iso} on the same B sequences and combines it with the perturbation magnitude.

Table D.8: **Spearman against the oracle, by calibration budget.** One operation (deletion). Qwen2.5-7B carries the profile at its reported 32-sequence budget. Pythia-410M is swept at every budget and reported at one probe, the other probe counts moving by at most 0.01. "Jaccard" is the overlap of the selected top-k set with the oracle's.

model	route	16 seqs	32 seqs	64 seqs	top-k Jaccard
Qwen2.5-7B	direct	0.966	0.975	0.994	—
	profile	—	0.569	—	—
Pythia-410M	direct	0.997	0.997	0.999	1.00
	profile	0.137	0.129	0.134	0.33

Direct measurement wins at every budget on both checkpoints, and on Pythia-410M by a wide margin (0.997 against 0.13), its top-k set matching the oracle's exactly where the profile's shares a third of it. Reading more sequences does not close the gap: the profile's correlation is flat from 16 to 64 sequences, because what limits it is the isotropic approximation, not the sample. That is the honest form of the amortisation argument, the profile is not the better estimator at any budget, it is the one that can be read before the operation is chosen. Reproduce with `analysis/verify_budgetaccuracy.py`, which re-derives every entry from the released `p1b_410m.json` and `p1b_qwen7b.json` and exits non-zero on any mismatch.

D.10 GPU cost of the estimator

The whole procedure, with every input named. The profile is the only operation-free part, so we give the algorithm in full. Pass counts are per calibration batch, L layers, r probes, and the calibration budget multiplies all rows equally.

step	forwards	needs an operation?	how often
curvature profile $\kappa_\ell^{\mathrm{iso}}$	$2Lr$	**no**	once per checkpoint
perturbation magnitude $\|\epsilon_\ell\|$	1, every layer at once	yes	once per operation
validity gate R_3	$2L$	**yes**, and it consumes a_ℓ	once per operation
direct damage a_ℓ (the alternative)	L	yes	once per operation

Figure D.1 draws the two readings and which is reused.

The magnitude row is one pass because a single-layer substitution sees the clean input h_ℓ: one clean forward per batch, with the substituted module evaluated beside each layer on the input that pass gives it, returns $\|\epsilon_\ell\|$ at every layer. Against the per-layer route in the same process it agrees exactly on all eleven operations at five checkpoints (55 of 55, relative difference 0). It reproduces the released $\|\epsilon_\ell\|$ to 6.5×10^{-4} on four of them and to 4.2×10^{-3} on TinyLlama's five core operations, rank order unchanged (Spearman 1.000; `115_onepass_eps.py`). The grid scripts measure $\|\epsilon_\ell\|$ inside the per-layer runs only because they need a_ℓ there anyway.

The gate is not operation-free: by the $t{=}1$ identity of Appendix C.1 its numerator *is* a_ℓ, so running it costs at least as much as measuring that operation's per-layer damage. Only $\kappa_\ell^{\mathrm{iso}}$ can be read before an operation is chosen.

At matched budget, each route at the precision it needs, the profile route is ahead from the eighth operation. EXP123 (`prereg_fp32_cost.md`, `123_fp32_profile_cost.py`, `verify_fp32_cost.py`) times the reads themselves on Qwen2.5-7B, on the released protocol (32 sequences of 1024 tokens), every step on one card, CUDA-synchronised, model loading excluded. In fp32 the isotropic profile costs 705.1 s (912 forward passes), the same profile with the clean-output

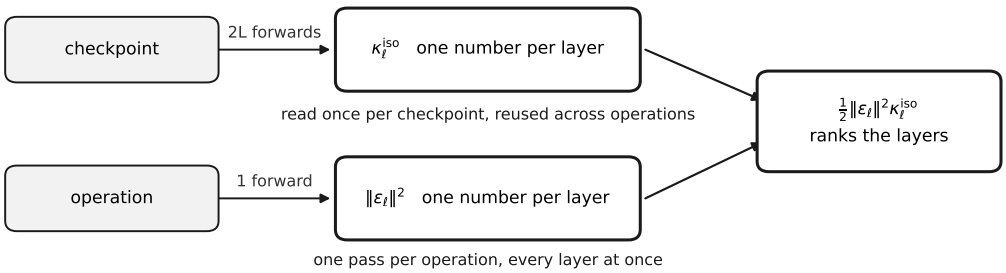


Figure D.1: **The two readings, and which of them is reused.** The profile is a property of the checkpoint, read once and before any operation is named. The displacement belongs to the operation and costs one pass, every layer at once. Only the first is amortised across operations, which is the ordering claim of §4. This section prices it: measuring directly stays the more accurate route, and is the cheaper one only for a single operation on a checkpoint whose profile has not been read.

step 703.3 s and the directional read 694.3 s, and each is the profile the paper uses: it correlates with the released file at $\rho = 0.993$, 0.991 and 1.000. Direct measurement of deletion costs 99.7 s per operation and the one-pass displacement 3.57 s, both in bf16, the precision they are deployed in. The profile route breaks even at 7.3 operations (7.2 for the directional read) and is ahead from the eighth; over the eleven-operation grid it costs 744 against 1097 s. Timed in fp32 as well, direct measurement costs 344.9 s per operation, the break-even falls to 2.1 operations and the eleven cost 841 against 3794 s. Either way a shipped profile makes each operation 27.9× cheaper, because that ratio contains no read. EXP116 (`116_decision_cost_onepass.py`) had priced the profile as $2L$ bf16 forward passes, 203.1 s, a read bf16 cannot perform: read in bf16, the profile correlates with the released one at $\rho = -0.56$, and 21 of 28 layers return a non-positive curvature. An earlier accounting charged the magnitude L passes per operation, as the grid scripts pay it, and found the order reversed at equal budget (1376 against 1199 s, `exp84b_fair_cost_7b.py`); that term is what the one pass replaces. Direct measurement is still the more accurate estimator, with split-half reliability 0.93–0.99 at this budget (Appendix D.8), so the comparison prices a ranking, not the same ranking. A practitioner with one operation already chosen, on a checkpoint whose profile nobody has read, should measure it.

That comparison prices operations that are cheap to *apply*. Measuring a_ℓ costs L evaluations *plus* L applications of the operation, and the paragraph above charges nothing for the second term because `prune` zeroes a sublayer and round-to-nearest rounds a tensor. A real post-training quantiser is not free to apply: deciding where to run GPTQ by measuring means running GPTQ L times, which is why Appendix H.2 scores under the RTN proxy rather than under the deployed quantiser. EXP111 prices that term (`111_decision_cost.py`, `decision_cost_qwen7b.json`). On Qwen2.5-7B on one RTX 4090, at the budget of Appendix H.2: a forward pass is 115.3 ms, the 32-sequence evaluation 3.88 s, and GPTQ on one block 3.03 s, of which 1.96 s is calibration and 1.06 s the damped Cholesky and column sweep.

The direct arm is given the stronger accounting rather than the obvious one. A competent pipeline hooks every block at once and collects all the calibration statistics in a single sweep, so that sweep (1.90 s) is paid once and only the per-block remainder (0.06 s) is paid L times. Charging every block its own calibration would inflate the baseline by 36% and is not what an implementer would do.

Three figures in that table are checks rather than new claims. EXP116's L-pass per-operation term, 101.6 s, reproduces EXP111's 103.3 s; its profile, 203.1 s, reproduces EXP111's 206.6 s and the 209.5 s timed separately below; so the two profile rows differ in the magnitude term alone.

What this does and does not establish. It does not make the profile the better estimator: at this budget direct measurement still reaches $\rho = 0.975$ against the oracle where the profile reaches 0.569 (Appendix D.9). It does make the profile the cheaper route to a ranking once more than one operation is weighed: level at two and ahead from the third for deletion, ahead from the second under GPTQ, whether or not the operation is expensive to apply. A profile published with the checkpoint is ahead from the first, 28× for deletion (106.3 against 3.8 s) and 29× for GPTQ (138.3 against 4.8 s): its 203.1 s is paid once, by whoever releases the model.

On a V100S-32GB at Pythia-410M a forward pass takes 155.4 ms (seq-len 1024, batch 2), so a one-per-model profile at r=1 ($2L = 48$ forwards *per batch*, ≈ 7.5 s) and per-substitution magnitude

Table D.9: **What it costs to decide *where* to apply an operation**, Qwen2.5-7B, $L = 28$, one RTX 4090, at the budget of Appendix H.2. The profile is read once per checkpoint; each operation then pays one forward pass per batch for its displacement (EXP116), or L under the accounting the grid scripts use (EXP111, kept for comparison). Direct measurement pays the operation and an evaluation at every layer, with its calibration shared across layers. K is the number of operations the decision ranges over. The one-pass row gives deletion and GPTQ either side of the slash, and the break-even column is against it.

route	once	per operation	K to break even	K=11
profile, one-pass magnitude (EXP116)	203.1 s	3.8 / 4.8 s	—	245 / 256 s
profile, L-pass magnitude (EXP111)	206.6 s	103.3 s	—	1343 s
direct, `prune` (free to apply)	—	106.3 s	**2.0**	1169 s
direct, GPTQ (3.03 s per block)	—	138.3 s	**1.5**	1522 s

($L = 24$ as the grid scripts run it, ≈ 3.7 s) are per-batch pass counts, not the price of a profile, the 32-sequence calibration budget multiplies them by the sixteen batches. Beyond two substitution decisions on the same model the profile is the cheaper route *at these budgets*, paying $2.01\times$ less time and $2.35\times$ less memory than the backpropagation baseline (Table D.3). Under that L-pass accounting, equalising the sequence count reverses that.

The same claim on a clock. Timed on one GPU at Pythia-410M in fp32, seq-len 1024: the profile ($2Lr$ probe forwards, r=2, once per checkpoint) takes 207.7 s, reading $\|\epsilon_\ell\|$ for one operation (L forwards, per decision) 52.2 s, direct measurement of a_ℓ (L full evaluations, per decision) 409.3 s. The smallest n with profile $+ n \cdot$ magnitude $\leq n \cdot$ direct is therefore $n = 1$ (259.9 s against 409.3 s), one decision earlier than the pass count predicts. Under the L-pass accounting, equalising the sequence count reverses that. The prediction registered before the run was $n = 2$, with $n \leq 2$ CONFIRMED, $n \in \{3, 4\}$ WEAKENED and $n \geq 5$ REFUTED.
Two secondary numbers from the same run: peak memory 2056 MB for the profile route against 2446 MB for direct measurement, a $1.19\times$ margin in the profile's favour, since it builds no backward graph while direct measurement runs the full model over the selection corpus. And the ratio of one full evaluation to one probe forward, the constant the priced scoreboard's horizontal axis is drawn in, 7.84 against the 8 the paper assumes. One caveat cuts both ways: the selection corpus yielded 246 of the nominal 256 sequences, making direct measurement about 4% cheaper than it should be. Correcting for it moves the evaluation constant to 8.2 and leaves $n = 1$ unchanged, so the break-even holds under a bias against us. This prices selection only, not the compressed model, on one model and one device.

At 7B, on the selection budget. Direct measurement on the 256-sequence selection budget costs at least 873.4 s per operation in bf16 on EXP111's card; the fp32 profile read on the 32-sequence calibration budget costs 705.1 s on EXP123's. At that budget the profile route would be ahead from the first decision, but the two figures come from different cards, so the matched comparison is the paragraph above.

D.11 THE FOUR ROUTES, PRICED

§4 summarises this in prose. The table is here (Table D.10).

Table D.10: **The four routes to "where should I compress?", and what each one cannot do.** Coverage is over the eleven operations of §3 and Appendix D.2; the weight-space row counts the operations that score has an object for *and* ranks, so four further operations displace no weights at all, and on three more the object exists, we ran it, and it does not rank them (Appendix E.11). "Needs the operation" asks whether the score can be computed before the operation is chosen. Cost is measured on Qwen2.5-7B for pricing all eleven (Appendix D.10). The profile is $2L$ forwards paid once plus one per operation, the oracle is L full evaluations per operation. †*These two are not a matched comparison and must not be read as one*: they are the budgets each route is actually run at (32 against 255 sequences), with the magnitude at L passes. Equalised, with the magnitude in one pass, the profile route is still ahead, 245 s against 1169 s over the eleven operations (Appendix D.10). Only the curvature profile is free of the operation. The magnitude and the R_3 gate are not, though the rate gate of Appendix I.13 is. Head-to-head where both are defined, HAWQ-V2's score is level with us on int4 at Pythia-410M ($\rho = 0.92$ against 0.93), a little behind on int3 (0.93 against 0.97), and behind on low-rank (0.48 against 0.69. Appendix E.10).

route	curvature object	coverage	needs the operation	cost
weight-space score (Dong et al., 2020; Yu et al., 2022)	$\mathrm{tr}(H_W)$	$4/11$	yes	—
free depth heuristic (Gromov et al., 2025)	none	$11/11$	no	0
direct measurement of a_ℓ (the oracle)	none	$11/11$	yes	9616 s†
CurvProbe	$\mathrm{tr}(H_\ell)/d$	$\mathbf{11/11}$	profile **no**, full score yes	**1377** s†

E RANKING DAMAGE ACROSS OPERATIONS

E.1 STATISTICS BEHIND THE CURVATURE-CORRECTION RESULT

Table E.2 gives the per-cell values for two of the five operations (the complete 11×5 grid is Table S6) and Table E.3 the profile controls, including the free priors. The $\exp(-3\ell/(L-1))$ row of the main text is the strongest of them.
The complete per-cell grid is in the supplementary appendix (Table S6).

Why not a per-cell test. The 55 cells are dependent, so we permute the before/after assignment jointly within each model: only the observed one of the exact 2^{11} assignments reaches 53 improved cells, and all eleven per-model gains are positive (sign-flip $p = 1/2048$). The cells are also imprecise one at a time: each is a rank correlation over one model's layers, so the count is the unit the claim is read at, and no single cell is offered as a verdict of its own. This is the sense in which the counts in Figure 4 and Table E.18 are aggregate.

Where the magnitude rule fails, and the forecast. Over the 89 core cells, one run per checkpoint, the rule is below chance on 31. A checkpoint term alone explains 57% of the variance in its ρ, an operation term 15% (35% against 26% on the eleven-model core of Table E.7). Four checkpoints carry most of the failure: LLaMA-13B (4 of 5 operations below chance, mean -0.45), Mistral-7B (5 of 5, -0.42), Llama-3-8B (5 of 5, -0.32) and GPT-J-6B (4 of 5, -0.26). The forecast replaces damage by the corrected score, $\rho(\|\epsilon_\ell\|^2, \frac{1}{2}\|\epsilon_\ell\|^2\kappa_\ell^{\mathrm{iso}})$, so it needs the profile and one pass per operation and no damage at all. It tracks the rule's measured ρ at 0.82 over the 89 cells and at 0.75 over the 39 cells of the six out-of-sample operations, and orders the eighteen checkpoints at 0.83. It flags 17 of the 128 cells, on five checkpoints and eight operations, and all 17 are below chance (median -0.48). It does not find every failure: 22 milder ones go unflagged (median -0.21), and of the 13 cells below -0.4 it flags 9. Within a checkpoint, deletion and linearisation rank 0.33 worse than the other three on 16 of 18 checkpoints (two-sided sign test $p = 0.0013$); on the eleven-model core the gap is 0.41, on ten of eleven ($p = 0.012$). The analysis was not registered. `analysis/verify_failure_forecast.py` recomputes every number in this paragraph from the released files.

The same test on the extended grid. 121 of the 128 cells improve and the exact sign-flip permutation puts 1 of the $2^{18} = 262{,}144$ assignments at or above the observed sum ($p = 3.8 \times 10^{-6}$), the smallest value the test can return. Three of the 128 cells are later measurements on GPT-J-6B, from the re-run on the corrected adapter of Appendix D.3, and all three improve; Bloom-7B1's five are the re-run on the fixed harness of Appendix F.7 (1 of 5 before it, 5 of 5 after).

The two cells where the correction does not improve. Both are on GPT-2-medium (Table E.2), the one grid model whose measured κ^{iso} is negative at any layer (layer 0, -6.5×10^{-6}). Dropping that layer does not repair either cell (Table S6), so the anomaly is not the explanation.

Is the Llama-3-8B reversal one layer's doing? Dropping the grid's most damaging layer moves Llama-3-8B's corrected correlation by at most 0.025 either way while the magnitude rule stays negative on all five, so the reversal belongs to the layer population, not to the single most damaging layer.

E.2 THE PROFILE TRANSFERS TO A SECOND PERTURBATION SITE

EXP 60 moves the hooked module one sub-layer later, to the MLP output, so every operation here is MLP-site (`quant4`, `quant3` and `lowrank` on the MLP projections, `prune` zeroing the sublayer, `headprune` a fraction of its neurons). Table E.6 gives the uncorrected magnitude and the MLP-site profile per cell, the cross-site rows in the supplementary appendix.
The complete per-cell grid is in the supplementary appendix (Table S2), with the released JSON/CSV.
A model's five operations share calibration data and one profile, so the 25 cells are dependent: under the exact 2^5 assignments only the observed one achieves 23 improved cells ($p = 0.031$), a block bootstrap gives 95% CI of $[21, 25]$ with all five per-model gains positive (sign-flip $p = 0.031$), and the cross-site profile improves 22 of 25 cells.
The profiles agree because their layer-wise *shape* does: the Spearman correlation between κ^{iso} at the MLP and attention sites is 0.98–1.00 on every checkpoint, so the cross-site row is the same shape at another scale rather than independent evidence for a third score, which is why one profile, read once, prices operations at both sites.

Table E.1: **Would a free depth prior do instead?** Every row scores $\frac{1}{2}\|\epsilon\|^2 \times$prior and the metric is mean Spearman with true damage over each model's five operations. The prior is $\exp(-c\,\ell/(L-1))$, its c chosen *without* seeing the held-out model: we report $c = 3$, which wins four checkpoints, and tuning it does not reach the profile — the two differ in the shape of the risk, not the mean. The calibration behind c is in the supplementary appendix.

	prior wins				profile wins							
held-out model	160M	410M	1.4B	GPT-J	GPT-2-m	Qwen-7B	Qwen-0.5B	TinyL	Llama-3-8B	Qwen3-8B	OLMoE-1B	mean
free prior, c=3	**.828**	**.577**	**.757**	**.620**	.173	.459	.477	.235	.244	.150	.182	.427
measured profile	.771	.512	.723	.538	**.417**	**.536**	**.800**	**.722**	**.746**	**.397**	**.507**	**.606**
gap	−.057	−.065	−.034	−.082	+.244	+.076	+.323	+.487	+.502	+.247	+.325	+.179

Table E.2: **Per-cell detail behind Table E.7, first block.** Spearman with true per-layer damage before and after multiplying the magnitude score by the probe-estimated curvature profile κ^{iso} (one profile per model, applied to all five operations, of which `linear` and `prune` are printed here and the other three in Table S6); it improves 53 of the 55 cells, the two exceptions (marked *) both on GPT-2-medium. Mean ρ $+0.152 \to +0.606$, median $+0.152 \to +0.616$, cells with $\rho \geq 0.5$ $11 \to 38$, cells with $\rho < 0$ $19 \to 1$.

substitution	score	160M	410M	1.4B	GPT-2-m	Qwen-0.5B	TinyL	GPT-J-6B	Qwen-7B	Llama-3-8B	Qwen3-8B	OLMoE-1B
`linear`	$\|\epsilon\|^2$	+0.455	−0.148	+0.141	+0.230	−0.024	−0.346	−0.819	−0.090	−0.333	+0.011	−0.582
	$+\,\kappa^{\text{iso}}$	**+0.902**	**−0.117**	**+0.666**	+0.179*	**+0.604**	**+0.623**	**+0.154**	**+0.335**	**+0.670**	**+0.422**	**+0.209**
`prune`	$\|\epsilon\|^2$	−0.091	−0.167	−0.037	+0.606	+0.200	+0.010	−0.537	+0.152	−0.386	+0.046	−0.332
	$+\,\kappa^{\text{iso}}$	**+0.462**	**+0.134**	**+0.428**	+0.442*	**+0.891**	**+0.752**	**+0.683**	**+0.548**	**+0.780**	**+0.512**	**+0.491**

Table E.3: Controls on the profile over the 11×5 grid; every row applies $\frac{1}{2}\|\epsilon\|^2 \times$ profile and Spearman is scale-invariant, so only the profile's shape across layers matters. Table E.1 gives the leave-one-model-out comparison.

per-layer profile	cells improved	mean ρ
κ^{iso}, measured by random probes (ours)	$\mathbf{53}/55$	$\mathbf{0.606}$
$\exp(-3\ell/(L-1))$, free	$44/55$	0.427
$1/(\ell+1)$, free	$44/55$	0.405
$1/(\ell+1)^2$, free	$36/55$	0.427
constant (= no correction)	$0/55$	0.152
κ^{iso} shuffled across layers	$24.1/55$ $[18, 30]$	0.144
κ^{iso} reversed	$7/55$	0.068

The full extended grid, the free prior's calibration, the null distributions of both permutation tests, the weight-noise and layer-permuted controls, the per-layer and per-operation values behind the caveats above, the shuffle and reversal readings, the cell-by-cell correction figure and the cross-site MLP rows are in the supplementary appendix.

E.3 The eleven-operation grid

Figure 3 plots the per-operation means. Table E.7 prints them, with the model counts and the weight-space column that the figure does not carry. The per-cell values behind its first block are Table E.2, and the six registered out-of-sample operations are defined in Appendix D.2.

E.4 The looser, source-row split

§5 reports the article-held-out test, in which the curvature profile, the perturbation magnitudes and Block Influence are all measured on calibration articles. Table E.8 gives the looser variant: the magnitudes and the damage are measured on the evaluation segment and only the curvature profile is transferred.

On four checkpoints the profile's damage-ranking correlation is $\rho = 0.42$ to 0.96 (the widest span across the three calibration profiles is 0.16), while the magnitude rule scores -0.41 on Llama-3-8B and at most $+0.20$ on the other three. The gap is largest exactly where §5 predicts it, turning -0.41 into $+0.72$ to $+0.85$ there. It does not uniformly outperform the alternatives. Block Influence is higher on Llama-3-8B and Qwen2.5-7B, and deepest-k is close behind on Llama-3-8B.

Table E.4: **Each factor alone against the product.** The 89 core cells, one run per grid checkpoint of §3. A cell is *negative* when that score's Spearman with true per-layer damage falls below zero, so a score that ranks layers worse than chance is counted there. Neither factor carries the ranking alone: displacement is negative in 31 cells and the curvature in 31, against 1 for the product, which beats whichever single factor is better on 70 of the 89 and on 16 of the 18 checkpoints (sign test over checkpoints, $p = 0.001$). Pooling all 28 released runs, replicas included, gives 57, 41 and 4 of 140.

score	cells negative	median ρ
displacement alone, $\|\epsilon_\ell\|^2$	$31/89$	$+0.169$
curvature alone, $\kappa_\ell^{\mathrm{iso}}$	$31/89$	$+0.203$
their product (ours)	$\mathbf{1}/89$	$\mathbf{+0.666}$

Table E.5: **Every cell the released files carry**: the extended grid behind the 121 of 128 quoted in the main text, eleven operations over eighteen checkpoints, up to five core operations each. *cells* counts model–operation cells with a released per-layer measurement, *improved* those where the corrected score's rank correlation with true damage exceeds the magnitude rule's, and the last column is the mean gain in ρ over that checkpoint's cells; Bloom-7B1 is the re-run on the fixed harness of Appendix F.7. The 128 cells are not independent (models share corpora and architectures), so significance is claimed over checkpoints, not cells.

checkpoint	cells	improved	mean $\Delta\rho$
Pythia-160M	11	11	+0.401
Pythia-410M	11	10	+0.084
Pythia-1.4B	8	7	+0.226
GPT-2-medium	11	8	+0.159
Qwen2.5-0.5B	11	11	+0.589
TinyLlama-1.1B	11	10	+0.528
GPT-J-6B	8	8	+0.991
Qwen2.5-7B	8	8	+0.261
Llama-3-8B	5	5	+1.069
Qwen3-8B	5	5	+0.297
OLMoE-1B-7B	5	5	+0.436
LLaMA-13B	5	5	+1.031
Gemma-2-9B	5	5	+0.607
Mistral-7B-v0.3	5	5	+1.026
OPT-6.7B	4	3	+0.026
OLMo-1-1B	5	5	+0.175
OLMo-1-7B	5	5	+0.102
Bloom-7B1	5	5	+0.141
all eighteen	**128**	**121**	

Table E.6: **Per-cell detail behind the site-transfer result.** Spearman with true per-layer damage for five MLP-site operations on five checkpoints under three scores: uncorrected magnitude; times a profile measured at the MLP site (primary); times the attention-site profile unchanged (cross-site). Bold improves over the uncorrected row, * does not; mean ρ over the 25 cells is $0.402 \to 0.646$ (MLP site) and $0.402 \to 0.655$ (attention site). Each cell reads uncorrected $\to$ MLP-site corrected; the cross-site rows are in the supplementary appendix.

substitution	160M	410M	TinyL	GPT-J-6B	Qwen-7B
`quant4`	$+0.615 \to \mathbf{+0.720}$	$+0.717 \to \mathbf{+0.900}$	$+0.708 \to \mathbf{+0.857}$	$-0.019 \to \mathbf{+0.488}$	$+0.689 \to +0.670^*$
`quant3`	$+0.797 \to \mathbf{+0.832}$	$+0.857 \to +0.836^*$	$+0.909 \to \mathbf{+0.973}$	$-0.493 \to \mathbf{+0.297}$	$+0.635 \to \mathbf{+0.902}$
`lowrank`	$-0.049 \to \mathbf{+0.231}$	$+0.603 \to \mathbf{+0.617}$	$+0.783 \to \mathbf{+0.921}$	$-0.391 \to \mathbf{+0.570}$	$+0.790 \to \mathbf{+0.875}$
`prune`	$+0.021 \to \mathbf{+0.196}$	$+0.537 \to \mathbf{+0.634}$	$+0.320 \to \mathbf{+0.432}$	$-0.026 \to \mathbf{+0.350}$	$+0.599 \to \mathbf{+0.697}$
`headprune`	$-0.217 \to \mathbf{+0.042}$	$+0.570 \to \mathbf{+0.602}$	$+0.652 \to \mathbf{+0.756}$	$-0.349 \to \mathbf{+0.923}$	$+0.799 \to \mathbf{+0.829}$

E.5 The frozen protocol, both operations

§5 states the article-held-out deletion result, every comparator in Table E.9. §5 reports that the same frozen profile serves the second, low-rank operation, Table E.10 the held-out ranking, with the article split, the profile and the magnitude/Block-Influence stages unchanged from the deletion run and the test articles supplying damage only, Table E.11 the joint evaluations at $k{=}L/4$ (8 of 32 layers on Llama-3-8B, 10 of 42 on Gemma-2-9B). Low-rank was registered as optional, not part of predictions P1–P6.

Two kappa variants: *ref* takes the probe amplitude from the norm of the reference operation's linearisation error (§4), *clean* uses $c\|y_\ell\|$ from the clean network alone. They order the layers almost

Table E.7: **One profile per model, applied to eleven compression operations it was never shown.** Spearman correlation with true per-layer damage before and after multiplying the magnitude rule by κ^{iso}, meaned over models. The two operations on which the magnitude rule is *negatively* correlated with damage are the two that replace the sublayer's function outright, and the two that move furthest. The last column asks whether a second-order *weight-space* score ranks the layers. It fails for two different reasons, four operations displace no weights at all, and on three that do, we ran it and it does not rank them (EXP106), split out in Appendix E.10. Second block registered out-of-sample. Per-cell values in Appendix E.1.

operation	models	$\rho(\|\epsilon\|^2, a)$	$\rho(\frac{1}{2}\|\epsilon\|^2\kappa^{\mathrm{iso}}, a)$	weight-space score
`linear` *(softmax → ReLU kernel)*	11	-0.137	$+\mathbf{0.423}$	does not apply
`prune` *(sublayer removed)*	11	-0.049	$+\mathbf{0.557}$	applies ($\Delta W{=}-W_o$); fails
`quant4`	11	$+0.320$	$+\mathbf{0.662}$	defined
`quant3`	11	$+0.421$	$+\mathbf{0.734}$	defined
`lowrank` *(25% rank)*	11	$+0.203$	$+\mathbf{0.656}$	defined
six registered out-of-sample operations	5–8	$+0.294$	$+\mathbf{0.683}$	applies to 1 of 6
all 94 cells of this eleven-operation grid		**88** improved		applies to 4 of 11

Table E.8: **Transfer of the curvature component across source-row text segments.** Curvature profiles fit on three non-overlapping source-row segments of WikiText-103 training text (byte-pinned by sha256) score per-layer damage on a fourth evaluation segment. Perturbation magnitudes and damage are measured on that segment. The transferred component is the curvature profile. Profile ρ is per-calibration-segment, meaned over the three. All selector scores use the convention that smaller values indicate lower predicted damage, so deepest-k uses negative layer index. Block Influence is read from the existing substitutions files and has a different calibration source.

checkpoint	profile ρ (3 calsets)	magnitude ρ	deepest-k ρ	BI ρ
Qwen2.5-0.5B	0.95 (0.94–0.96)	$+0.20$	-0.00	$+0.51$
Llama-3-8B	0.78 (0.72–0.85)	-0.41	$+0.64$	$+0.89$
Qwen2.5-7B	0.50 (0.42–0.58)	$+0.20$	-0.10	$+0.56$
Gemma-2-9B	0.49 (0.46–0.55)	$+0.02$	-0.03	$+0.09$

Table E.9: **Article-held-out, fully-frozen confirmation.** Curvature profile, perturbation magnitude and Block Influence are all measured on calibration articles, damage on held-out articles; profile ρ is the range over the three calibration articles. *Permuted* shuffles the measured profile across layers with the magnitudes held fixed, so it isolates the ordering. Smaller selector values mean lower predicted damage, so deepest-k uses negative layer index.

checkpoint	profile ρ	magnitude ρ	permuted	deepest-k	BI	direct
Llama-3-8B	0.63–0.86	-0.37	-0.24	$+0.63$	$+0.87$	0.96–0.99
Gemma-2-9B	0.49–0.55	-0.03	$+0.03$	-0.05	$+0.15$	0.97–0.98

identically on both models ($\rho = 0.98$ on the calibration segments, Appendix D.7), moving the numbers but not the reading.

Table E.10: **Second operation, article-held-out, fully frozen.** Low-rank approximation at 25% rank on the attention projections. Strict is the frozen score $\frac{1}{2}\|\epsilon_\ell\|^2\kappa_\ell^{\mathrm{iso}}$, averaged over calibration sequences with both factors measured on calibration articles; the test articles only measure damage. Permuting the profile across layers drops the correlation to -0.25 (Llama-3-8B) and -0.25 (Gemma-2-9B).

model	κ	cal_0	cal_1	cal_2	magnitude	deepest-k	direct
Llama-3-8B	ref	$+0.534$	$+0.860$	$+0.786$	-0.33	$+0.60$	$+0.90$–$+0.96$
	clean	$+0.387$	$+0.853$	$+0.752$	-0.33	$+0.60$	$+0.90$–$+0.96$
Gemma-2-9B	ref	$+0.882$	$+0.896$	$+0.882$	-0.05–-0.08	-0.74	$+0.96$–$+0.97$
	clean	$+0.901$	$+0.897$	$+0.886$	-0.05–-0.08	-0.74	$+0.96$–$+0.97$

The complete per-cell grid is in the supplementary appendix (Table S7), with the released JSON/CSV. Downstream tasks, low-rank arm on Llama-3-8B, paired over the same items: CurvProbe against direct is $+2.4$ points on LAMBADA (95% interval $[-0.6, +5.4]$), $+2.1$ on ARC-Easy ($[+0.2, +4.0]$), $+1.2$ on HellaSwag ($[-0.4, +2.8]$) and -0.8 on MMLU ($[-2.6, +1.0]$), three intervals cross zero, the fourth lying just above it in the profile's favour, so the task-level cost of not measuring is zero at this resolution.

Table E.11: **Joint evaluation at $k{=}L/4$, mean token NLL:** one-shot application of the named layer set to the held-out stream, lower better. The printed rows are calibration segment cal_0 (κ = ref), clean variant in parentheses where it differs; "direct" selects the lowest directly-measured calibration damage. The other two segments and the all-layers row are in the supplementary appendix (Table S7). Llama-3-8B drops 8 of 32 layers, Gemma-2-9B 10 of 42, the latter evaluated with a document-start <bos>.

arm	Llama-3-8B		Gemma-2-9B	
	deletion	low-rank	deletion	low-rank
profile, cal_0	2.212 (2.315)	2.143 (2.203)	3.366 (3.514)	2.301 (2.296)
direct, cal_0	2.185	2.112	2.704	2.290
magnitude rule	7.395	4.862	2.919	2.503
deepest-k	2.979	2.342	3.608	2.656

Table E.12: **Downstream tasks, Llama-3-8B low-rank at $k{=}L/4$.** Accuracy (%) on 500 items per task (479 for ARC-Easy) and mean token NLL on the same stream; CurvProbe's set is the published one, direct is cal_2's lowest-damage layers, Block Influence the eight least-influential sublayers, all applied to the same 8 of 32 sublayers.

	CurvProbe	direct	deepest-k	Block Influence	magnitude	all layers
LAMBADA	65.8	63.4	57.2	55.8	6.4	0.0
ARC-Easy	67.9	65.8	64.9	65.1	38.4	29.9
HellaSwag	60.0	58.8	57.8	58.4	32.6	27.2
MMLU	40.2	41.0	40.4	41.2	28.2	27.2
mean token NLL	2.125	2.115	2.342	2.145	4.862	7.732

The profile beats the cheap baselines in three of the four cells and direct measurement beats it in all four, by at most 0.03 nats in three and by 0.66 on Gemma-2-9B deletion, where the profile is second-worst: its lowest-scoring layers there are 3–14, while Gemma-2-9B's safest are its deepest (smallest $\|y_\ell\|^2$), which the budget magnitude rule happens to name and direct measurement finds, so a positive, moderate global correlation ($\rho = 0.49$–0.55) does not translate into a better top-k selection. The profile wins the low-rank cell decisively on the same model and both cells on Llama-3-8B: a boundary, not a general failure.

Block Influence, the strongest published selector, picks the eight least-influential sublayers (bottom-8 of ShortGPT's per-layer score, layers 22–29) and lands on deepest-k on every task (LAMBADA 55.8 against 57.2, paired 34/27, $p = 0.44$); against the profile it loses LAMBADA (68/18 of the 500 items, exact $p = 5 \times 10^{-8}$) and ARC-Easy ($p = 0.029$), and separates from it on neither of the other two.

E.6 Does the calibration corpus move either selector?

Table 1 has Block Influence ahead of the isotropic profile on GSM8K. Since Block Influence too is read from calibration data, the objection is that it was calibrated on better text. Re-reading both selectors on mathematical text removes it.

Only the corpus changes, same checkpoint, estimator, probe count, seed and batch size: GSM8K's own problem text from the *train* split, 1921 problems with worked solutions, 1.0M characters, 1901 with explicit arithmetic, so it cannot contain an evaluated item. At $2L$ forward passes for the profile and one per batch for Block Influence, the test is minutes on one card, not a training run.

The reference point is not $\rho = 1$: two WikiText calibration segments already disagree with each other, and a corpus landing inside that spread would tell us nothing, so the internal spread is measured first and every number below is read against it.

Table E.13: **The calibration corpus does not move either selector.** Scores read on GSM8K's mathematical text against the same scores read on WikiText, for Llama-3-8B, $k{=}L/4$. The last column is the spread between the three WikiText calibration segments themselves, the reference point.

selector	ρ vs WikiText	top-8 overlap	WikiText internal spread
profile κ^{iso}, ordering	$+0.88$–$+0.98$	8, 8, 8 of 8	$+0.91$–$+0.99$
profile, selection made	—	5, 7, 7 of 8	—
Block Influence, ordering	$+0.993$–$+0.994$	7, 8, 8 of 8	$+0.996$–$+0.999$

Neither selector moves: the profile returns the same eight lowest-curvature layers (the ones it ranks safest) against all three WikiText segments. Two caveats: on this checkpoint those eight are layers 24–31, exactly the deepest-k set, so the agreement is on a degenerate tail and weaker evidence than it looks. And the eight *highest*-curvature layers agree in 7, 8 and 8 of 8, not all three. Block Influence is steadier: agreement across WikiText segments is 0.996 to 0.999, so a corpus effect would have to be smaller than that to be invisible, and it is.

The ordering rows are about κ^{iso} alone. The selection the pipeline makes weights that ordering by $\frac{1}{2}\|\epsilon_\ell\|^2$, reshuffling near-equal layers. The sets differ only inside one deep band ($[11, 18\text{–}20, 22\text{–}25]$ against $[18\text{–}20, 22\text{–}26]$ on the segment that differs most): tie-breaking among near-equal layers, not a corpus-driven reordering, and calling the profile targetable would require the ordering to move.

The boundary stands, now measured rather than assumed: Block Influence's advantage on GSM8K is not better calibration text but a difference in what the two quantities measure. $\kappa_\ell^{\mathrm{iso}} = \mathrm{tr}(H_\ell)/d$ is a property of the checkpoint, and no corpus turns it into a property of the task.

Reproduce with `make_math_corpus.py` (builds the math corpus from the GSM8K train split), `p1a_strict_measure.py --stage {kiso,bi} --seg math` (reads either selector) and `p1a_corpus_pretest.py` (compares them). Artifacts in `iclr2027/results/p1a_corpus_pretest/`.

E.7 The layer-scale competitor

Dividing the squared perturbation magnitude by each layer's own residual-stream or attention-output RMS^2 is the cheapest rival correction and the one a reader is most likely to reach for without measuring anything, since it uses quantities the forward pass already produces. It is read on the eight checkpoints that carry both the substitution grid and the stream norms, $8 \times 5 = 40$ cells. A checkpoint missing either file is dropped from the whole table rather than from one row, so every score is read on exactly the cells the profile is.

The curvature runs opposite to the norm, and that comparison is largely depth. The table asks whether a layer's own scale can stand in for its curvature inside a *score*. The two quantities can also be compared directly, and they move in opposite directions: on these eight checkpoints $\kappa_\ell^{\mathrm{iso}}$ is negatively rank-correlated with the residual-stream RMS on seven, at a median $\rho = -0.82$, and the layer carrying the largest curvature sits at or beside the smallest norm on six. A confound is that both quantities are close to monotone in depth, so most of that number is the depth trend, not a relation between them: comparing adjacent layers, where the trend is removed, the median drops to $\rho = -0.01$ and the sign survives on two of the eight. The exception at the raw level is GPT-2-medium at $\rho = -0.005$, the one checkpoint whose profile also fails on more than one operation (§5). What is *not* depth is the score comparison of Table E.15: dividing by a layer's own scale still trails the curvature at 0.396 and 0.266 against 0.627 over forty cells, so a score built from stream magnitude is reaching for a different quantity, and the reading that survives the control is the one the proposal rests on. `analysis/verify_curvature_vs_norm.py` re-derives every entry, and `analysis/verify_depthconfound.py` the control.

Table E.14: **Curvature against residual-stream norm**, the eight checkpoints carrying both. Rank correlation of $\kappa_\ell^{\mathrm{iso}}$ with the block-input RMS, and where layer 0 falls in each ordering (1 is largest).

checkpoint	L	$\rho(\kappa^{\mathrm{iso}},\ h_{\mathrm{rms}})$	layer 0 by κ^{iso}	layer 0 by h_{rms}
Pythia-160M	12	-0.993	1	12
Qwen2.5-7B	28	-0.962	1	27
TinyLlama-1.1B	22	-0.839	2	22
Pythia-1.4B	24	-0.830	1	24
GPT-J-6B	28	-0.818	1	28
Qwen2.5-0.5B	24	-0.793	1	24
Pythia-410M	24	-0.359	1	24
GPT-2-medium	24	-0.005	24	24
median		-0.824		

Neither rescaling closes the gap: the better of the two still loses 31 of the 40 cells to the measured profile, which is the result the correction needs, what the profile contributes is not a per-layer scale a cheaper statistic could have supplied. The table comes from `24_profile_ablation.py --cohort scale` (the same script and cohort rule that produce the eleven-model controls of Table E.3 at `--cohort all`), and `analysis/verify_scalecontrol.py` re-derives all six figures from `substitutions_*.json` and `stream_norms_*.json`, exiting non-zero on any

Table E.15: **The layer-scale competitor, 40 cells.** Mean Spearman against true per-layer damage, and the number of cells on which the curvature profile ranks better than the row. Rescaling by a layer's own norm recovers part of the gain and not the rest.

score	mean ρ	profile ranks better
$\|\epsilon_\ell\|^2$ (uncorrected)	0.228	—
$\|\epsilon_\ell\|^2/\|h_\ell\|^2$ (residual stream)	0.396	31 of 40
$\|\epsilon_\ell\|^2/\|y_\ell\|^2$ (attention output)	0.266	35 of 40
$\frac{1}{2}\|\epsilon_\ell\|^2\kappa^{\mathrm{iso}}$ (ours)	**0.627**	—

mismatch. The last column is a head-to-head count per cell. The "cells improved" column the script prints answers a different question and reads 28 and 22.

E.8 Depth is a proxy, not a mechanism

Depth-based conversion heuristics are usually justified by a Lipschitz cascade: with $\delta_{t+1} = \delta_t + \Delta f_t$ one has $\|\delta_L\| \le \|\delta_{\ell+1}\| \prod_{t>\ell}(1 + M_t)$ where $M_t = \|\Delta f_t\|/\|\delta_t\|$, which is exponential in $L - \ell$. Every M_t is directly measurable, since $\Delta f_t = (\tilde{h}_{t+1} - \tilde{h}_t) - (h_{t+1} - h_t)$. Table E.16 puts the predicted amplification against the realised one. The bound is a product along the path the perturbation actually takes, not a static constant per layer, which is why it is not monotone in the depth of injection: injecting at ℓ=0 multiplies more factors than injecting at ℓ=5, and the measured M_t differ between the two paths.

Table E.16: Predicted versus realised amplification of a single-layer linearisation, Pythia-410M, 64 sequences of 1024 tokens. "Realised" is $\|\delta_L\|/\|\delta_{\ell+1}\|$. The bound is not merely loose but loose by a factor that itself grows with depth of injection. The median ratio over all 24 layers is 148 and realised amplification never exceeds 7.8×.

converted layer ℓ	bound $\prod_{t>\ell}(1 + M_t)$	realised	ratio	$\cos(\delta_{\ell+1}, \delta_L)$
0	5,608	7.79	720	+0.005
5	34,823	7.64	4,555	+0.056
12	851	4.80	177	+0.151
18	30.9	2.48	12	+0.318
21	4.3	1.73	2	+0.413
23	1.0	1.00	1	+1.000

The last column explains the gap. Downstream reactions dominate the final perturbation in magnitude (the injected error accounts for only 12.8% of $\|\delta_L\|$ when layer 0 is converted), but they are nearly orthogonal to the injection and to each other, so they accumulate as a random walk rather than coherently. The bound compounds a worst-case alignment that is never realised.

The same measurement gives the sharpest one-line statement of the paper's argument. Across the 24 layers of Pythia-410M, the relative perturbation *at the point of injection* correlates with damage at $\rho = -0.050$, while the same quantity measured *at the network output* correlates at $\rho = +0.987$. Every local salience score measures the first.

E.9 Per-field salience baselines

Each compression literature has its own published salience score, and a fair reading of §5 must ask whether those beat the generic magnitude score on their own operation. They do not: Table E.19 correlates each score with per-layer damage for the operation it was designed for, and Table E.20 asks which *subset* the score chooses and at what cost, against exhaustively measured damage, so regret is known rather than estimated. One score is also transplanted in granularity rather than only in operation: AWQ's salience is defined per input channel, and the reduction to one number per layer, the unit every comparison here is made at, is ours.

The field-wide view (Figure 4). Table E.18 is the grid behind the main text's figure: every published score we can recompute, on the operation it was designed for, one run per grid checkpoint of §3. Each cell is one checkpoint under one operation, and the column that matters is the last one, how often a score ranks layers *worse than chance* on the operation it was written for. The displacement magnitude does so on 31 of its 89 cells and AWQ's own salience on 8 of 34. The corrected score, on the same 89, does so on 1, and on none of AWQ's 34. On Block Influence's 17 `prune` cells the corrected score's median is +0.543 against +0.483, and it is negative on none of them against Block Influence's 3.

The substitution files hold 28 runs, not one per checkpoint: eight models appear a second time calibrated on C4 rather than WikiText, GPT-J-6B appears four times across its adapter and unification variants (Appendices I.8, D.3), and OPT-6.7B carries no Block Influence or AWQ column, so it enters the magnitude row only. Counting every run would count one checkpoint up to four times, and significance here is claimed over checkpoints, so the figure and the main text take one run each, the WikiText run; the two GPT-J adapter runs give the same counts. Pooling all 28 runs reads the same way (right half of the table), with one difference worth stating: the replicas lift Block Influence's `prune` median to $+0.536$, level with the corrected score's $+0.528$. We recompute both halves read-only from `results/substitutions_*.json`, which already store `block_influence`, `awq_salience`, `kappa_iso` and each operation's `eps_norm` and `d_nll`. We ran no measurement for the figure. `results/analysis/verify_fieldscores.py` reproduces every number in this section and in the figure, and fails loudly on any that drifts.

The spread across checkpoints. The counts above are pooled, which leaves open whether a few checkpoints carry them. Table E.17 gives the corrected score's mean rank correlation on each of the seventeen that carry all five core operations separately. The floor is Qwen3-8B at $+0.397$ and the ceiling OLMo-7B at $+0.889$; every checkpoint is positive on the mean, and the one below-chance cell of the eighty-five, Pythia-410M `lowrank` at -0.117, sits on a checkpoint whose mean is $+0.512$. We looked for a property of the profile that would predict where it does better and did not find one: the dynamic range of κ^{iso} across layers spans $3.3\times$ (OLMo-1B) to $5470\times$ (Qwen3-8B) and is uncorrelated with the mean above ($\rho = +0.18$, $p = 0.49$), as is the ratio of layer 0 to the within-network median. The shape is nonetheless consistent: κ^{iso} falls with depth on sixteen of the seventeen, several at $\rho = -1.000$ exactly, Gemma-2-9B the sole inversion at $+0.857$. `results/analysis/verify_profilespread.py` reproduces the floor, the ceiling and the placement of that cell.

Table E.17: The corrected score per checkpoint, one run each, over the five core operations. Ordered by the mean. "worst" and "best" are over that checkpoint's five operations.

checkpoint	mean ρ	worst	best
qwen3_8b	$+0.397$	$+0.150$	$+0.599$
gpt2m	$+0.417$	$+0.179$	$+0.592$
olmoe	$+0.507$	$+0.209$	$+0.741$
olmo1b	$+0.511$	$+0.015$	$+0.724$
410m	$+0.512$	-0.117	$+0.971$
qwen7b	$+0.536$	$+0.335$	$+0.691$
gptj6b	$+0.538$	$+0.154$	$+0.859$
bloom7b	$+0.542$	$+0.350$	$+0.714$
llama13b	$+0.577$	$+0.244$	$+0.850$
mistral7b	$+0.606$	$+0.438$	$+0.836$
gemma29b	$+0.706$	$+0.443$	$+0.914$
tinyllama	$+0.722$	$+0.607$	$+0.816$
1.4b	$+0.723$	$+0.428$	$+0.970$
llama38b	$+0.746$	$+0.670$	$+0.780$
160m	$+0.771$	$+0.462$	$+0.923$
qwen05	$+0.800$	$+0.604$	$+0.891$
olmo7b	$+0.889$	$+0.846$	$+0.952$

Table E.19 is the per-cell view on the six models where every score is defined, kept because it shows *where* each score fails rather than how often.

The complete per-cell grid is in the supplementary appendix (Table S9), with the released JSON/CSV.

E.10 The weight-space baseline in full

`linear`, `swa`, `kvquant` and `actquant` touch no weight, so there is no Δw to score. `prune`, `headprune` and `sparse24` do displace weights and their weight-space Hessian is well defined. What fails is the quadratic model, the displacement being the whole parameter block, far outside the regime OBD/OBS-style scores are derived in, hence the layer-removal literature's distance criteria. EXP106 (Appendix E.11) measures it: ΔW from the operation itself gives mean $\rho = 0.005$, losing to κ^{iso} on all 15 cells. Seven of eleven thus escape a weight-space second-order score: four with no object to score, three with one that does not rank them.

Table E.18: Every published salience score we can recompute, on the operation it was designed for. "cells" is one run under one operation; "< 0" counts the cells where the score orders layers worse than chance. Left: one run per grid checkpoint (18), the figure's population. Right: all 28 runs, the C4 replicas and GPT-J variants included and OPT-6.7B absent. The corrected score is also shown restricted to each published score's own cells, so the comparison is like for like.

		one run per ckpt. (18)			all 28 runs		
score	its own operations	cells	median	< 0	cells	median	< 0
$\|\epsilon_\ell\|^2$ displacement	all five	89	$+0.169$	31	140	$+0.086$	57
AWQ salience (Lin et al., 2024)	`quant4`, `quant3`	34	$+0.340$	8	56	$+0.290$	13
Block Influence (Men et al., 2025)	`prune`	17	$+0.483$	3	28	$+0.536$	5
$\frac{1}{2}\|\epsilon_\ell\|^2\kappa^{\text{iso}}$ (**ours**)	all five	89	$+\mathbf{0.666}$	**1**	140	$+\mathbf{0.611}$	**4**
restricted to `quant4`, `quant3`	vs AWQ	34	$+0.727$	0	56	$+0.733$	0
restricted to `prune`	vs Block Infl.	17	$+0.543$	0	28	$+0.528$	1

Table E.19: Each literature's own salience score on its own substitution, across the six models (Spearman with true per-layer damage). No score is dependable across the grid: AWQ-style is strong on the three Pythia models and weak elsewhere, Block Influence changes sign between model families, and the substitution-distance score is negative in 7 of 30 cells.

score	substitution	160M	410M	1.4B	GPT-2-m	Qwen	TinyL
AWQ-style activation-weighted	`quant4`	$+0.720$	$+\mathbf{0.904}$	$+0.850$	$+0.183$	$+0.147$	$+0.290$
AWQ-style activation-weighted	`quant3`	$+0.182$	$+\mathbf{0.885}$	$+0.873$	$+0.466$	$+0.243$	$+0.497$
Block Influence (Men et al., 2025)	`prune`	-0.280	-0.160	$+0.352$	$+0.431$	$+0.480$	$+0.483$
attention-KL (sm‖lin)	`linear`	-0.671	-0.606	—	—	—	—
$\|\epsilon_\ell\|^2$	`linear`	$+0.455$	-0.148	$+0.141$	$+0.230$	-0.024	-0.346
$\|\epsilon_\ell\|^2$	`prune`	-0.091	-0.167	-0.037	$+0.606$	$+0.200$	$+0.010$
$\|\epsilon_\ell\|^2$	`quant4`	$+0.846$	$+0.935$	$+0.928$	$+0.396$	$+0.241$	$+0.103$
$\|\epsilon_\ell\|^2$	`quant3`	$+0.538$	$+0.957$	$+0.953$	-0.030	$+0.273$	$+0.478$
$\|\epsilon_\ell\|^2$	`lowrank`	$+0.287$	$+0.456$	$+0.188$	$+0.315$	$+0.169$	$+0.132$

Table E.20: Selection against exhaustively measured damage: linearisation at $k = 3$, all $\binom{12}{3} = 220$ triples at Pythia-160M, 302 sampled subsets at Pythia-410M. "Regret" is the ΔNLL of the chosen subset above the best measured (nats), "rank" its position among all scored. At 410M directly measured per-layer damage is exactly optimal; the two attention-distribution scores and the first-order score, all worse than a random subset there, are printed in Table S9 of the supplementary appendix. The failure is in what the scores measure, not in the per-layer form of the rule (Proposition 7). CurvProbe, scored on the exhaustive 160M table, lands there on deepest-k's triple. Two triples this table talks about are absent from the 302 — the displacement's alternative $(0, 2, 4)$, and CurvProbe's $(2, 4, 9)$ — so both were measured on the oracle's own protocol (`results/measure_subset.py`, `--layers 0 2 4`, `--layers 2 4 9`); the clean baseline reproduces the oracle's 3.018946 to six decimals. They land at 0.698 (rank 173 of 302) and 0.479 (rank 121). Both beat the random draw and neither is optimal: at this budget the corrected score selects mid-pack on the one operation whose k=3 landscape is sampled rather than enumerated, the ranking-versus-selection gap §7 measures. ‡The 410M $\|\epsilon_\ell\|$ row of Table S9 uses the original subset evaluation's per-layer displacement (`score_410m.json` via `04_score.py`, predictor `v1 d_ell`).

	Pythia-160M (220)		Pythia-410M (302)	
selector	regret	rank	regret	rank
measured a_ℓ, per-layer	0.000	1	0.000	1
$\frac{1}{2}\|\epsilon_\ell\|^2\kappa^{\text{iso}}$ (**CurvProbe**)	0.092	2	0.479	121
random k-subset	1.080	106	0.599	147

Table E.21 is the re-run with HAWQ-V2's own $\|\Delta W_\ell\|^2$ displacement, from the same converged runs (100 and 200 probes). How the trace is built, and the earlier mis-specified runs, are in the supplementary appendix.

Correcting it moves the baseline both ways: slightly down on quantisation ($+0.934 \to +0.923$, $+0.961 \to +0.928$ at 410M. $+0.868 \to +0.851$, $+0.899 \to +0.835$ at 1.4B), up and by more on low-rank ($+0.413 \to +0.483$, $+0.480 \to +0.596$), strengthening the baseline where we had claimed

the widest margin. Unchanged reading: competitive where designed to be used, below κ^{iso} on all six cells (int4 at 410M narrowly, 0.92 against 0.93. 1.4B by more, 0.84 against 0.95 on int3).

Convergence. Layers drifting beyond 0.25 from every correlation below are *excluded* and the κ^{iso} column recomputed on the same subset, so the two stay paired: Hutchinson's estimator is noisy, the probe count deciding whether its output is a property of the Hessian or of the budget. The runs use 100 probes (Pythia-410M, $20/24$ converged) and 200 (Pythia-1.4B, $18/24$). At 50 too many layers dropped out, an under-probed baseline being worthless as a comparison.

Table E.21: Weight-space (HAWQ-V2, $\overline{\text{tr}}(H_W)\|\Delta W\|^2$) against layer-output (ours): Spearman with true per-layer damage over converged layers (20 of 24 Pythia-410M, 18 of 24 Pythia-1.4B), not comparable to Table E.2 (all layers); the `prune` and `linear` rows are in Table S10.

	Pythia-410M		Pythia-1.4B	
operation	$\overline{\text{tr}}(H_W)\|\Delta W\|^2$	κ^{iso} (ours)	$\overline{\text{tr}}(H_W)\|\Delta W\|^2$	κ^{iso} (ours)
`quant4`	$+0.923$	$+0.931$	$+0.851$	$+0.915$
`quant3`	$+0.928$	$+0.971$	$+0.835$	$+0.946$
`lowrank`	$+0.483$	$+0.693$	$+0.596$	$+0.643$

Cost. Weight space needs a backward graph and pays for it, hence the limit on its usability in serving stacks exposing only forward inference (timings in the supplementary appendix).

E.11 EXP106: RUNNING THE WEIGHT-SPACE SCORE WHERE WE SAID IT COULD NOT GO

What forced this. Table E.7 says a second-order *weight-space* score does not apply to seven of eleven operations. Three of those seven (`prune`, `headprune`, `sparse24`) are genuine weight displacements, and a reviewer was right that "not run" does not support "cannot be applied". This runs them.

An error it corrects on the way. EXP36 defined the baseline as $\overline{\text{tr}}(H_W)\|Q(W) - W\|^2$ but multiplied $\overline{\text{tr}}(H_W)$ by $\|\epsilon_\ell\|$, the *activation*-space perturbation, never forming a weight displacement, which is not HAWQ-V2's score. Building this experiment surfaced it, and we have since re-run Appendix E.10 with the correct displacement. Here we snapshot the weights, apply the paper's own operation, and difference; the run aborts if the weights do not restore. `prune`, a module swap, uses the equivalence $\Delta W = -W_o$ the paper states elsewhere, labelled an equivalence rather than a measurement. We report the as-published variant beside the corrected one.

	$\|\epsilon\|^2$	$\overline{\text{tr}}(H_W)\|\Delta W\|^2$	$\overline{\text{tr}}(H_W)\|\epsilon\|^2$	κ^{iso} (ours)
mean ρ over 15 cells	0.225	$\mathbf{0.005}$	0.141	$\mathbf{0.639}$

What it returned. Five checkpoints (the ones carrying all three operations), three operations each, on the layers where the Hutchinson trace converged: the corrected score averages $\rho = 0.005$ with true per-layer damage, absent, not weak, and negative on six cells (Pythia-160M alone reads -0.82 on `prune`, -0.14 on `sparse24` and -0.75 on `headprune`), it loses to κ^{iso} on 15 **of** 15 cells and is beaten even by $\|\epsilon\|^2$ alone. The mis-specified variant scores 0.141: on these three operations the error had been flattering the baseline, on quantisation and low-rank the other way (Appendix E.10).

What this changes in the claim. The coverage number stays seven of eleven but no longer rests on definability, the wrong ground: three of the seven have a weight-space object, we computed it, and it does not rank them, the displacement being a whole projection or half the heads, far outside the regime a quadratic model in w is derived in. That measures where the weight-space model breaks, and is a better claim than the one it replaces. The remaining four (`linear`, `swa`, `kvquant`, `actquant`) have no ΔW at all and are unchanged.

One limit. The trace must converge before it can be read, and on the smaller models it does not everywhere: $7/12$ layers on Pythia-160M, $16/24$ on Pythia-410M, $18/24$ on GPT-2-medium, $24/24$ on Qwen2.5-0.5B, $22/22$ on TinyLlama, so Pythia-160M's three negative cells rest on seven layers. The two fully converged checkpoints show the same thing more weakly (ρ from 0.07 to 0.38 against κ^{iso}'s 0.44 to 0.89), so the conclusion does not depend on the sparse rows.

F OPERATION FAMILIES, DIRECTION AND ANISOTROPY

F.1 ANISOTROPY BY OPERATION

Epsilon-direction overlap is null. Pairwise $|\cos|$ between the per-layer ϵ directions of the five core operations at Pythia-410M is null: means 0.259 within removal, 0.233 within noise, 0.225 across families (medians 0.178, 0.188, 0.182). The CGT split comes from curvature concentration, not raw ϵ direction, removal is not removal because its ϵ points as others' do, but because the loss responds anisotropically to it. The figure that plots this measurement is in the supplementary appendix.
Per-layer values: `results/aniso_*.json` (field `aniso`, one file per model). Pooled statistics: `results/aniso_summary.json`. Pythia-410M interquartile spreads: `quant3` 0.514, `quant4` 0.546, `lowrank` 0.612, `prune` 1.403, `linear` 1.489 (weight noise in a band disjoint from capability removal) and across the nine checkpoints measured here the spread ranks operations against the magnitude model's relative error at $\rho = +0.697$ over 1090 layer–operation pairs.

What "curvature-generic" means. For $\hat{\epsilon}$ uniform on S^{d-1} and symmetric H, $\mathbb{E}[\hat{\epsilon}^\top H\hat{\epsilon}] = \operatorname{tr}(H)/d$ and

$$\operatorname{Var}(\hat{\epsilon}^\top H\hat{\epsilon}) = \frac{2\big[d\operatorname{tr}(H^2) - \operatorname{tr}(H)^2\big]}{d^2(d+2)}, \qquad \text{so} \qquad \frac{\operatorname{Var}(\kappa)}{(\kappa^{\mathrm{iso}})^2} = \frac{2}{d+2}\Big(\frac{d}{r} - 1\Big), \quad r = \frac{\operatorname{tr}(H)^2}{\operatorname{tr}(H^2)}.$$

Both are standard fourth-moment identities for the sphere. $r \le d$ (Cauchy–Schwarz), and for positive semidefinite H it lies in $[1, d]$ as the Hessian's effective rank. Measured layer Hessians have negative directions, so r is a spread diagnostic rather than a PSD rank, and $\kappa \approx \kappa^{\mathrm{iso}}$ says r_ℓ is large, not that the loss surface is flat or isotropic.
And it is, on the layers we measured. The bound $\operatorname{tr}(H^2) \le \lambda_{\max}\operatorname{tr}(H)$ and its corollary $r \ge d/(\lambda_{\max}/\bar{\lambda})$ need $H \succeq 0$, which measured layer Hessians deny. Measured at 128 probes per layer, r runs from 127 at the worst layer to 1.22×10^4, median 944, and each κ^{iso} averages 16 probes: 3.1% spread at the worst layer, 1.2% at the median. So the $\kappa/\kappa^{\mathrm{iso}}$ of 1.13 to 3.99 in Table F.2 is not estimator noise (smallest departure 13 points against a 3.1% band, four times it. Largest 95 times it) but alignment: real compression perturbations occupy directions the network uses, how strongly being what the two families differ in. Neither identity predicts a *value* for a non-uniform ϵ. The argument below concerns the ordering the mechanism predicts.

Proposition 11 (Worst-case directional deviation)**.** *For any unit $\hat{\epsilon} \in S^{d-1}$ and symmetric H,*

$$\big|\hat{\epsilon}^\top H\hat{\epsilon} - \kappa^{\mathrm{iso}}\big| \;\le\; \|H - \kappa^{\mathrm{iso}} I\|_{\mathrm{op}} \;=\; \max_i \big|\lambda_i(H) - \kappa^{\mathrm{iso}}\big|,$$

where $\|\cdot\|_{\mathrm{op}}$ is the spectral norm and $\lambda_i(H)$ the eigenvalues of H.

What this adds over the variance. The identity controls the *typical* direction, Proposition 11 the *worst case*: a large r narrows the band on average, yet one $\hat{\epsilon}$ can sit at the spectral edge. Curvature-genericity is thus an *operation* property, not a checkpoint one (one checkpoint reads generically for rounding and poorly for linearisation) and with $\lambda_{\max}/\bar{\lambda} \ge 10.9$ the worst-case band is at least that factor wide in κ^{iso}.

Proposition 12 (Expected anisotropy of an operation's direction)**.** *Let ϵ be a random unit perturbation with $\mathbb{E}[\epsilon] = 0$ and covariance $\Sigma = \mathbb{E}[\epsilon\epsilon^\top]$, independent of H. Then*

$$\mathbb{E}[\kappa/\kappa^{\mathrm{iso}}] = \mathbb{E}[\epsilon^\top H\epsilon]\,\frac{d}{\operatorname{tr}(H)} = \frac{d\,\operatorname{tr}(H\Sigma)}{\operatorname{tr}(H)}.$$

In particular $\Sigma = I/d$ (the isotropic null the CGT tests against) gives exactly 1, and the ratio is the Schur inner product of Σ with H, rescaled by $d/\operatorname{tr}(H)$.

What this pins down. The mechanism sentence below (rounding inherits no direction, deletion the sublayer's) concerns the two covariances: a near-isotropic Σ scores $\operatorname{tr}(H\Sigma)/\operatorname{tr}(H) \approx 1/d$, and any excess above $1/d$ measures Σ's mass in the loss's high-curvature subspace. The identity does not predict which operations are generic (the CGT reads that).
Selection regret. A top-k selection-regret bound in the supplementary appendix gives $\sum_{\ell\in S^\star} a_\ell - \sum_{\ell\in S} a_\ell \le (1 - 1/\gamma_{\mathrm{eff}})\sum_{\ell\in S^\star} a_\ell$. Compression selects the bottom-k set, where the coefficient is $\gamma'_{\mathrm{eff}} - 1$ rather than $1 - 1/\gamma$, so transplanting the top-k form understates the bound by a factor γ'_{eff}, violated on 9 of the 45 cells of the supplementary appendix, the corrected bound on all 45. Across all 45 model–operation cells at $k = L/4$ both hold: *top-k* regret is a median 1.9% of

$\sum_{S^\star} a$ (maximum 55%, 160M deletion), the *bottom-k* regret compression pays larger, median 10.5%, maximum 122% on Llama-3-8B `quant3`. Only the second is what §G.1 optimises, and past $\gamma'_{\text{eff}} = 2$ the bound permits regret above the optimal subset's own damage, as the 122% cell shows, a post-hoc characterisation of where the isotropy error lands, not an a priori guarantee (`results/cgt_regret_bound.py`).

What the ranking metric hides about the tail. Set-level tail metrics at $k = L/4$ over the eighty-nine core cells (tail recall, Jaccard overlap, relative regret against the oracle) put the corrected score at mean recall 0.603 and Jaccard 0.466 against the magnitude rule's 0.360 and 0.265. Rank correlation and recall disagree in both directions: twenty-three of the eighty-nine cells have correlation above 0.40 at recall at most 0.50, while on Bloom-7B1 recall runs ahead of correlation, 0.66 against +0.54 over its five cells. Hence the boundary of §7 is stated per checkpoint and per operation: a positive global correlation is compatible with a badly picked tail. Relative regret is the metric we drop, unstable when the oracle's k layers are nearly free (`analysis/tail_metrics.py`. The per-checkpoint recall, the named cells and the worked example are in the supplementary appendix).

Proposition 13 (Ordering preservation under κ^{iso} substitution)**.** *With $X_\ell = \log a_\ell$ and $Y_\ell = \log r_\ell$ ($r_\ell = \kappa_\ell/\kappa_\ell^{\text{iso}}$), let $\varepsilon^2 = \mathrm{Var}(Y)/\mathrm{Var}(X)$ and $\rho = \mathrm{Corr}(X, Y)$. Then*

$$\mathrm{Corr}(\log \hat{s},\, X) = \frac{1 - \rho\varepsilon}{\sqrt{1 + \varepsilon^2 - 2\rho\varepsilon}},$$

tending to 1 as $\varepsilon \to 0$, that is, as the anisotropy's cross-layer log-spread becomes negligible next to the damage's.

What the identity is, and what it is not. The expression above holds whenever $\log \hat{s} = X - Y$, that is, whenever $\hat{s}_\ell = a_\ell/r_\ell$. That is true of the *second-order* damage $\frac{1}{2}\|\epsilon_\ell\|^2\kappa_\ell$ and not of the measured a_ℓ, which also carries the linear term and the third-order remainder of Eq. 4. Verifying it by setting $\log \hat{s} := X - Y$ therefore checks an algebraic identity about correlation coefficients, true of any X and Y, and says nothing about the measurement. We did that and it agrees to 2.2×10^{-16}, which is the machine epsilon and not evidence. Substituting the real $\hat{s} = \frac{1}{2}\|\epsilon_\ell\|^2\kappa_\ell^{\text{iso}}$ instead leaves a median $|\log \hat{s} - (X - Y)|$ of 0.303 and moves the correlation by 0.199. The proposition is a statement about the second-order approximation. The measured quantity is reported next to it rather than in place of it.

Table F.1: **Proposition 13 against the measurement**, over the 45 cells that carry a directional κ. "formula" evaluates $(1 - \rho\varepsilon)/\sqrt{1 + \varepsilon^2 - 2\rho\varepsilon}$ from the measured ε and ρ; "measured" is $\mathrm{Corr}(\log \hat{s}, \log a)$ with $\hat{s}$ the score the pipeline actually computes. The gap between them is what the linear term and the remainder cost, and it is larger where ε^2 is larger.

family	cells	median ε^2	formula	measured	Spearman$(\hat{s}, a)$
weight noise (`quant4`, `quant3`, `lowrank`)	27	0.154	0.933	0.695	0.770
removal (`prune`, `linear`)	18	0.375	0.803	0.557	0.574
all	45	0.172	0.921	**0.637**	0.678

The ordering the proposition predicts survives the substitution: ε^2 is smaller on the weight-noise family and the measured correlation is higher there (0.695 against 0.557), which is the order-preservation the mechanism gives that family. The measured median of 0.678 at the rank level is also what §5's per-operation means bracket (0.423 to 0.734). What the proposition does not license is reading its 0.921 as an achieved correlation.

The ordering the mechanism predicts, measured. Rounding perturbs every weight near-independently, so the ϵ it induces at a layer's output sums many contributions and inherits no direction of its own. Deletion sets ϵ_ℓ to minus that sublayer's own output, one learned, low-dimensional vector. So the mechanism predicts quantisation nearest the isotropic null and deletion or linearisation furthest, which Table F.2 shows on nine checkpoints from a quantity no part of the selection pipeline consumes. The families differ in level and in spread alike: the median $\kappa/\kappa^{\text{iso}}$ is 1.47 for weight noise against 2.07 for removal, and its median across-layer IQR 0.53 against 1.21 (per-checkpoint values in `results/aniso_summary_9model.json`. `analysis/verify_generic.py`).
The cross-substitution correlations behind the family split, Spearman between the per-layer damage of two operations measured on the same checkpoint, pool to +0.741 (95% CI $[0.676, 0.795]$, 12

Table F.2: **Median $\kappa/\kappa^{\text{iso}}$ per operation**, one row per checkpoint, over the nine models carrying a directional κ measurement. Linearisation is the largest entry in 9 of 9 rows and a quantisation the smallest in 8 of 9; the exception, GPT-2-medium, is the one checkpoint whose profile also fails on more than one operation (§5).

checkpoint	`quant4`	`quant3`	`lowrank`	`prune`	`linear`
Pythia-160M	**1.13**	1.14	1.26	1.41	*1.96*
Pythia-410M	**1.34**	1.39	1.64	1.70	*2.15*
Pythia-1.4B	**1.40**	1.53	2.01	2.25	*2.96*
GPT-2-medium	1.74	1.46	**1.25**	1.25	*1.99*
GPT-J-6B	**1.64**	1.87	2.04	2.98	*3.99*
TinyLlama-1.1B	1.41	**1.39**	1.77	1.86	*3.47*
Qwen2.5-0.5B	**1.37**	1.47	1.87	1.75	*1.88*
Qwen2.5-7B	**1.74**	1.75	2.39	2.52	*2.76*
Llama-3-8B	**1.13**	1.28	1.53	1.54	*2.56*
median	1.40	1.46	1.77	1.75	2.56

pairs) within a family against -0.039 ($[-0.131, 0.054]$, 24 pairs) across families, the measurement behind the family split of §I.1. The per-checkpoint grid is in the supplementary appendix (Table S11).

F.2 Is the two-family partition recovered from the data?

We test whether the weight-noise vs degree-of-freedom-removing grouping makes the within-versus-between contrast circular. Five operations answer exhaustively: exactly $2^5/2 - 1 = 15$ splits into two non-empty groups, each scored by the contrast's criterion gap $= \overline{\rho}_{\text{within}} - \overline{\rho}_{\text{between}}$ on the Fisher-z pooled correlation matrix, six checkpoints, so our partition is ranked, not merely shown positive. It ranks **first of** 15, gap $= 0.450$ (within 0.552, between 0.102); held-one-checkpoint-out, 4 of 6 folds. Both losing folds move `lowrank` across the boundary, the weakness Table F.2 names independently (low-rank, deletion unseparated by $\kappa/\kappa^{\text{iso}}$). Dropping `lowrank`, the other four rank it **first of** 7, gap $= 0.757$ (within 0.718, between -0.038), recovered on **6** of **6** folds. So the split is data-recovered, not our labelling; only `lowrank`'s side remains unsupported, and the paper does not assert it. Reproduced by `family_recovery.json`.

F.3 The family rule under two continuous knobs

§F.1 tests whether family membership is set by how much of the mechanism a substitution destroys, with predictions registered before each sweep (`21_swa_window_sweep.py`, `31_more_substitutions.py`); both outcomes are reported, non-confirming included.

Window sweep (confirms 4/4 on both models). Sweeping $w \in \{8, \ldots, 512\}$: separation falls monotonically in window size ($+0.122 \to -0.874$ on 160M, Spearman -1.000; $+0.133 \to -0.249$ on 410M); the assignment flips exactly once, at $w = 16$ on 160M and $w = 32$ on 410M, inside the registered bounds; $\rho(\text{position}, a)$ rises monotonically with window on both (Spearman $+1.000$), $\rho(\|\epsilon\|^2, a)$ with window ($+1.000$ on 160M, $+0.571$ on 410M); the $w = 128$ column reproduces the out-of-sample row of §F.1 to three decimals.

Temperature sweep (direction confirms, crossing does not). Table F.4: separation rises with T on both models, the assignment capability-removal from $T = 2$ upward everywhere — the registered direction; the registered *single crossing inside* $[2, 16]$ does not occur. 410M shows no sign change, and 160M's one change is at $T = 2$, the lower bound — $T = 1$ is the identity substitution, its per-layer damage profile numerical noise and its family label uninformative, placing the true crossing in $(1, 2)$ below the sweep's resolution. Of the four registered predictions, $\rho(\text{position}, a)$ and $\rho(\|\epsilon\|^2, a)$ fall with T on 410M (-0.348, -0.143), not 160M ($+0.086$, $+0.232$). We claim the window sweep as a confirmed test of the rule, the temperature sweep as directional support only.

Three further registered substitutions. 2:4 structured sparsity on the attention projections (registered weight-noise), int4 KV-cache quantisation (weight-noise) and GQA-style head merging (capability-removal) were committed before measurement. Five of six model–substitution cells land as registered: KV-cache quantisation and head merging confirm on both Pythia-160M and 410M (-0.920/-1.406, $+1.185$/$+0.350$), 2:4 sparsity on 410M (-1.160) but not 160M ($+0.357$, i.e. capability-removal). The pair was chosen because the readings disagree on it — 2:4 sparsity removes half the weights leaving softmax and every head intact, head merging the reverse — so the refutation lands on the discriminating case.

Table F.4: Attention-temperature sweep. Separation is mean ρ (capability-removal) minus mean ρ (weight-noise); positive means capability-removal. $T = 1$ is the identity substitution, reported for completeness.

	Pythia-160M			Pythia-410M		
T	sep.	$\rho(\text{pos}, a)$	$\rho(\|\epsilon\|^2, a)$	sep.	$\rho(\text{pos}, a)$	$\rho(\|\epsilon\|^2, a)$
1	-0.923	$+0.532$	$+0.595$	$+0.244$	-0.077	-0.122
2	$+1.206$	-0.895	$+0.769$	$+1.374$	-0.695	$+0.312$
4	$+1.129$	-0.853	$+0.832$	$+1.387$	-0.676	$+0.285$
8	$+1.143$	-0.846	$+0.832$	$+1.392$	-0.671	$+0.138$
16	$+1.185$	-0.797	$+0.825$	$+1.415$	-0.683	-0.030
32	$+1.290$	-0.811	$+0.720$	$+1.415$	-0.683	-0.099

F.4 Cross-lineage transfer (fifteen checkpoints, nine lineages)

The order-preservation §F.1 reports was registered before measuring a seventh lineage (Mistral-7B). Across fifteen checkpoints spanning nine lineages — Pythia (160M/410M/1.4B), GPT-2, GPT-J, Llama (TinyLlama, Llama-3, LLaMA-13B), Qwen (Qwen2.5-0.5B/7B, Qwen3-8B), Gemma-2, Mistral, OLMoE and BigScience (Bloom-7B1) — the corrected score improves on salience in 82 of 90 model–operation cells (91%): the five core operations on all fifteen checkpoints (75) plus Table E.7's three registered out-of-sample operations on their five models (15), and is never negative on weight-noise (n=55: quant4/quant3/lowrank on all fifteen, `swa` and `actquant` on their five; minimum ρ=+0.1498 by `verify_lowerbound.py`). Four of the seven exceptions are Bloom-7B1, the registered out-of-family probe failing its own validity check (Appendix F.7); excluding it, 82 of 85 over fourteen checkpoints. LLaMA-13B, added for the OWL comparison of Appendix G.12, improves on all five cells, up to $-0.590 \rightarrow +0.850$ on deletion. Counts are WikiText cells only; on the held-out C4 subset the same profiles cover eight of these checkpoints again, improving 39 of 40 cells with no negative on the 24 weight-noise cells — deliberately *not* pooled into the 90 (Appendix D.7), the same model–operation pairs on a second corpus. Four caveats: (a) of the three regressing cells, two are the GPT-2-medium residual of §5, one is 410M `actquant` ($+0.907 \rightarrow +0.903$); (b) negative weight-noise salience also appears on Gemma-2-9B, Llama-3-8B and GPT-J-6B, not only Mistral-7B, and the correction restores it on all four; (c) one weight-noise cell sits at ρ=0.1498 — the guarantee is no negative cell, not that every cell clears $+0.15$; (d) on Pythia-1.4B the corrected score's *selection* at k=$L/4$ (prune) is significantly worse than deepest-k on LAMBADA (-0.15 retention), so Table G.1's envelope "no losses at $L/4$" holds on its three families but not every checkpoint. One mismatch surfaced and was fixed during replication: the GPT-J-6B transfer had run at $512/8$ against $1024/32$; rerunning at $1024/32$ moved the corrected ρ by at most 0.04 (cross-layer κ^{iso} rank correlation 0.9978). All thirteen models passed a re-check of ρ (recomputation to 10^{-16}), layer alignment (a-direct transfer vs. substitution damage, ρ=1.0) and κ^{iso} consistency.

F.5 The genericity test used as a predictor, and what that refuted

§I.1 fits τ on eleven operations and reports leave-one-out accuracy on the same eleven. Here we freeze $\tau = 0.775$, register a family for two never-seen operations, and only then measure. Registration, SHA-256 and execution log ship with the code.

The two operations, and why they were chosen. Both make the eliminate/perturb rule and the naive "how much machinery did you break" reading point *opposite* ways, the criterion of Appendix A.2(C). `ropetrunc` keeps the lowest 25% of RoPE frequency pairs, replacing the rotation on the rest by the identity, every weight used, softmax intact (naive: mild), but fine-grained relative position is a functional degree of freedom, removed. Registered: capability removal. `headscale` scales head h's output by a fixed $s_h \sim U(0.5, 1.5)$, every head touched (naive: severe), but the functional form is untouched and the output projection absorbs it. Registered: weight noise.

A first run was voided by a check that knows nothing about the registration. Damage was positive at every layer while the symmetric-difference κ was negative at almost every layer, which Eq. 1 forbids. A baseline-convention error in the new script: B is the *sum* of the losses at $\pm t$, so a constant baseline offset does not cancel but enters κ as $2c/(t^2\|\epsilon\|^2)$. That one constant explains the whole discrepancy, at $R^2 = 0.96$ to 1.000, with the same $c \approx -0.13$ from both operations. The quadratic model never failed. The run below uses the per-batch baseline and every cell passes the sign check.

Table F.5: **Two operations the paper does not contain, scored against a frozen $\tau = 0.775$.** z is $|\mathrm{IQR} - \tau|$ in units of the leave-one-layer-out standard deviation of the IQR; $z < 1$ cells are not resolved and are no evidence either way. *Behaves like* reads the operation's own salience correlation against the split §5 measures ($+0.31$ weight noise, -0.09 capability removal).

model	operation	IQR	z	CGT verdict	$\rho(\|\epsilon\|^2, a)$	behaves like
Pythia-160M	`ropetrunc`	0.748	0.29	*unresolved*	$+0.797$	weight noise
Pythia-410M	`ropetrunc`	0.998	2.17	capability removal	$+0.492$	weight noise
Pythia-160M	`headscale`	0.815	0.45	*unresolved*	-0.056	capability removal
Pythia-410M	`headscale`	0.870	1.10	capability removal	$+0.134$	capability removal

What was refuted. Both registrations: `ropetrunc` (registered capability removal) behaves like weight noise on both checkpoints, `headscale` (registered weight noise) like capability removal on both. The rule of §I.1 got both wrong, our strongest evidence that it is a heuristic, not a criterion.
What was not refuted. The measurement: in three of the four cells the frozen threshold lands on the same side as the operation's own damage behaviour (exception: `ropetrunc` on Pythia-410M), the case for measuring rather than reasoning that §4 makes.
A resolution limit the fitted set hides. Two of the four cells sit within one leave-one-layer-out standard deviation of $\tau = 0.775$, against 3 of the paper's 45 fitted cells (6.7%), which sit far from the boundary (median IQR 0.526 weight noise, 1.211 capability removal): the leave-one-out 0.867 of §I.1 says nothing about near-boundary operations, and one paper cell (Pythia-160M `linear`) is both misclassified and jackknife-unstable. A near-τ value is *no answer*, not a weak one.
A repair we propose but do not claim. Abstaining when $|\mathrm{IQR} - \tau| < 2\,\mathrm{sd}_{\mathrm{jack}}$ leaves 80% coverage at 0.917 accuracy against 0.867 at full coverage. A proposal, not a result.

F.6 EXP80: A REGISTERED FROZEN TEST ON SIX NEVER-SEEN DEPLOYMENT CELLS

Before any measurement after EXP79 we registered three never-seen deployment operations (per-query top-16 sparse attention, `kvevict`. Pairwise head averaging, `headavg`. Attention-output dropout at inference with p=0.5 and $\times 2$ compensation, `outdrop`) with the frozen threshold $\tau = 0.775$ from `family_predictor.json`, the preregistered assignments (`capability_removal`, `capability_removal`, `weight_noise`), the $z < 1$ abstention band and the behaviour rule (nearest family mean, $+0.31$ / -0.09. Unresolved within 0.02). Measurement reproduces EXP79's protocol on Pythia-410M and GPT-2-medium, κ on four batches, damage on sixteen.

Model	Op	IQR($\kappa/\kappa^{\mathrm{iso}}$)	$\mathrm{sd}_{\mathrm{jack}}$	z	CGT	Behaviour
410M	kvevict	0.849	0.041	1.83	removal	noise
410M	headavg	0.894	0.118	1.01	removal	noise
410M	outdrop	0.854	0.107	0.74	*abstain*	removal
GPT-2-m	kvevict	1.225	0.033	13.81	removal	removal†
GPT-2-m	headavg	0.849	0.066	1.12	removal	noise
GPT-2-m	outdrop	0.630	0.026	5.48	noise	noise

Verdict (as registered). Three of six cells sit at or inside the noise band (z=0.74 abstains by the registered rule. The two $z \leq 1.12$ cells are only marginally resolved). On the five resolved cells the family rule and the frozen threshold agree in all five, the threshold matches the operation's own damage behaviour in two of five, and a sixth, `kvevict` on GPT-2-medium, is a sign-consistency outlier ($19/24$ layers, below the 80% rule, reported degenerate). Every failure lies near τ, four cells with IQR 0.849–0.894, z 0.74–1.12, three classified removal while their behaviour reads noise: *near the threshold the statistic and the rule agree with each other and both disagree with the operation's behaviour*.
Registration note. `outdrop` was registered with a mechanism tension: its $\epsilon = \pm y_\ell$ is structured along the layer's own output directions, so the Σ-identity of Proposition 12 predicts a removal-side reading while the operation's form says weight-noise. The outcome is model-dependent, removal-like on Pythia-410M (the mechanism wins, the statistic abstains), weight-noise-like on GPT-2-medium (the statistic agrees), two observations, not a law.

F.7 EXP82: THE REGISTERED FOURTH-FAMILY PROBE, AND WHERE THE WHOLE CHAIN FAILS

Why Bloom-7B1 was out of the audit, and why it no longer is. The audit's inclusion rule is the relative third-order remainder $|a - (A + B)|/|a|$ with A and B fitted along ϵ_ℓ, and Bloom-7B1 failed it at 0.92 on the harness the first run used. That harness dropped Bloom's residual stream.

`BloomAttention` returns its output with the residual already added, the replacement modules did not add it back, and so `prune` and `linear` removed the whole stream (single-layer damage 7.4–15.2 nats) while the probe amplitude, set from `linear`'s ϵ_ℓ, came out about $50\times$ too large. With the residual restored (`common.py 56b95b95`; every other adapter reproduces bit for bit, and Bloom's weight-space operations to 5.5×10^{-8}) the remainder is 0.19, inside the gate, and Bloom is scored like every other checkpoint. Its rate moves from 0.94 to 0.19 and the normaliser split stands as it was: centred mean 1.68 against 5.32, exact permutation $p = 1.2 \times 10^{-5}$. EXP105's operation-free re-read of Bloom predates the fix and is not repeated here. The first run's files are kept, labelled, in `superseded/bloom_residual_bug/`.

Before measurement we registered a fourth envelope family, `bigscience/bloom-7b1` (BigScience lineage, base model, 7.1B, 30 attention layers, ALiBi, no RoPE, structurally distinct from Qwen2.5-7B, Llama-3-8B and Gemma-2-9B), with the EXP76 criteria reproduced verbatim and hashed into the verdict (`35876245...`): **R1** the transfer correction must improve prune damage prediction, $\rho(\frac{1}{2}\|\epsilon\|^2\kappa^{\text{iso}}, a) > \rho(\|\epsilon\|^2, a)$. **R2** at $k{=}L/4$, zero tasks where deepest-k significantly beats the corrected selection. **R3** at $k{=}L/2$, at least one task where deepest-k significantly beats it. **R4** the free depth prior beats the corrected selection on at most one task at $L/4$. CONFIRMED iff R1, R2 and R3 hold. Stages 2–5 are the exact EXP76 commands (the bloom adapter passed its two-tier gate first). ALiBi never enters the substitution path, the same convention as Gemma2's sliding-window mask.

Verdict (as registered). On the fixed harness R1 is CONFIRMED (ρ $+0.013 \to +0.350$ on prune; the first run read $+0.233 \to -0.033$). R2 is REFUTED on both runs: deepest-k beats the corrected selection on all four tasks at $L/4$ (fixed harness: LAMBADA -0.08, ARC-Easy -0.09, MMLU -0.08, HellaSwag -0.23). R3 is confirmed on both (HellaSwag -0.08 at $L/2$). R4 is REFUTED on both. Overall: PARTIAL, as registered; the ledger keeps the first run's verdict.

Where it fails is the set, not the reading. On the fixed harness the quadratic model holds on Bloom (third-order remainder median 0.003, against 0.017 on Gemma-2-9B) and the corrected score beats the magnitude rule on all five operations. What fails is the step from single layers to a set. Deleting any one of Bloom's attention sublayers costs 0.001 to 0.021 nats, so every per-layer score is ranking near-ties, and the eight the profile picks (0, 3–9) cost 2.10 nats together against 0.05 summed, $42\times$. Deepest-k's contiguous block compounds less ($7\times$), which is R2. A search restricted to the profile's 16 cheapest layers finds eight that cost 0.12 nats, within 0.02 of the full search (Appendix G.9).

F.8 The effective rank, and a registered prediction it refutes

§F.1 reads the variance identity as: the isotropic approximation is accurate where the layer Hessian's effective rank r_ℓ is large. Two commitments follow, one confirmed, one refuted.

r_ℓ is measurable forward-only and $r_\ell \ll d$. CurvProbe's probes give r_ℓ free: their spread about κ_ℓ^{iso} gives $V_\ell = \text{Var}(\hat\epsilon^\top H_\ell \hat\epsilon)/(\kappa_\ell^{\text{iso}})^2$, $r_\ell = 2/V_\ell$ while $r_\ell \ll d$. Over 60 layers of Pythia-160M, Pythia-410M and Qwen2.5-0.5B (128 probes each), r_ℓ runs from 86 to 12,160 against d of 1.6–2.1 million, so $r_\ell/d \le 0.0065$ everywhere, the $\approx 2/r_\ell$ form §F.1 quotes, not an assumption. On larger checkpoints r_ℓ is smallest earliest, largest last (127 at layer 2, 12,160 at layer 23 on Pythia-410M), unregistered, not built on.

The prediction it was supposed to license is refuted. Registered before running: small-r_ℓ layers are where the correction fails, so $\sqrt{2/r_\ell}$ should track the measured anisotropy $|\kappa_\ell/\kappa_\ell^{\text{iso}} - 1|$ across layers for capability-removal operations, *weakly* for weight noise (near-isotropic everywhere), where a strong correlation is evidence against. Both halves:

	L	`linear`	`prune`	`quant4`	`quant3`	`lowrank`	capability	weight noise
Pythia-160M	12	-0.66	-0.10	$+0.12$	$+0.02$	$+0.11$	-0.38	$+0.08$
Pythia-410M	24	$+0.64$	$+0.62$	$+0.33$	$+0.15$	$+0.45$	$+\mathbf{0.63}$	$+0.31$
Qwen2.5-0.5B	24	$+0.62$	$+0.46$	$+0.72$	$+0.31$	$+0.47$	$+0.54$	$+0.50$

Only Pythia-410M satisfies the registration: Pythia-160M runs *backwards* on linearisation (-0.66). Qwen2.5-0.5B passes the first half, fails the second, weight noise correlating as strongly as capability removal ($+0.50$ against $+0.54$), `quant4` alone at $+0.72$ exceeding either capability operation, which the registration calls evidence against, not for. Averaged over the three, the family separation the prediction rests on is -0.03.

A refuted registration: r_ℓ is measurable and confirms the bound's form, but carries no per-layer diagnostic, so §F.1's bound stays a bound and the anisotropy $\kappa_\ell/\kappa_\ell^{\text{iso}}$, needing the operation's ϵ, hence post-hoc, remains the only per-layer diagnostic we can support. `analysis/exp71_verdict.py`

re-scores the three runs. The original scoring in `71_effective_rank.py` tested only the first half of the registration and passed Qwen2.5-0.5B in error.

F.9 Does the anisotropy of H_ℓ buy anything the scalar does not?

Measuring H_ℓ anyway, we tested whether its anisotropy moves choices a scalar metric cannot see. Input whitening already carries the result, so the scalar suffices (the experiment is in the supplementary appendix).

What the isotropic average costs. The same measurement prices the approximation the profile makes. Over the 30 cells where a directional κ_ℓ was measured, ranking by $\|\epsilon_\ell\|^2$ alone reaches mean $\rho = 0.300$, multiplying by the isotropic profile reaches 0.657, and substituting the measured directional κ_ℓ reaches 0.892. The profile beats displacement on 28 of the 30 and the directional reading beats the profile on the same 28, so the isotropic average costs 0.235 in mean ρ, and the size of that cost tracks how anisotropic the cell is ($\rho = 0.536$ between the two). The directional quantity needs the operation in hand, which is what the profile is read without (`results/kappa_ceiling.json`, `13_aniso_by_substitution.py`).

F.10 Is the operation's direction readable without running the operation?

The isotropic average is not the only free probe. Every substitution's `capture_eps` computes the clean sublayer output y_ℓ and throws it away, so a third read can be taken along y_ℓ, a second $2L$-pass read that needs no operation, alongside the isotropic Gaussian and the directional ϵ_ℓ itself. For `prune` this third read *is* the directional one: the substitution returns zero, so $\epsilon_\ell = -y_\ell$, and $\kappa(u) = u^\top H_\ell u$ being quadratic, $\kappa(-\epsilon_\ell) = \kappa(\epsilon_\ell)$. Table F.6 reads all three at the same site, normalised by the same $\|\epsilon_\ell\|^2$.

Table F.6: **The clean-output read against the isotropic one.** Change in Spearman ρ against measured per-layer damage when the isotropic probe is replaced by the clean sublayer output y_ℓ, over five checkpoints spanning four architectures (106 layers). The last row is the cosine between ϵ_ℓ and y_ℓ for `prune`, which is -1 to float32 precision on every checkpoint: there the two reads are the same quantity, and κ agrees layer for layer with zero relative difference. The read helps wherever the operation moves the sublayer along a fixed direction and fails on the quantisers, which round elementwise and leave no direction to read. Recomputed from `results/dirprobe_*.json`, `112_direction_probe.py`.

	ρ gain of the clean-output read over κ^{iso}				
operation	Pythia-160M	Pythia-410M	TinyLlama-1.1B	GPT-2-medium	Qwen2.5-0.5B
`prune`	$+0.552$	$+0.834$	$+0.177$	$+0.049$	$+0.041$
`linear`	$+0.035$	$+0.840$	$+0.146$	$+0.079$	$+0.150$
`lowrank`	$+0.028$	-0.393	$+0.158$	$+0.077$	$+0.064$
`quant4`	-0.063	-0.081	-0.038	$+0.028$	$+0.075$
`quant3`	0.000	-0.145	$+0.097$	$+0.191$	-0.089
$\cos(\epsilon_\ell, y_\ell)$, `prune`	-1.0000	-1.0000	-1.0000	-1.0000	-1.0000

The gain is positive in fourteen of the fifteen cells outside the quantisers, and on `prune` it is not a gain but an identity. What the quantisers show is the complement: where the perturbation is elementwise rounding rather than a displacement along a direction, there is no direction for the clean output to recover, and the read is as good or as bad as the isotropic one.

At 7–9B the read beats Block Influence where the profile does not. On six grid checkpoints Block Influence ranked deletion better than the isotropic profile (five after the Bloom fix of Appendix F.7). We registered that the clean-output read would out-rank Block Influence on at least five of the six and match its $k{=}L/4$ tail recall on at least five (`prereg_kclean_bi.md`). It did both on all six (Table F.7), and on none of them did it rank linearisation below the profile. Gemma-2-9B, added after registration, needs the probe after its post-attention norm, which rescales any perturbation along y_ℓ before the residual stream sees it; read there, and scored on the second-order term itself rather than on a magnitude taken at a different site, it ranks deletion at 0.626 against Block Influence's 0.118 (`prereg_kclean_bi_rev1_gemma_site.md`, `_rev2_gemma_score.md`).

The endpoint. On Llama-3-8B the read names exactly direct measurement's eight layers. Deleted under the protocol of Table 1 they give WikiText-2 perplexity 14.28 against Block Influence's 16.22 and LAMBADA 0.601 against 0.479 (paired McNemar 305:62, $p = 1.1 \times 10^{-39}$). On GSM8K, re-evaluated in one run with Block Influence's and the isotropic sets and a clean arm that matches the

Table F.7: **The directional read against Block Influence on deletion.** Spearman ρ with measured per-layer damage, and the share of direct measurement's $k{=}L/4$ cheapest layers each score also names. The first six rows are the registered set. Bloom-7B1 on the fixed harness; Gemma-2-9B read after its post-attention norm.

	ρ with damage			tail recall at $k{=}L/4$		
checkpoint	BI	isotropic	directional	BI	isotropic	directional
Llama-3-8B	$+0.896$	$+0.780$	$\mathbf{+0.963}$	0.75	0.75	**1.00**
OLMoE-1B-7B	$+0.750$	$+0.491$	$\mathbf{+0.915}$	0.25	0.50	**0.75**
GPT-J-6B	$+0.737$	$+0.680$	$\mathbf{+0.967}$	0.71	**0.86**	**0.86**
OLMo-1-1B	$+0.682$	$+0.453$	$\mathbf{+0.871}$	0.50	0.50	**0.75**
Qwen2.5-7B	$+0.670$	$+0.548$	$\mathbf{+0.836}$	**0.86**	0.57	**0.86**
Bloom-7B1	-0.276	$+0.350$	$\mathbf{+0.853}$	0.14	0.57	**0.71**
Gemma-2-9B	$+0.118$	$+0.543$	$\mathbf{+0.626}$	0.20	**0.60**	**0.60**

released one on all 200 problems, the set keeps 0.405 against 0.370 (28:21, $p = 0.39$) and against the isotropic set's 0.240 (47:14, $p = 2.7 \times 10^{-5}$). Moved unchanged to the Instruct derivative it leads Block Influence on LAMBADA (0.540 against 0.462, 59:20, $p = 1.3 \times 10^{-5}$), is not separated from it on ARC-Easy, HellaSwag or MMLU, and trails the isotropic set on ARC-Easy (0.593 against 0.641, $p = 0.002$). Its Instruct GSM8K, 0.665, comes from a run whose clean arm differs from the released one on 17 of 200 problems, so it is reported and not paired.

On a second checkpoint the read improves on the set it replaces, not on Block Influence. The test was registered before it ran (EXP119, `prereg_olmoe_directional_endpoint.md`): OLMoE-1B-7B, deletion at $k{=}L/4{=}4$ of 16, one-shot, fp32, every arm of Table F.8 in one process per task group. Against the isotropic set it replaces, the read gains in loss (3.151 nats against 3.502), on LAMBADA (0.364 against 0.277, 297:122, $p = 6.2 \times 10^{-18}$), on ARC-Easy ($p = 0.015$) and on HellaSwag ($p = 0.046$), and is not separated on MMLU ($p = 0.39$). Against Block Influence both registered criteria fail: 3.151 nats against 3.070, and LAMBADA 0.364 against 0.458 (156:342, $p = 4.9 \times 10^{-17}$). ARC-Easy goes to the read (46:27, $p = 0.034$), HellaSwag and MMLU to Block Influence (18:41, $p = 0.0038$; 19:42, $p = 0.0044$).

The ranking is not what falls short: the read's stays ahead of Block Influence's ($+0.915$ against $+0.750$, Table F.7), and its set holds three of direct measurement's four layers. The fourth, layer 4 where direct measurement has 12, costs it the loss margin: direct measurement's own set leads Block Influence at 2.993. On LAMBADA no per-layer set reaches Block Influence, direct measurement's included (0.344). There what decides is how the deleted layers combine, not how each ranks alone, the case Appendix G.9 takes up. GSM8K is not run: the clean model scores 0.085, below the harness floor.

Table F.8: **EXP119: every arm at the endpoint on OLMoE-1B-7B.** Deletion at $k{=}4$ of 16, one-shot, fp32. Loss is mean token NLL (nats) under the WikiText-2 protocol; LAMBADA 2000 items, ARC-Easy 479, HellaSwag and MMLU 500, every arm on the same items. Bold is best among the compressed arms, underline second. From `downstream_olmoe_k4_endpoint_{A,B}.json`.

arm	layers deleted	loss	LAMBADA	ARC-Easy	HellaSwag	MMLU
Clean	—	2.652	0.678	0.653	0.554	0.386
CurvProbe, directional	4,6,7,14	3.151	<u>0.364</u>	<u>0.603</u>	0.482	0.320
Block Influence	8,11,13,14	<u>3.070</u>	**0.458**	0.564	<u>0.528</u>	**0.366**
CurvProbe, isotropic	3,4,6,7	3.502	0.277	0.562	0.448	0.334
Direct measurement	6,7,12,14	**2.993**	0.344	**0.618**	0.512	0.332
Deepest-k	12–15	4.050	0.123	0.534	**0.560**	<u>0.360</u>
Free depth prior	1–4	4.276	0.126	0.466	0.368	0.298

The registered Gemma endpoint failed. Scored on the held-out stream, the clean-output set on Gemma-2-9B costs 4.00 nats against direct measurement's 3.02, outside the registered 0.10. A first run, whose set was chosen on mixed scales, is reported as such (2.93, `e2e_gemma29b_prune_sets_postnorm.json`). Every per-layer set loses on this checkpoint, direct measurement's included, and Appendix G.9 measures why.

Table G.1: **The budget at which per-layer ranking stops being the right object.** Retention difference against deepest-k, 95% paired-bootstrap CIs, n=2000 items per task, deletion one-shot; bold: excludes zero in our favour, †: favours deepest-k. Three families derived the envelope; Bloom-7B1 was registered as a fourth and fails the $L/4$ side (fixed harness, Appendix F.7). Per-arm values: Appendix H.6.

	Qwen2.5-7B (L=28)		Llama-3-8B (L=32)		Gemma-2-9B (L=42)		Bloom-7B1 (L=30)	
task	$L/4$	$L/2$	$L/4$	$L/2$	$L/4$	$L/2$	$L/4$	$L/2$
LAMBADA	$\mathbf{+0.36}$	$\mathbf{+0.08}$	$\mathbf{+0.32}$	$-0.08^{\dagger}$	$\mathbf{+0.11}$	$-0.01^{\dagger}$	$-0.08^{\dagger}$	$\mathbf{+0.04}$
ARC-Easy	$\mathbf{+0.42}$	$+0.01$	$\mathbf{+0.12}$	$-0.13^{\dagger}$	$\mathbf{+0.11}$	$\mathbf{+0.11}$	$-0.09^{\dagger}$	$+0.02$
MMLU	$\mathbf{+0.11}$	$-0.07^{\dagger}$	$+0.03$	$-0.15^{\dagger}$	$+0.00$	-0.03	$-0.08^{\dagger}$	$+0.02$
HellaSwag	$\mathbf{+0.06}$	$-0.18^{\dagger}$	-0.02	$-0.28^{\dagger}$	$-0.06^{\dagger}$	$-0.14^{\dagger}$	$-0.23^{\dagger}$	$-0.08^{\dagger}$

G SELECTING LAYER SETS

G.1 DOES IT PICK BETTER LAYERS, AND WHERE DOES THAT STOP?

Does the score pick the cheaper-to-compress layers? Over `prune`, `lowrank`, `quant3`, four checkpoints and two budgets it takes **22** of 24 settings against deepest-k, ahead of the free depth prior ($18/24$) and $\|\epsilon\|^2$ ($10/24$), behind direct measurement of a_ℓ ($24/24$), the oracle it approximates and the better route once the operation is chosen (§5). All four counts at 2σ against three-seed noise, $22/24$ single-seed too (Appendix G.2). Linearisation, the fourth operation, needs an alignment step whose bf16 noise is unresolved (below). Published selectors, same grid: Shortened-LLaMA (Kim et al., 2024) anti-correlated with true damage on four of five models. AWQ's salience and a reconstruction-error selector below deepest-k on average. OWL (Yin et al., 2024) below ours on eleven of twelve checkpoints ($p = 0.003$). Best of them, ShortGPT's Block Influence (Men et al., 2025), $14/24$ to our $22/24$ and 13 of 16 checkpoints lost (Appendices E.9, G.12, G.10). Held-out C4 narrows it ($6/15$ wins, $5/15$ ties, $4/15$ losses). Under real GPTQ, AWQ and SparseGPT reconstruction all close within a point (Appendices G.6, H.2, H.3).

Per-layer ranking is right at $L/4$, wrong at $L/2$ (Table G.1): doubling k reverses the advantage rather than halving it. Deletion, three derivation families: eight cells won, three tied, one lost at k=$L/4$. Two won, two tied, eight lost at k=$L/2$, per-layer ranking pricing layers one at a time, interactions dominating at $L/2$ (Appendix C.5). The *transfer* fails, not per-layer pricing: direct measurement of a_ℓ stays ahead of deepest-k at the same budget (-1.03 on Llama-3-8B at k=16, Table G.4. -0.86, -2.48 on Pythia-1.4B and Qwen2.5-0.5B at k=$L/2$, Table G.3), §7's reading at $L/2$.

Linearisation arms withdrawn: at k=7 they select *identical* layers to the free prior under every operation, and 44 of 48 margins are unresolved against a 0.19–0.57 nat alignment-noise band, leaving the $L/2$ collapse on Qwen2.5-7B ($+3.16$). The one-shot operations are deterministic, unaffected, and carry this section and §6 (Appendix D.4).

Ranking survives what generation does not. Eight-shot GSM8K: on Qwen2.5-7B at $L/4$, retaining 92% of clean MMLU, generation collapses for every arm. On Llama-3-8B it survives and the free prior fails instead (12% against our 53%. Appendix H.11).

G.2 PER-CELL SELECTION RESULTS

§G.1 reports the pooled scoreboard. Table G.2 gives every cell. On `lowrank` and `quant3` the corrected score tracks direct measurement of a_ℓ to within a few hundredths of a nat, as Eq. 1 predicts for curvature-generic perturbations. On `prune`, where the magnitude score collapses by 2.3 to 5.8 nats, it recovers most of the distance to direct measurement, and its only two trailing cells are `prune` on Pythia-1.4B, the checkpoint the gate of Appendix G.11 routes to direct measurement in advance.

Practice's own scores land below both: Table G.3 prices three deployment baselines (activation-weighted weight salience (Lin et al., 2024), Block Influence (Men et al., 2025), a reconstruction-error selector) on the same 24-cell grid, repeated at 7B scale in Table G.4. Two average *worse* than deleting the deepest layers (mean $+1.5$, $+1.0$ nats). The best, Block Influence, wins $14/24$ against the corrected score's $\mathbf{22/24}$ and mean damage -0.29 against -0.40 (single-seed. Three-seed re-adjudicated, -0.30, -0.41), exactly the two Qwen2.5-7B deletion cells, where a nearest-layer similarity score should be most informative. Everywhere else it loses, including both Qwen2.5-0.5B deletion cells the corrected score wins.

Not a single-seed artefact: the B2 campaign re-measures all twenty-four one-shot cells on three evaluation seeds, selection fixed on the seed-2 corpus, so the across-seed spread is pure sampling noise. Calling a cell a tie unless its mean advantage exceeds 1σ or 2σ of its cross-seed

spread leaves the corrected score at $22/24$ under both margins, direct measurement at $24/24$ at 2σ, the free depth prior at $18/24$, $\|\epsilon\|^2$ at $9/24$ (1σ. $10/24$ at 2σ). Pooling the eleven single-seed linearisation cells gives $34/35$ for direct measurement and $\mathbf{28/35}$ for the corrected score (`analysis/b3_full35_rejudication.py`). Those thirty-five predate Qwen3-8B, whose two linearisation cells, both corrected-score wins, give $\mathbf{30/37}$ across all four selection-tested operations, not §G.1's headline, since the retraction below shows its thirteen linearisation cells to be noise-unresolved, leaving $22/24$ on the three deterministic operations. Against the depth prior §G.1 quantifies, corrected-score and free-prior selections overlap (mean Jaccard 0.55 at $k = L/4$) while the corrected score's marginal wins fall where they diverge (mean Jaccard 0.43, `analysis/free_prior_divergence.py`).

Table G.2: **Selection for three substitutions the profile was never tuned on**, four models, two budgets ($k = L/4$, $L/2$). ΔNLL relative to deepest-k, negative better, best per row in bold. Deletion and quantisation are one-shot in deployment (Men et al., 2025; Frantar et al., 2023): the faithful comparison, not an omitted training stage.

		low budget ($k = L/4$)				high budget ($k = L/2$)			
sub.	model	a_ℓ	**ours**	free	$\|\epsilon\|^2$	a_ℓ	**ours**	free	$\|\epsilon\|^2$
`prune`	Pythia-1.4B	$\mathbf{-0.00}$	$+0.08$	$+0.18$	$+2.95$	-0.86	$+0.98$	$\mathbf{-0.89}$	$+2.26$
	Qwen2.5-0.5B	$\mathbf{-1.49}$	-1.30	$+1.06$	$+5.80$	$\mathbf{-2.48}$	$\mathbf{-2.48}$	-0.54	$+3.98$
	GPT-J-6B	$\mathbf{-0.02}$	-0.00	-0.00	$+3.77$	$\mathbf{-0.61}$	-0.33	-0.48	$+2.44$
	Qwen2.5-7B	-0.88	$\mathbf{-1.33}$	$\mathbf{-1.33}$	$+5.85$	$\mathbf{-2.75}$	-1.67	-2.74	$+3.63$
`lowrank`	Pythia-1.4B	$\mathbf{-0.044}$	-0.038	-0.035	-0.043	$\mathbf{-0.085}$	-0.070	-0.034	-0.052
	Qwen2.5-0.5B	$\mathbf{-0.364}$	-0.355	-0.366	$+1.557$	$\mathbf{-0.705}$	-0.694	-0.597	$+1.638$
	GPT-J-6B	$\mathbf{-0.004}$	-0.002	$+0.002$	$+0.003$	$\mathbf{-0.008}$	-0.001	$+0.000$	$+0.003$
	Qwen2.5-7B	-0.081	$\mathbf{-0.089}$	$\mathbf{-0.089}$	$+0.137$	-0.104	$\mathbf{-0.129}$	$+0.125$	$+0.117$
`quant3`	Pythia-1.4B	$\mathbf{-0.42}$	-0.41	-0.40	-0.41	$\mathbf{-1.09}$	-1.05	-1.06	-1.05
	Qwen2.5-0.5B	$\mathbf{-0.18}$	-0.15	$+0.80$	$+0.90$	$\mathbf{-0.23}$	-0.21	$+1.14$	$+2.04$
	GPT-J-6B	$\mathbf{-0.06}$	-0.03	-0.04	-0.04	$\mathbf{-0.06}$	-0.05	-0.06	$+0.00$
	Qwen2.5-7B	$\mathbf{-0.10}$	-0.07	-0.08	-0.07	$\mathbf{-0.14}$	-0.12	-0.11	-0.11

Table G.3: **Deployment baselines over the same 24-cell grid.** Each selector scores layers and keeps the lowest-k; "wins" counts cells where it beats deepest-k "strictly", a tie being an identical layer set reported in the row. AWQ is activation-weighted weight salience (Lin et al., 2024), Block Influence $1-\cos(h_\ell, h_{\ell+1})$ (Men et al., 2025), Recon the reconstruction error $\mathbb{E}\|h_{\ell+1} - h_\ell\|^2$. Two of the three average *worse* than deleting the deepest layers, and the strongest still trails the corrected score on both counts and mean damage. The *wins* are unadjusted single-seed counts; the noise-adjudicated recount over three seeds reads $22/24$ for the corrected score and $18/24$ for the free depth prior (Appendix G.2). Every entry is read from `scoreboard_{1.4b,gptj6b,qwen05,qwen7b}.json`, the four checkpoints that carry all seven selectors on the same grid; `analysis/verify_deploy.py` re-derives the column.

selector	wins / 24	mean ΔNLL vs deepest-k
direct measurement of a_ℓ	24	-0.53
$\frac{1}{2}\|\epsilon\|^2\kappa^{\text{iso}}$ (**ours**)	**21** ($+1$ tie)	$\mathbf{-0.40}$
free depth prior	16 ($+1$ tie)	-0.23
Block Influence (Men et al., 2025)	14	-0.29
recon	9	$+0.98$
$\|\epsilon\|^2$	7	$+1.47$
AWQ (Lin et al., 2024)	7	$+1.53$

G.3 GPT-2-MEDIUM: WHERE THE PER-LAYER OBJECT ITSELF FAILS

The GPT-2-medium column is here rather than in Table G.6 because the per-layer object itself fails here, not our estimate of it: *direct measurement of* a_ℓ also lands at or below the free heuristic, the reading §7 gives for $L/2$. The corrected score does lose on Pythia-1.4B and Qwen3-8B as well (Tables G.6, G.2).

Three readings matter. First, the corrected score trails deepest-k by 0.38 and 0.56 nats, inside the 0.6-nat bound quoted throughout §G.1, which is why that bound is 0.6 and not smaller. Second, *direct measurement of* a_ℓ *trails deepest-*k *too*, by $+0.26$ and $+0.59$: not a failure of the isotropic

Table G.4: **Scale check: Llama-3-8B, deletion and int4 at $k = 8, 16$.** ΔNLL vs deepest-k over the same one-shot grid as Table G.3, negative better, single seed. The curvature profile's transfer prediction holds at this scale ($\rho = +0.67$ to $+0.78$ for all five operations, against -0.22 to -0.39 for the uncorrected $\|\epsilon\|^2$), the free depth prior collapses (it loses all four cells), and the two weaker baselines average *worse* than deepest-k; Block Influence keeps pace here, which the 28-cell grid shows without changing the 24-cell table above. Entries are relative to deepest-k; the absolute figures here are anchored on deepest-k's mean token NLL at `prune`, k=16, which `e2e_llama38b_prune_k16.json` records alongside each arm.

selector	prune k=8	prune k=16	int4 k=8	int4 k=16
direct measurement of a_ℓ	-0.80	-1.03	-0.005	-0.003
$\frac{1}{2}\|\epsilon\|^2\kappa^{\text{iso}}$ (**ours**)	$\mathbf{-0.77}$	$+0.26$	$\mathbf{-0.005}$	$+0.003$
free depth prior	$+3.47$	$+2.47$	$+0.013$	$+0.013$
$\|\epsilon\|^2$	$+4.34$	$+2.37$	$+0.017$	$+0.022$
AWQ (Lin et al., 2024)	$+3.63$	$+2.85$	$+0.017$	$+0.033$
Block Influence (Men et al., 2025)	-0.64	-0.72	-0.004	-0.001
recon	$+2.31$	$+2.08$	$+0.014$	$+0.031$

Table G.5: **Layer deletion on GPT-2-medium.** ΔNLL relative to deepest-k, negative better, same protocol and tiers as Table G.6. Every per-layer selector lands at or below the free heuristic, direct measurement of a_ℓ included; the search tier alone beats it here. $\|\epsilon\|^2$ has no entry: it degenerates on this model.

selector	tier	k=6	k=12
deepest-k	free	0.00	0.00
free depth prior	free	-0.13	-0.64
$\|\epsilon\|^2$	per-layer	—	—
Taylor (Kim et al., 2024)	per-layer	-0.22	-0.20
$\frac{1}{2}\|\epsilon\|^2\kappa^{\text{iso}}$ (**ours**)	per-layer	$+0.38$	$+0.56$
direct measurement of a_ℓ	search	$+0.26$	$+0.59$
SLEB (Song et al., 2024)	search	-0.32	-1.06

approximation but a setting where per-layer damage, however exactly measured, is the wrong object, because interactions between deleted layers dominate, a per-layer score cannot be blamed for missing what a per-layer measurement also misses. Third, the profile §4 already flags this checkpoint: its layer 0 measures $\kappa^{\text{iso}} < 0$ (Qwen3-8B's layers 18–20 are the grid's only other negative readings, at 10^{-9}), and GPT-2-medium alone contributes more than one of the five non-improving cells in Table E.7, the diagnostic naming it before any selection is run, the behaviour §I.1 claims for it.

G.4 OLMoE-1B-7B: one-shot selection on a mixture-of-experts

The six ΔNLL cells of §G.1's mixture-of-experts paragraph come from `35_select_other_subs.py` with `--substitutions-file results/substitutions_olmoe.json --substitutions prune quant3 lowrank --ks 4 8 --out selectsubs_olmoe.json`: deletion, int3 and low-rank at $k = 4$ ($L/4$) and $k = 8$ ($L/2$), priced against deepest-k as in Table G.6, through the script producing the `selectsubs_*.json` files of Appendix G.2. They are measured selection damage (joint evaluation), not sums of single-layer a_ℓ. Read-only sanity check: summing the released per-layer prune damages `substitutions_olmoe.json` over the released selections `selection_by_profile_olmoe.json` reproduces the paragraph's ordering (corrected score -0.11, -0.09. $\|\epsilon\|^2$ $+2.61$, $+2.53$. Free prior $+0.00$, $+0.03$ nats against deepest-k), attenuated as expected for an additive proxy.

G.5 Layer deletion against published selectors, priced

The per-cell table and its cost tiers are here, the cost column in §G.1's pass units.

G.6 Cross-corpus selection

§G.1 reports the cross-corpus selection check: joint one-shot evaluations on held-out C4, each selector's chosen $k = L/4$ layers applied together and the NLL measured, with no alignment and no interaction model, as in the WikiText selection table (Table G.2).
The complete per-cell grid is in the supplementary appendix (Table S1), with the released JSON/CSV.

Table G.6: **Layer deletion against published selectors, priced.** ΔNLL relative to deepest-k, negative better. Cost is per compression *operation*, in forward passes over the calibration batch, and the tiers are the point: among selectors scoring layers independently the corrected score has the smallest worst case against deepest-k, its two positive entries both Qwen3-8B, which trails deepest-k at *both* budgets, k=9=$L/4$ and k=18=$L/2$ (Appendix G.8); off it the binding case is GPT-2-medium at $+0.56$ (Appendix G.3). The search tier re-pays its cost per operation; the profile's $2L$ probes are paid once.

			Qwen2.5-7B		Qwen2.5-0.5B		Qwen3-8B	
selector	tier	cost / op.	k=7	k=14	k=6	k=12	k=9	k=18
deepest-k	free	0	0.00	0.00	0.00	0.00	0.00	0.00
free depth prior	free	0	−1.33	−2.74	+1.06	−0.54	+4.93	+5.10
$\|\epsilon\|^2$	per-layer	L fwd	+5.85	+3.63	+5.80	+3.98	+5.08	+5.12
Taylor (Kim et al., 2024)	per-layer	L fwd+bwd	+5.79	+3.59	−1.17	−1.50	—	—
$\frac{1}{2}\|\epsilon\|^2\kappa^{\mathrm{iso}}$ (**ours**)	per-layer	L fwd	**−1.33**	**−1.67**	**−1.30**	**−2.48**	**+0.98**	**+3.64**
direct measurement of a_ℓ	search	L evals	−0.88	−2.75	−1.49	−2.48	+0.65	+2.34
SLEB (Song et al., 2024)	search	129–301 evals	−2.83	−4.14	−1.58	−2.68	—	—

Table G.7: Joint one-shot selection on held-out C4, $k = L/4$. Entries are ΔNLL relative to deepest-k, negative better. The corrected selector beats deepest-k in six of fifteen cells and ties five; `quant3` transfers in four of five models, `prune` and `lowrank` only on Qwen2.5-0.5B. Rows shown: the Pythia-410M pruning and `quant3` cells and the Qwen2.5-0.5B column; the remaining cells are in the supplementary appendix.

model	operation	corrected	$\|\epsilon\|^2$	depth prior
Pythia-410M	`prune`	+0.271	+1.611	+0.216
Pythia-410M	`quant3`	−1.116	−1.123	−1.074
Qwen2.5-0.5B	`prune`	−1.195	+1.558	+0.934
Qwen2.5-0.5B	`quant3`	−0.139	+0.708	+0.423
Qwen2.5-0.5B	`lowrank`	−0.305	+1.174	−0.307

G.7 Does the per-layer object compose into a set?

Table E.20 answers this for the sampled k=3 landscape and in the form the paper uses: the subset chosen by *measured* per-layer damage is the best measured subset, rank 1 of 220 and of 302, which is what licenses a per-layer rule at all. What that table does not report is how closely the additive object tracks the joint damage it is standing in for, or whether it does so under the flagship operation. This section measures both, and separates the two questions Table E.20 keeps together: whether the per-layer *form* is sound, and whether the *score* is accurate enough to exploit it. The form is what this section tests, using measured a_ℓ; the score is tested in Table E.20 and in §G.2.

Summing per-layer damage over a set is a valid way to price that set only if the layers do not interact, so we measure the interaction directly: sum a_ℓ over a set, and compare against the damage of removing that set jointly. Table G.8 collects every setting where both quantities are measured. Three are the released oracle files, which enumerate or sample layer sets under `linear`; the fourth is measured here for the deployment setting, Llama-3-8B under `prune` at k=8 of 32 (k=$L/4$), where the per-layer vector and 302 joint evaluations are taken on one stream at one dtype (`results/subset_additivity_llama38b.json`, `results/subset_additivity.py`). The released per-layer and end-to-end files sit at baselines of 2.0107 and 0.695 nats and were deliberately not mixed.

Table G.8: **The additive per-layer sum against the measured joint damage.** ρ is Spearman over the listed subsets, between $\sum_{\ell\in S} a_\ell$ and the damage of applying the operation to every layer in S at once. "Argmin" records whether the set with the smallest sum is also the set with the smallest measured damage. The first row enumerates every 3-subset of 12 layers; the others sample. Recomputed from `results/oracle_*.json` and `results/subset_additivity_llama38b.json`.

checkpoint	operation	n	k	subsets	ρ
Pythia-160M	`linear`	12	3	220 (all)	+0.988
Pythia-160M	`linear`	12	6	403	+0.928
Pythia-410M	`linear`	24	3	302	+0.972
Llama-3-8B	`prune`	32	8	302	+0.907

The additive sum orders the joint damage closely in every setting, and in all four the set with the smallest sum is the set with the smallest measured damage. On the 220 exhaustive 3-subsets of Pythia-160M the two orderings agree at the top exactly. That is the property a per-layer rule spends, and it holds where it is spent, under `prune` as well as under `linear`.
Table G.9 prices the arms of Table 1 the same way. The set the score chose is not among the 302 and beats all of them: the best of the 300 random draws (0.223) is 8.4% worse than it (0.204). Deepest-k measures 4.6 times the score's choice jointly, and the magnitude rule's eight 22.5 times.

Table G.9: **Four sets on Llama-3-8B under `prune` at k=8 of 32.** Left is the sum of the measured per-layer damages; right is the measured damage of removing all eight layers at once, on the same stream. The ratio is the two; it is below one everywhere, so the sum orders sets without pricing them. The three named sets are the arms of Table 1; "all" removes every attention sublayer. From `results/one_shot_sets.json`.

set	$\sum_{\ell\in S} a_\ell$	joint a	ratio
CurvProbe's eight	0.153	**0.204**	0.75
deepest-k	0.317	0.948	0.33
magnitude rule's eight	2.745	4.590	0.60
all 32	3.707	6.418	0.58

Two limits belong with this. The sum is an orderer, not a price: over the 302 sets its median ratio to the measured joint damage is 0.55, and the discrepancy grows with how much of the model is removed, from 0.75 on the selector's eight to 0.58 on all thirty-two. A paper that quoted $\sum a_\ell$ as the cost of a set would be understating it by roughly a factor of two. What survives is the ordering, which is what the selector uses.
The second limit is a negative one, and it closes an explanation §7 offers. The profile's margin over deepest-k reverses at k=$L/2$, and the natural reading is that layers stop being independent once half the model is gone. They do not: the Pythia-160M row at k=6 of 12 is k=$L/2$, and its ρ is +0.928 with the optimal set still found exactly. Whatever reverses the margin at $L/2$, it is not set interaction, which leaves the scoring rather than the object as the place to look.

G.8 Linearisation selection, per model

Table G.11 gives the absolute ΔNLL against the clean model for all seven selectors, three seeds per model except GPT-J-6B (one), train and evaluation on disjoint corpus halves. Pythia-160M's 12 layers make $k = 12$ degenerate and it is excluded from all counts. Qwen2.5-7B is run at the two deployment budgets rather than swept (k=7 rows in Table G.10, the k=14 margin the 3.16 nats of §G.1).

Table G.10: **The retired linearisation cell.** Qwen2.5-7B at k=7 of 28, 300 alignment steps, ΔNLL on WikiText-103 at 1024 tokens, mean of three seeds (within-run sd in parentheses). Against the ±0.4-nat alignment-reproducibility band (Appendix D.4) it supports only the separation to $\|\epsilon\|^2$ alone; the first four rows' ordering is withdrawn. Retained because §G.1 counts it among the 44 margins the band absorbs.

selector	layers converted	ΔNLL
direct a_ℓ (L evals)	4, 12, 13, 14, 15, 16, 17	0.145 (0.005)
depth prior (free)	1, 2, 4, 5, 15, 16, 17	0.191 (0.026)
CurvProbe $\frac{1}{2}\|\epsilon\|^2\kappa^{\text{iso}}$	1, 2, 4, 5, 15, 16, 17	0.195 (0.016)
deepest-k	21–27	0.355 (0.009)
$\|\epsilon\|^2$ alone	0–6	3.666 (0.141)

G.9 Search restricted to $2k$ candidate layers

The complete per-cell grid is in the supplementary appendix (Table S3), with the released JSON/CSV.
The restricted search, run where per-layer sets lose. On Qwen2.5-7B, Gemma-2-9B and Bloom-7B1 every per-layer set trails a set chosen jointly, and the reason is measured: deleting the per-layer sets together costs 7 to 68 times the sum of their single-layer damages, direct measurement's included (Table G.14). Deletions compound. EXP118 (`prereg_seeded_greedy.md`, `118_seeded_greedy.py`) restricts a greedy search to the $2k$ layers with the lowest clean-output deletion score and adds, k times, the pool layer that minimises the joint NLL on the 32 calibration

Table G.11: Absolute ΔNLL after 300 alignment steps, lower better. Low budget: $k = 6$ of 24, 6 of 12 on Pythia-160M ($L/2$ there), 7 of 28 on Qwen2.5-7B and GPT-J-6B. High budget: $k = 12$ of 24, 14 of 28 on Qwen2.5-7B and GPT-J-6B; on Pythia-160M it would be 12 of 12, degenerate and excluded from all counts. Two columns support no claim: at k=7 on Qwen2.5-7B this selector picks the *same* layers as the free prior, retiring that cell (Table G.10), and the GPT-J-6B column predates the adapter correction of Appendix D.3.

selected by	160M	410M	1.4B	GPT-2-m	Qwen2.5-7B	GPT-J-6B
low budget						
a_ℓ, direct measurement	0.482	0.185	0.150	0.054	0.177	0.246
$\frac{1}{2}\|\epsilon\|^2\kappa^{\text{iso}}$ (ours)	0.394	0.254	0.155	0.095	0.162	0.313
$\frac{1}{2}\|\epsilon\|^2 e^{-3\ell/(L-1)}$, free	0.482	0.292	0.193	0.117	0.743	0.392
$\|\epsilon\|^2$	0.692	0.516	0.595	0.273	1.871	3.836
deepest-k	0.482	0.210	0.193	0.159	0.241	0.230
uniform spacing	0.670	0.309	0.292	0.066	0.358	0.437
shallowest-k	1.228	0.671	0.705	0.444	0.912	3.863
high budget						
a_ℓ, direct measurement	—	0.459	0.446	0.228	0.512	0.811
$\frac{1}{2}\|\epsilon\|^2\kappa^{\text{iso}}$ (ours)	—	0.941	0.515	0.587	0.744	3.696
$\frac{1}{2}\|\epsilon\|^2 e^{-3\ell/(L-1)}$, free	—	0.557	0.494	0.345	0.951	3.633
$\|\epsilon\|^2$	—	0.880	0.963	0.604	1.916	3.931
deepest-k	—	0.699	0.739	0.467	0.772	0.918
uniform spacing	—	0.571	0.563	0.205	1.681	1.124
shallowest-k	—	1.907	1.492	1.196	1.350	3.945

Table G.12: **Does the profile's pool contain what the search finds?** Per (checkpoint, k) with SLEB run: SLEB's chosen layers inside the profile's top-m pool, the hypergeometric null for a same-size random pool, and the greedy evaluations each route needs — reachability, not what a restricted greedy would return. One row per checkpoint: the pool holding the largest share of SLEB's chosen layers, ties broken toward the cheaper pool; the full pool-size sweep is in the supplementary appendix.

checkpoint	k	pool m	inside	null	evals (pool / full)
Pythia-410M	6	6	3/6	0.250	21/129
Pythia-1.4B	6	6	4/6	0.250	21/129
GPT-2-medium	12	18	9/12	0.750	150/222
Qwen2.5-0.5B	6	18	6/6	0.750	93/129
Qwen2.5-7B	7	21	7/7	0.750	126/175

Table G.13: **The same question where the answer is exact.** Pythia-160M, linearisation, k=3: all $\binom{12}{3} = 220$ triples measured, so a pool-restricted greedy can be replayed exactly rather than argued about. The top-$1.5k$ pool reaches the exhaustive optimum in 9 evaluations against the unrestricted greedy's 33; the profile alone lands on the second-best triple at zero evaluations.

route	layers chosen	rank	evaluations
oracle (exhaustive)	8, 9, 11	1	—
greedy, unrestricted	8, 9, 11	1	33
greedy, pool $1.5k$	8, 9, 11	1	9
greedy, pool $2k$	8, 9, 11	1	15
greedy, pool $3k$	8, 9, 11	1	24
greedy, pool k	9, 10, 11	2	6
profile alone, no search	9, 10, 11	2	0
deepest-k	9, 10, 11	2	0

sequences; the full greedy does the same over all L layers. Registered: G1, the restricted set at or below Block Influence on the held-out stream on every checkpoint; G2, within 0.05 nats of the full greedy; G3, at most half its evaluations. G1 and G3 held on all three. G2 held on Qwen2.5-7B (+0.029) and Bloom-7B1 (+0.019) and failed on Gemma-2-9B (+0.156), whose 42 layers leave a 20-layer pool less room.

Two departures from the registration change no verdict. The stream holds 121 to 127 sequences of 1024 tokens rather than the registered 200, and the 32 calibration sequences are drawn from the

Table G.14: **EXP118, held-out NLL after deleting each set**, bf16, the EXP74 stream with the checkpoint's own document-start convention; lower is better. "joint/sum" is the set's joint damage over the sum of its layers' single-layer damages. Bold: best among the sets that need no full search.

	Qwen2.5-7B, k=7		Gemma-2-9B, k=10		Bloom-7B1, k=8	
set	NLL	joint/sum	NLL	joint/sum	NLL	joint/sum
clean	2.159	—	2.247	—	2.637	—
restricted greedy (pool $2k$)	**2.390**	1.6	**2.531**	1.7	**2.761**	2.1
full greedy	2.359	1.4	2.378	2.1	2.742	1.7
Block Influence	3.027	6.5	2.881	2.4	3.792	9.3
deepest-k	5.348	2.6	3.608	4.7	3.355	7.2
direct measurement, per layer	4.395	18.2	3.020	20.4	3.214	15.8
directional read, per layer	4.004	14.2	3.999	10.5	5.728	67.7
isotropic profile, per layer	3.873	7.0	5.217	4.0	4.733	41.6
evaluations, restricted / full	77 / 175		155 / 375		100 / 212	

same stream, so they sit inside the held-out set. Re-scored on the 89 to 95 sequences that are not calibration, every difference in Table G.14 moves by at most 0.02 nats.

Four further checkpoints. EXP120 (`prereg_additivity_screen.md`, `120_additivity_screen.py`) scores the profile's per-layer deletion set on Mistral-7B-v0.3, Qwen3-8B, LLaMA-13B and OLMo-1-7B, none of which had a joint deletion evaluation, at $k{=}L/4$ and $k{=}L/2$, beside Block Influence, deepest-k and direct measurement (Table G.15). At $k{=}L/4$ the profile's set beats Block Influence on Mistral-7B and LLaMA-13B and is within 0.03 nats of it on Qwen3-8B and OLMo-1-7B; at $k{=}L/2$ it beats both Block Influence and deepest-k on Mistral-7B and LLaMA-13B. The greedy restricted to the profile's own $2k$ cheapest layers, run on the two $k{=}L/4$ sets the screen below flags, brings the damage to 0.19 and 0.28 nats against Block Influence's 0.90 and 1.00, at 126 and 100 evaluations (A2 confirmed). With EXP118, a search restricted to $2k$ candidate layers beats Block Influence on all five checkpoints it was run on; EXP121 below measures how much of that the pool itself contributes.

The additivity screen. The registration also put a routing rule at risk: the ratio R of a set's joint damage to the sum of its layers' single-layer damages, with $\tau = 3$ read off the 29 released joint evaluations, where every set with $R \geq 4$ had lost and 21 of the 23 with $R \leq 2$ had held. On the new checkpoints the low side held in all three settings, LLaMA-13B at $k{=}L/2$ among them. The high side flagged three sets that were already within 0.05 nats of the better of Block Influence and deepest-k (A1 refuted, 5 of 8 against ≥ 7): Mistral-7B at $k{=}L/2$, at $R = 3.9$, and the two $k{=}L/4$ sets on which the restricted search then made its largest gains. On Qwen3-8B six layers lower the loss when deleted alone, which pulls the denominator towards zero. A3, $R > \tau$ at $k{=}L/2$ on at least three of the four checkpoints, held.

Table G.15: **EXP120, held-out damage after deleting each frozen set** (nats above the clean model), bf16, the EXP74 stream, no document-start token. R is the profile set's joint damage over the sum of its layers' single-layer damages from the grid. The restricted greedy (evaluations in parentheses) runs only where $R > 3$ at $k{=}L/4$, as registered. Bold: best in the row; underline: second.

checkpoint	k	R	profile	Block Influence	deepest-k	direct	restricted greedy
Mistral-7B-v0.3	8	1.9	**0.157**	0.262	0.909	**0.157**	—
	16	3.9	1.112	1.467	2.088	**0.868**	—
Qwen3-8B	9	37.1	0.925	0.900	2.942	0.447	**0.186** (126)
	18	20.8	3.756	2.501	7.666	**2.401**	—
LLaMA-13B	10	1.8	0.192	0.229	0.379	**0.075**	—
	20	2.7	0.547	0.791	0.870	**0.453**	—
OLMo-1-7B	8	5.3	1.016	0.998	6.955	0.829	**0.284** (100)
	16	14.7	6.142	**5.649**	8.321	6.517	—

What the pool contributes. Any pool of $2k$ layers gives the evaluation saving, so EXP121 (`prereg_pool_control.md`, `121_pool_control.py`) asks whether the read's pool also leads the search to a better set. On Qwen2.5-7B, Qwen3-8B and OLMo-1-7B it reruns the same greedy from the original pool, from Block Influence's $2k$ cheapest layers, from the deepest $2k$ and from three random pools of the same size (Table G.16). The original pool reproduces its earlier set exactly on all three. The joint evaluation does the work: every pool's greedy beats Block Influ-

ence's per-layer set, by 0.11 to 0.75 nats. The read's pool is within 0.05 nats of Block Influence's pool on all three (C1 confirmed), but it is clearly ahead of random pools only on OLMo-1-7B (C2 refuted, 1 of 3; C3 refuted, 0 of 3), so the restricted search's gain belongs to restricting the search rather than to which read chose the pool. Gemma-2-9B was registered but not run: no box held the checkpoint the grid measured, and a mirrored copy missed the grid's clean NLL by 0.13 nats (`prereg_pool_control_addendum.md`).

Table G.16: **EXP121: the same greedy from pools of the same size.** Held-out NLL of the set each pool's greedy chooses ($k{=}L/4$, pool $2k$, bf16, the EXP74 stream). "read" is the pool of the original run: the clean-output read on Qwen2.5-7B, the profile on the other two. Random: mean of three pools, range in brackets. The last column is Block Influence's per-layer set, with no search. Bold: best pool in the row, underline: second, random pools taken at their mean.

checkpoint	clean	read	profile	Block Influence	deepest	random	BI per-layer
Qwen2.5-7B	2.159	<u>2.390</u>	2.423	**2.386**	2.468	2.401 [2.374, 2.418]	3.027
Qwen3-8B	2.512	2.698	—	**2.657**	2.704	<u>2.695</u> [2.662, 2.755]	3.412
OLMo-1-7B	2.296	<u>2.580</u>	—	**2.561**	3.184	2.783 [2.680, 2.946]	3.295

G.10 The same audit on deepest-k and Block Influence

§I.1 audits an exponential prior. Whether $\hat{c}$ audits *deepest-k*, which deletes the last k layers, and *Block Influence* (Men et al., 2025), the strongest published selector on our grid, is answered from the released grids without a new measurement: deepest-k is the layer index and Block Influence is stored per layer in `substitutions_*.json` (`analysis/three_selector_audit.py`).

As selectors, both lose to the corrected score. Over the seventeen audited checkpoints, mean Spearman against measured damage across the five operations, the corrected score wins by more than the 0.05 margin on $\mathbf{16/17}$ against deepest-k and $\mathbf{13/16}$ against Block Influence (absent on OPT-6.7B), against $13/17$ for the exponential prior. The only two cells going the other way are the two largest RMSNorm checkpoints, where Block Influence edges it: Mistral-7B (-0.005), Llama-3-8B (-0.012).

The gate does not transfer, and the reason is structural. $\hat{c} > 3$ or $\hat{c} < 0$ classifies the exponential prior at $12/15$ on the checkpoints where it returns a call, but scores $7/15$ on deepest-k and $7/15$ on Block Influence, both worse than a constant unsafe vote ($14/15$, $12/15$), not a tuning failure, since the gate thresholds a *rate* and only a rule that commits to one can be wrong about it. Deepest-k fixes an ordering, so every $c > 0$ induces the same ranking and only the sign could condemn it. Block Influence is measured with no depth assumption, so $\hat{c}$ carries no information about it either. This is the boundary of §I.1: it prices rules of the form $\|\epsilon_\ell\|^2 e^{-c\ell/(L-1)}$, and a rule that leaves the rate implicit has to be priced by measuring damage.

G.11 The spot-check gate

The fidelity, single-draw error and leave-one-out tables are in the supplementary material (`spotcheck_gate*.json`). *this section states what they return.*

A practitioner needs to know, before anything expensive, whether the profile can be trusted here: the signal is the profile's own rank correlation against damage, estimated from m random layers rather than all L. Over the twelve pairs of Table G.2 it tracks the selection advantage at Spearman -0.55, and the ranking is more useful than the correlation, since the pair where the corrected score is furthest behind, `prune` on Pythia-1.4B, has the *lowest* ρ of all twelve ($+0.43$ against $+0.51$–$+0.97$): a threshold in $(0.43, 0.51)$ routes exactly that pair to direct measurement and every other pair to the profile, with no errors in either direction. But the interval is read off these twelve pairs, not calibrated on held-out models, so it describes them, and a new checkpoint's estimate can fall inside it without being the pair this gate was built to catch: the gate is a screen, not a certificate.

Estimating ρ is cheap: four random layers already recover the model ordering at Spearman $+0.98$ against the full-grid values ($+1.00$ from six), and at $m = 8$, a third of the price of measuring every layer, a single draw wrongly trusts the profile 5–9% of the time, falling to 2–4% at $m = 12$. The gate buys a failure detectable before a conversion is run rather than after one.

G.12 The name-baseline re-allocation

The protocol, the per-cell grids, the registered prediction files and the figure of the causal chain are in the supplementary material (`84_owl_fix_table.py`,

86_owl_prediction_grid.py, 87_waterfill.py. owl_sweep_*.json, owl_grid_summary.json, owl_pred_locked.json, owl_wf_*.json). *This section states what they return.*

The profile is applied here to a different question from the rest of the paper: not which layers to compress, but how much sparsity to give each one. OWL (Yin et al., 2024) allocates that budget from a layer-outlier statistic D^ℓ. We substitute the corrected score and hold OWL's clip, its mean sparsity and its pruning executor fixed, so every arm is read as a degradation ratio under one protocol.

The ranking transfers. The budget does not. At equal budget the corrected score ranks per-layer damage better than D^ℓ on eleven of twelve checkpoints, and inverts nowhere the outlier statistic does not. The budget built from that ranking wins twenty-one of thirty cells, while the sign of the ranking gap — readable before compressing — calls only twenty of them, no better than answering *ours* every time. Ordering single-layer damage better does not make a score a better allocator, and here it does not. The two objects differ in kind: a per-layer ranking prices a fixed perturbation at one layer, whereas a non-uniform budget needs the marginal damage curve $\partial\mathcal{L}/\partial s_\ell$ and the cross-layer terms, so a ranking that is exact at one perturbation size does not define the optimal allocation.

Two registered predictions, both refuted. Before the six held-out checkpoints were measured we locked a joint rule on quantities readable without compressing: a cell gains over uniform if and only if uniform's degradation lies in $[2.5\times, 25\times]$ *and* the checkpoint's $|R_3|/a$ is at most 0.5. The envelope is refuted at both edges (4 of 13 held-out cells, below chance ($p = 0.95$); 3 of 13 once Bloom-7B1 is re-measured on the fixed harness, $p = 0.99$) and an earlier registered prediction was refuted as well. Both are carried in the ledger of Appendix A.1 with the outcomes above.

Water-filling the budget instead of clipping it. Registered separately: the primary criterion holds, 4 of 5 cells against the linear map. The secondary does not, 1 of 5.

H Deployment

H.1 End-to-end deployment test: the magnitude rule's disaster

§5 ranks layers by correlation. A deployment asks whether the ranking survives when the chosen layers are deleted and the model re-evaluated. We run the one-shot attention-sublayer deletion of §6 end to end on Llama-3-8B (k=8 of 32, bf16, one-shot): each arm's layers are removed and mean token NLL is measured on the same evaluation stream (EXP74). Table H.1 reports the four arms.

The profile's eight layers hold mean token NLL at 2.21. The magnitude rule's eight cost 7.39, closer to deleting everything (8.40) than to the profile, and the per-layer correlation the same run measured ($\rho = +0.72$ to $+0.85$ on the three calibration segments) is what realises the gap. This is the application the scoreboard's negative correlations pointed at: on a production model the field's default magnitude rule chooses almost the worst possible layers.

The ranking is stable across seeds, not one draw. The per-cell numbers in §5 are each a single evaluation. To show that the profile's advantage is not one lucky sample, the whole grid was re-run under three further seeds and the profile's dNLL against deepest-k recorded (Table H.2). Across the five checkpoints the spread is at most 0.22 nats, on four of them below 0.05, and on GPT-J-6B exactly zero.

H.2 Selection under real GPTQ and AWQ

Appendix A.2(B) asks whether a real quantiser's perturbation still points the way round-to-nearest does, a mechanism question. The deployment question is whether the choice still pays after scoring with the cheap proxy and quantising the chosen layers with the real algorithm. The asymmetry is deliberate: the *score* comes from RTN, because a practitioner will not run GPTQ L times to decide where to run GPTQ, and the *deployed* operation is the real quantiser, so this measures proxy transfer. Table H.3 gives the twelve cells.

The registered rule, frozen before the run: the corrected score stays at or better than deepest-k in at least as many cells as under the RTN control, and never breaks the 0.6-nat guarantee of §G.1. Both hold: $2/2$ cells under each real quantiser on each model, no cell within 0.6 nats of breaking.

The table cuts against a triumphant reading. On Qwen2.5-0.5B the RTN control is the only setting where selection is worth much: $\|\epsilon\|^2$ is catastrophic there ($+0.898$, $+2.042$ nats worse than deepest-k) and the correction rescues it. Under GPTQ and AWQ every arm collapses into a band of ≤ 0.14 nats, the free depth prior inside it. On Pythia-1.4B the arms are indistinguishable under all three quantisers, spread < 0.05 nats. On Qwen2.5-7B the ordering is unstable, $\|\epsilon\|^2$ and the corrected score swapping places between budgets under both quantisers, each swap < 0.011 nats, not a ranking.

Table H.1: **The magnitude rule is a disaster, not a mismeasurement.** Llama-3-8B, one-shot deletion of $k{=}8$ attention sublayers, bf16, mean token NLL (nats) on the EXP74 evaluation stream. Clean NLL is 2.0037. "magnitude" selects the eight layers the magnitude rule considers cheapest; "all" deletes every layer as a floor. The magnitude rule lands 1.01 nats below deleting all thirty-two layers.

arm	layers deleted	NLL
clean	—	2.00
CurvProbe (profile)	11, 18–20, 22–25	**2.21**
magnitude ($\|\epsilon_\ell\|^2$)	0–6, 19	7.39
deepest-k	24–31	2.98
all layers	0–31	8.40

Table H.2: **Seed robustness of the profile's selection advantage (attention-sublayer deletion, prune).** For each checkpoint, the mean per-cell ΔNLL of the profile's chosen layers against deepest-k, under three seeds (the base run plus seed-17 and seed-42; seed-1234 was also run, and is reported in the scoreboard of Appendix G.2). A small spread means the profile's advantage is a property of the checkpoint, not of one batch draw.

checkpoint	seed base	seed-17	seed-42	spread
Qwen2.5-0.5B	-1.304	-1.298	-1.314	0.02
Qwen2.5-7B	-1.328	-1.482	-1.547	0.22
GPT-J-6B	0.000	0.000	0.000	0.00
Pythia-1.4B	$+0.083$	$+0.087$	$+0.094$	0.01
Llama-3-8B	-0.766	-0.747	-0.781	0.03

Table H.3: **Selection under real quantisers, int3.** ΔNLL relative to deepest-k, negative better. Scores come from the RTN proxy throughout, only the deployed quantiser changes; `rtn` is the control arm, reproducing fake-quantisation on this corpus, so a difference under `gptq` or `awq` is attributable to the quantiser, not the harness. Only the selected layers are quantised, each calibrated on the clean model: not a sequential GPTQ pipeline and not a GPTQ reproduction. Budgets are $L/4$ and $L/2$ on every checkpoint ($k \in \{6, 12\}$ on the 24-layer models, $k \in \{7, 14\}$ on the 28-layer Qwen2.5-7B), matching the selection grid of Appendix G.2.

model	quantiser	k	$\frac{1}{2}\|\epsilon\|^2\kappa^{\mathrm{iso}}$	$\|\epsilon\|^2$	depth prior	direct a_ℓ
Qwen2.5-0.5B	`rtn` (control)	6	-0.153	$+0.898$	$+0.797$	-0.177
		12	-0.208	$+2.042$	$+1.141$	-0.232
	`gptq`	6	-0.006	-0.004	-0.038	-0.034
		12	-0.026	$+0.001$	-0.041	-0.015
	`awq`	6	-0.039	-0.002	-0.050	-0.063
		12	-0.051	$+0.050$	-0.041	-0.086
Pythia-1.4B	`rtn` (control)	6	-0.412	-0.413	-0.396	-0.416
		12	-1.050	-1.050	-1.064	-1.087
	`gptq`	6	-0.375	-0.373	-0.367	-0.371
		12	-1.061	-1.061	-1.050	-1.066
	`awq`	6	-0.433	-0.431	-0.408	-0.433
		12	-1.123	-1.123	-1.120	-1.151
Qwen2.5-7B	`rtn` (control)	7	-0.074	-0.065	-0.076	-0.095
		14	-0.123	-0.114	-0.111	-0.137
	`gptq`	7	-0.025	-0.024	-0.025	-0.022
		14	-0.038	-0.039	-0.035	-0.041
	`awq`	7	-0.054	-0.043	-0.052	-0.065
		14	-0.076	-0.077	-0.070	-0.090

The absolute numbers explain it: at the low budget the corrected score's own damage under GPTQ against RTN is $+0.072$ vs $+0.147$ nats on Qwen2.5-0.5B, $+0.022$ vs $+0.065$ on Pythia-1.4B and $+0.007$ vs $+0.051$ on Qwen2.5-7B, two to seven times less damage from the better quantiser, and a smaller perturbation leaves less for any selector to recover. This is §I.1's prediction, not an exception to it: the correction pays where there is damage to rank and is inert where there is not, so the practical statement is narrower than "the profile helps you quantise": at int3 with a good quantiser, where you quantise matters little, and the profile's value is that it does not hurt and flags this case in advance.

H.3 Selection under real SparseGPT-style 2:4 pruning

The pruning analogue of the quantiser test. Our `sparse24` is a magnitude mask, zeroing the two smallest-magnitude weights in each group of four with no reconstruction. The literature's 2:4 is error-compensated, SparseGPT (Frantar & Alistarh, 2023) solving the same layer-wise reconstruction as GPTQ with a 2:4 mask in place of the quantiser. The deployment question: does the choice still pay

after scoring with the cheap naive-2:4 proxy and pruning the chosen layers with the error-compensated algorithm?

The asymmetry is deliberate, as in the quantiser test: the *score* comes from the naive-2:4 proxy, because a practitioner will not run SparseGPT L times to decide where to run SparseGPT, and the *deployed* operation is the error-compensated pruner. Only the selected layers are pruned, each calibrated on the clean model: not a sequential SparseGPT pipeline and not a SparseGPT reproduction. Table H.4 gives the sixteen cells.

Table H.4: **Selection under an error-compensated 2:4 pruner.** ΔNLL relative to deepest-k, negative better. Scores come from the naive-2:4 proxy throughout, only the deployed pruner changes; `sparse24` is the control, reproducing the proxy on this corpus, so a difference under `sparsegpt` is attributable to reconstruction, not the harness. Budgets are $L/4$ and $L/2$ everywhere: $k \in \{3, 6\}$ (12-layer Pythia-160M), $k \in \{5, 11\}$ (22-layer TinyLlama-1.1B), $k \in \{6, 12\}$ (24-layer checkpoints), $k \in \{7, 14\}$ (28-layer GPT-J-6B, Qwen2.5-7B), matching the selection grid throughout.

model	pruner	k	$\frac{1}{2}\|\epsilon\|^2\kappa^{\mathrm{iso}}$	$\|\epsilon\|^2$	depth prior	direct a_ℓ
Pythia-160M	`sparse24` (control)	3	$+0.447$	$+0.447$	$+0.000$	-0.021
		6	$+0.353$	$+0.889$	$+0.017$	-0.245
	`sparsegpt`	3	-0.025	-0.025	$+0.000$	-0.050
		6	-0.278	$+0.335$	-0.313	-0.234
Pythia-410M	`sparse24` (control)	6	-0.781	-0.661	-0.785	-0.815
		12	-1.099	-1.001	-1.424	-1.329
	`sparsegpt`	6	-0.840	-0.834	-0.820	-0.837
		12	-1.189	-1.212	-1.102	-1.189
Pythia-1.4B	`sparse24` (control)	6	-0.164	-0.013	-0.194	-0.191
		12	-0.259	-0.055	-0.368	-0.465
	`sparsegpt`	6	-0.106	-0.105	-0.103	-0.105
		12	-0.354	-0.352	-0.340	-0.365
GPT-2-m	`sparse24` (control)	6	-0.057	-0.036	-0.020	-0.040
		12	-0.051	$+0.025$	-0.041	-0.099
	`sparsegpt`	6	-0.014	-0.013	$+0.010$	-0.011
		12	-0.025	-0.030	$+0.001$	-0.048
TinyLlama-1.1B	`sparse24` (control)	5	-0.072	-0.017	-0.034	-0.077
		11	-0.069	$+0.002$	$+0.002$	-0.144
	`sparsegpt`	5	-0.011	-0.007	-0.011	-0.013
		11	-0.024	-0.020	-0.020	-0.028
GPT-J-6B	`sparse24` (control)	7	$+0.002$	$+0.200$	$+0.000$	-0.011
		14	$+0.001$	$+0.302$	-0.011	-0.041
	`sparsegpt`	7	-0.001	$+0.010$	$+0.000$	-0.001
		14	-0.008	$+0.011$	-0.005	-0.006
Qwen2.5-0.5B	`sparse24` (control)	6	-0.389	$+0.224$	-0.198	-0.387
		12	-0.427	$+0.741$	-0.072	-0.472
	`sparsegpt`	6	-0.008	$+0.006$	-0.002	-0.007
		12	-0.005	$+0.008$	-0.001	-0.006
Qwen2.5-7B	`sparse24` (control)	7	-0.236	-0.234	-0.235	-0.263
		14	-0.269	-0.268	-0.268	-0.291
	`sparsegpt`	7	-0.003	-0.005	-0.003	-0.004
		14	-0.013	-0.014	-0.008	-0.015

The registered rule, frozen before the run, was the pruning analogue of the quantiser rule: the corrected score stays at or better than deepest-k in at least as many cells under the error-compensated pruner as under the naive-2:4 control, and never breaks the 0.6-nat guarantee. Both hold: $2/2$ cells per model under `sparsegpt`, sixteen in total, no cell of the thirty-two worse than deepest-k by more than 0.6 nats, and the largest miss anywhere $+0.447$ nats, on Pythia-160M under the naive mask.

The exception is instructive, cutting as the quantiser table did: under the naive mask the corrected score trails deepest-k in both Pythia-160M cells ($+0.447$, $+0.353$), and reconstruction rescues both (-0.025, -0.278). Elsewhere the arms look alike, the four-selector spread narrowing from ≤ 1.2 nats under the mask to ≤ 0.11 nats under `sparsegpt`, the inertness §F.1 predicts for an error-compensated perturbation, repeating the quantiser finding of Appendix H.2. Surviving gaps still favour the corrected score: on Pythia-410M 0.8–1.2 nats ahead of deepest-k, on Pythia-1.4B 0.1–0.4.

H.4 End-to-end quantisation

Per-channel int4 quantisation on the $k = L/4$ layers each arm selects (ours = curvature-corrected score, uniform = all layers. The complete grid, the deepest-k, `gptq-sel` and `awq-sel` rows included, is Table S5 of the supplementary appendix): perplexity on the held-out half of the validation stream, four downstream tasks at 500 samples each (ARC-Easy uses its whole test set of 479), all

five checkpoints completing all arms, model names abbreviating the checkpoints of §3. Figure H.1 gives the same grid as a heat map.

Table H.5: Full end-to-end table for §6. Loss is mean token NLL (nats).

model	arm	NLL	LAMBADA	ARC-Easy	HellaSwag	MMLU
Qwen2.5-7B	ours	**2.163**	**0.666**	0.637	**0.580**	**0.414**
Qwen2.5-7B	uniform	2.195	0.644	0.630	0.574	0.398
Qwen3-8B	ours	2.520	0.622	0.645	0.526	**0.416**
Qwen3-8B	uniform	2.575	0.592	0.645	0.516	0.410
OLMoE-1B-7B	ours	2.281	**0.664**	**0.649**	**0.570**	0.378
OLMoE-1B-7B	uniform	2.325	0.630	0.635	0.554	0.384
Llama-3-8B	ours	**2.012**	0.674	0.666	0.590	0.404
Llama-3-8B	uniform	2.092	0.662	0.645	0.580	0.396
GPT-J-6B	ours	2.375	0.602	0.593	0.520	0.340
GPT-J-6B	uniform	2.408	0.600	0.589	0.496	0.344

The complete per-cell grid is in the supplementary appendix (Table S5), with the released JSON/CSV.

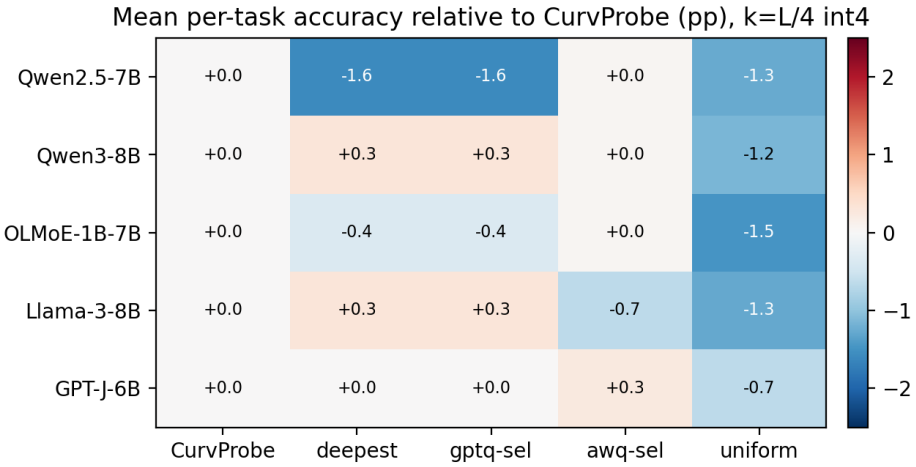


Figure H.1: End-to-end summary: mean per-task accuracy relative to CurvProbe (points) at $k = L/4$ per-channel int4. Blue is loss, red is gain, CurvProbe the reference column. Uniform quantisation is negative on every checkpoint. AWQ-style salience is the strongest baseline, tying within 0.7 points with a single visible gap on Llama-3-8B.

H.5 The distillation dissociation

Figure H.2 plots the three objectives of Table H.6: the one that shrinks $\|\epsilon_\ell\|$ most damages the model most — §4, shown directly.

Table H.6: Pythia-410M, identical data and seeds, best of three learning rates per objective at 300 steps: the objective that shrinks $\|\epsilon\|$ most is the worst by ΔNLL at both k, and at 3000 steps the standard objective is still worst, with the gap at 1.65×.

	$k = 6, 300$		$k = 12, 300$		$k = 12, 3000$
objective	$\|\epsilon\|$ red.	ΔNLL	$\|\epsilon\|$ red.	ΔNLL	ΔNLL
attn-output MSE (standard)	**61.4%**	+0.188	**41.5%**	**+4.876**	+0.743
final-hidden MSE	52.5%	+0.142	26.3%	+0.602	+0.452
hidden + logit	39.4%	**+0.115**	16.7%	**+0.461**	**+0.451**

H.6 Downstream tasks

Two operating points, and why only one is in the main text. Deployment applies deletion one-shot, so the task evaluation and the ΔNLL selection of Table G.6 (the deletion grid printed below, Table H.7) share that operating point; linearisation differs — our ΔNLL results follow 300

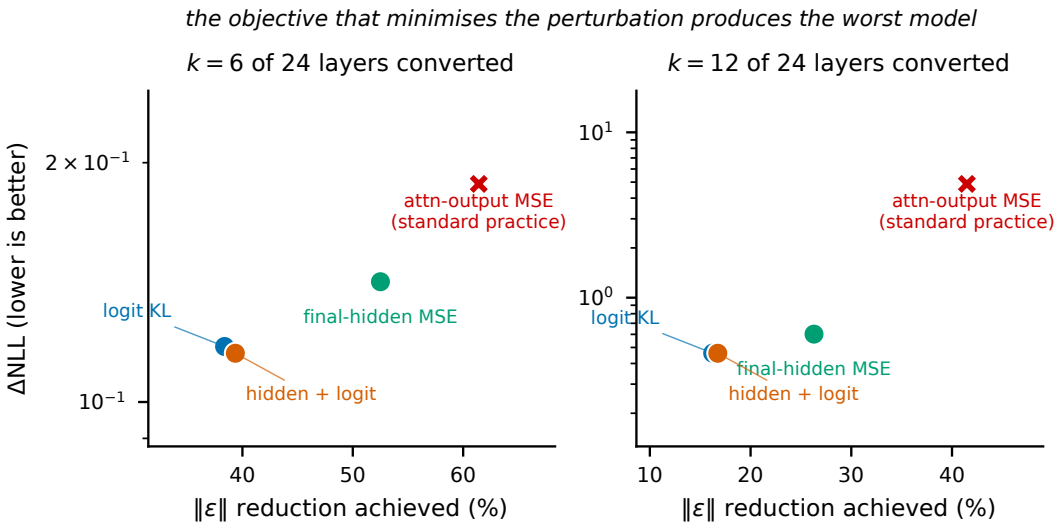


Figure H.2: Distilling a converted layer against its own output minimises $\|\epsilon\|$ better than any other objective and produces the worst model: §4 shown directly, the quantity local salience measures not being the quantity that determines damage.

alignment steps, the task evaluation below is one-shot — and at $k = 12$ alignment, not selection, decides the comparison: unaligned, the curvature-selected subset trails deepest-k on all three models ($+0.76$, $+1.61$, $+0.57$ nats), aligned it leads on two (-0.22, -0.03). The linearisation rows behind this paragraph are what we measured, not evidence about the selector.

Why these tasks. All models are base checkpoints except the two Qwen2.5-7B-Instruct rows, pruned with the base profile to show the selection transfers to an instruction-tuned sibling. GSM8K (base models answer near zero at this scale) and MT-Bench (free generation plus a judge) sit at the floor *before* compression and cannot separate selectors after it; LAMBADA discriminates, pure long-range retrieval being exactly what linearisation destroys and deleting attention layers degrades, while HellaSwag and ARC-Easy are broad-competence controls that move less. MMLU enters where the clean model is far enough above chance for headroom — at these scales the 7B checkpoint, not the sub-billion ones. Retention, not absolute accuracy, keeps columns comparable across models with very different clean scores.
Lower is better for the WikiText-2 ratio of perplexities; sampling noise is roughly $\pm 1.5\%$ on LAMBADA, ± 2–3% elsewhere.
A later re-run of the Llama-3-8B LAMBADA cell under the per-item evaluator reads 0.770 clean against the 0.686 here. The gap is the dataset release, not drift. That run read the OpenAI release of the test set, cased and spelled normally; this table uses the standard release, lower-cased and tokenised ("do n't"). With the same evaluator on the same first 2000 passages, the clean model scores 0.767 on the OpenAI release and 0.6845 on the standard one (EXP122, `122_lambada_source_check.py`). Every arm of this table is scored on the standard release, so its comparisons are unaffected.
Llama-3-8B replicates the pattern at the top of the stack: ours wins all four tasks at k=8, deepest-k retakes all four at k=16, the crossover Appendix G.4 shows on ΔNLL. Pythia-1.4B is the exception in both tables, the pair the gate of Appendix G.11 ranks last of twelve ($\rho = +0.43$) and routes to direct measurement; task and ΔNLL results fail in the same place for the same reason, which is what a gate should produce.

Llama-3-8B-Instruct, the three tasks not in Table 1. The base checkpoint's layer sets unchanged, fp32, 500 samples per task (479 for ARC-Easy), accuracy and retention against the clean Instruct model:

arm	ARC-Easy	MMLU	HellaSwag
clean	0.628	0.392	0.574
CurvProbe	0.641 (1.02×)	0.396 (1.01×)	0.568 (0.99×)
deepest-k	0.562 (0.89×)	0.372 (0.95×)	0.572 (1.00×)
free depth prior	0.303 (0.48×)	0.272 (0.69×)	0.284 (0.49×)

Table H.7: **Deletion** (one-shot, matches deployment); retention relative to the clean model. LAMBADA at 2000 samples, the other tasks at 500. "ours" is the curvature-corrected score $\frac{1}{2}\|\epsilon\|^2\kappa^{\mathrm{iso}}$, "depth prior" the free monotone prior of §5.

model (k/L)	selected by	ARC-E ↑	HellaSwag ↑	LAMBADA ↑	WT2 ppl ↓
Llama-3-8B $(8/32)$	**ours**	**0.997**	**0.977**	**0.744**	**1.25**
	deepest-k	0.916	0.946	0.510	2.37
	depth prior	0.472	0.503	0.066	57.11
Llama-3-8B $(16/32)$	**ours**	0.703	0.631	0.057	9.32
	deepest-k	**0.784**	**0.842**	**0.146**	**7.39**
	depth prior	0.484	0.487	0.062	70.22
Qwen2.5-7B-Instruct $(7/28)$	**ours**	**0.991**	**0.817**	**0.479**	**4.35**
	deepest-k	0.633	0.712	0.038	41.06
	depth prior	0.991	0.817	0.479	4.35
Qwen2.5-7B-Instruct $(14/28)$	**ours**	0.576	0.592	0.085	29.26
	deepest-k	0.585	**0.768**	0.002	213.99
	depth prior	**0.636**	0.575	**0.107**	**17.82**
Qwen2.5-0.5B $(6/24)$	**ours**	**0.962**	**0.948**	**0.635**	**1.52**
	deepest-k	0.758	0.876	0.054	5.88
	depth prior	0.689	0.671	0.081	20.12
Qwen2.5-0.5B $(12/24)$	**ours**	**0.932**	**0.905**	**0.399**	**4.08**
	deepest-k	0.633	0.809	0.007	46.58
	depth prior	0.636	0.676	0.028	28.20
GPT-J-6B $(7/28)$	*all three selectors*	0.943	0.958	0.672	1.23
	(identical layer sets; reported to show the degeneracy, not a tie)				
GPT-J-6B $(14/28)$	**ours**	**0.915**	0.776	**0.283**	2.85
	deepest-k	0.752	0.783	0.080	3.64
	depth prior	0.819	**0.825**	0.170	**2.38**
Pythia-1.4B $(6/24)$	**ours**	1.008	0.977	0.768	1.40
	deepest-k	0.941	0.958	**0.771**	**1.24**
	depth prior	**1.017**	**0.991**	0.741	1.47
Pythia-1.4B $(12/24)$	**ours**	0.809	0.699	0.040	12.07
	deepest-k	0.695	0.810	0.020	5.58
	depth prior	**0.835**	**0.806**	**0.149**	**2.68**

Deleting eight of thirty-two attention sublayers by the corrected score costs nothing measurable on broad competence: retention 0.99 to 1.02 on all three, both apparent gains inside sampling noise (± 2.2 points at these sample sizes). Against deepest-k only ARC-Easy separates (2.5 SE); MMLU is 0.8 SE, HellaSwag a tie. The free depth prior loses 31 to 52% of clean accuracy on every one, the verdict LAMBADA and GSM8K return in the main text; WikiText-2 perplexity was not run for this arm set.

The count, arm by arm. At the $k = L/4$ operating point §6 argues for, the corrected score over four checkpoint families and four tasks — ARC-Easy, HellaSwag, LAMBADA and WikiText-2 perplexity, the columns of Table H.7 — wins 24, ties 4 and loses 4 against the two rules together: 14/0/2 against deepest-k, 10/4/2 against the free depth prior. The four ties are Qwen2.5-7B-Instruct, where the prior selects the same layers; GPT-J-6B at $k = 7$ is excluded, all three selectors choosing an identical set, which Table H.7's row reports rather than scores. All four losses at $L/4$ are Pythia-1.4B, the smallest checkpoint in the grid, whose clean per-layer damage spans least. At $k = L/2$ the same four families and tasks read 16/0/16 — the margin is gone, the reversal §G.1 prices and the reason the endpoint is read at $L/4$. Every count above is a count over Table H.7, whose per-cell values are in the released `downstream_*.json` files.

H.7 A PROFILE SHIPPED WITH A CHECKPOINT, PRICED ON ITS DERIVATIVE

The setting this section prices is a checkpoint released with its profile attached, so that a later compressor starts from an ordering fixed before its operation existed. It was an assertion until we ran it. We take two base checkpoints whose profiles are already in the released files, apply each

unchanged to the instruction-tuned derivative of that checkpoint, and rank the derivative's per-layer damage without re-reading any curvature.
Three arms, all evaluated on the derivative: the shipped base profile (**A**), a profile re-read on the derivative itself (**B**), and $\|\epsilon_\ell\|^2$ with no curvature (**C**). Everything is held at the reading budget of the rest of the paper, 32 sequences of 1024 tokens.

Table H.9: **The shipped profile prices the derivative.** Spearman with true per-layer damage on the instruction-tuned model. **C** is displacement alone, **A** the base checkpoint's published profile applied unchanged, **B** a profile re-read on the derivative. On Llama-3-8B the re-read returned a negative isotropic curvature on 4 of 32 layers, so column **B** there is an under-resolved reading rather than an upper bound, and we mark it. `analysis/verify_profileship.py`.

	Llama-3-8B → Instruct			Qwen2.5-7B → Instruct		
operation	C	**A**	B†	C	**A**	B
`linear`	-0.197	$\mathbf{+0.722}$	$+0.545$	-0.191	$+0.218$	$+0.528$
`prune`	-0.353	$\mathbf{+0.689}$	$+0.610$	-0.005	$+0.345$	$+0.580$
`quant4`	-0.230	$\mathbf{+0.618}$	$+0.216$	$+0.253$	$+0.338$	$+0.340$
`quant3`	$+0.084$	$\mathbf{+0.575}$	$+0.551$	$+0.314$	$+0.495$	$+0.525$
`lowrank`	-0.345	$\mathbf{+0.378}$	$+0.298$	$+0.025$	$+0.339$	$+0.567$
mean	-0.208	$\mathbf{+0.596}$	$+0.444$	$+0.079$	$\mathbf{+0.347}$	$+0.508$

What the fine-tune does to the profile. We read the drift against a noise floor rather than against zero. Re-reading the base Llama-3-8B profile under an independent run reproduces it at $\rho = 0.9989$, and the four seeded re-readings of Qwen2.5-7B already in the files agree with the original at $\rho = 0.966$ to 0.978. Against those floors, instruction tuning leaves the Qwen profile untouched ($\rho = 0.981$, inside its own seed spread) and moves the Llama profile a long way ($\rho = 0.616$). The shipped profile prices the derivative in both cases, so the transfer does not depend on the fine-tune leaving the curvature alone.

†Why we do not read column B as an upper bound on Llama-3-8B. The derivative's own reading returned $\kappa_\ell^{\mathrm{iso}} < 0$ on layers 10 to 13, where a curvature estimated by symmetric differences has fallen below its own resolution at this budget: the Instruct model's curvatures are 0.15 to 0.5 times the base model's throughout, and the smallest of them do not survive the difference of two nearly equal losses. Dropping those four layers and recomputing both arms on the remaining 28 leaves the shipped profile ahead ($+0.610$ against $+0.406$), so the negative entries are not the whole of the gap. We have no repeat reading of the Instruct model, so we report this as a re-read that failed at the base model's budget, and draw no conclusion about whether re-reading helps in general. On Qwen2.5-7B, where the re-read is clean, it is worth $+0.161$ over the shipped profile.

H.8 Healing after deletion

H.9 Does deletion damage grow with distance?

The KV-cache saving of §6 is a claim about long sequences, but the quality table behind it is measured at 1024 tokens. It is not a long-context task suite, retrieval, needle-in-a-haystack or LongBench ask whether a compressed model still passes a task, but the deployment claim needs only that the *ordering* of selectors survives the sequence length the cache saving applies to. The position-resolved measurement below tests exactly that, no more: a layer cheap at every position can still be the one a retrieval task depends on, and no per-layer loss profile would predict it.
Setup. Llama-3-8B, k=8 deletion, the CurvProbe, deepest-k and free-prior arms of Table 1, on 16k-token sequences, token NLL bucketed into eight equal-width position bins. One-shot, no alignment, so the measurement is deterministic (unlike its linear-attention counterpart, Appendix D.4). Seven sequences (WikiText-103 holds few 16k-token documents), reported rather than padded (Table H.11).
The result. Averaged over the trained context the arms cost $+0.195$, $+0.965$ and $+5.032$ nats against the clean model. The first two reproduce the 1024-token measurements ($+0.20$, $+0.98$) to within 0.015 nats, so an eightfold longer context moves the corrected score's artefact by 0.005 nats. Deletion damage is flat in position, the property the cache claim needs and not one a fixed-state kernel could promise, a linear-attention state being bounded, so its error should grow with distance, hence measured, not assumed.
Where the measurement stops. Beyond the 8k rule the intact model is extrapolating, so the last four buckets measure the checkpoint's extrapolation failure, not the artefact's. Every deleted arm scores

Table H.10: **Healing the deletion with LoRA does not restore it, and on ΔNLL it costs us the ordering.** Mean over the cells in which each arm was run, at $k{=}L/4$ and $k{=}L/2$: mean token ΔNLL and LAMBADA accuracy before and after spending a LoRA repair on the compressed model. Healing lowers ΔNLL on five of the six arms (SLEB alone rises, from 0.523 to 0.844) but recovers none of them, and LAMBADA falls for every arm on every checkpoint. The four arms run on all ten cells are directly comparable, and the repair does not leave them where it found them: on ΔNLL the free depth prior ($1.636 \to 1.015$) and deepest-k ($2.091 \to 1.160$) both end ahead of the corrected score ($1.670 \to 1.169$), which before the repair only the prior did, and by 0.034 rather than 0.154. On LAMBADA the two end level, at 0.152 against 0.150, both far above deepest-k (0.067). The reversal has a shape: across those four arms the amount recovered tracks the amount lost almost exactly (correlation $+0.98$ between ΔNLL before repair and the reduction), so repair compresses the arms rather than reordering them on merit — it has the most to recover where the most was lost, and deepest-k, the worst arm before repair, recovers the most (0.93 against our 0.50). The ordering survives on the column repair does not target and not on the one it does. We report the reversal rather than the column that favours us: a repair step is something a practitioner will apply, and on next-token loss it erases the margin this paper reports at §6. That is a boundary on the selection, not on the reading.

arm	cells	ΔNLL	ΔNLL healed	LAMBADA	LAMBADA healed
direct measurement of a_ℓ	10	$+1.233$	$+1.027$	0.461	0.184
$\frac{1}{2}\|\epsilon\|^2\kappa^{\mathrm{iso}}$ (ours)	10	$+1.670$	$+1.169$	0.461	0.152
free depth prior	10	$+1.636$	$+1.015$	0.461	0.150
deepest-k	10	$+2.091$	$+1.160$	0.461	0.067
Taylor (Shortened-LLaMA)	8	$+3.360$	$+1.920$	0.468	0.075
SLEB, greedy search	8	$+0.523$	$+0.844$	0.468	0.274

Table H.11: Token NLL by position, 16k sequences, Llama-3-8B $k{=}8$ deletion. The rule at 8k is the model's trained context, not a property of any arm: the intact model itself degrades from 1.73 to 4.59 across it. Left of the rule the arms are separated and the ordering matches the 1024-token table; right of it the baseline is extrapolating and the comparison is uninformative.

	within trained context (0–8k)				beyond it (8–16k)				mean
arm	b1	b2	b3	b4	b5	b6	b7	b8	0–8k
clean	1.89	1.72	1.80	1.73	4.59	7.51	8.52	9.25	1.785
CurvProbe	2.11	1.90	1.98	1.93	4.73	7.08	7.30	7.55	**1.980**
deepest-k	2.91	2.70	2.70	2.69	5.57	7.99	7.86	7.81	2.750
depth prior	6.41	6.82	7.01	7.03	7.26	7.82	7.77	7.89	6.817

better than the clean model there, which says the baseline degrades, not that deleting attention helps, so we report those buckets and conclude nothing from them. More 16k sequences would not change this, a longer trained context would. The depth prior is destroyed at every position, including the first bucket, because it deletes layers 0–2: its failure is not a long-range one.

H.10 Serving footprint, and why we do not quote a concurrency number

§6 reports a per-sequence KV footprint and no concurrency number. This is why. **What is measured.** Memory is read from the KV tensors and the recurrent state, not inferred from a peak. At 16k on Qwen2.5-7B the intact model holds 896.00 MiB (28 layers $\times$ 2 (K and V) $\times$ 4 KV heads $\times$ 128 $\times$ 2 bytes $\times$ 16384, exactly). Converting seven layers gives 678.17 against 678.20 predicted, an agreement of 0.004%. And a *different* seven (21–27, not the selected $\{1, 2, 4, 5, 15, 16, 17\}$) gives the same to the last digit. Footprint is set by how many layers are converted, ΔNLL by which.

Deployment-level probe. Saving is fixed across the probe's context lengths (weight-only quantisation shrinks weights, not activations or KV cache) and throughput converges to the same bandwidth-bound value at long context (Table S12).

Linear-attention decode. Selected-linear against dense, manual decode loop: linear layers store empty KV slots plus their own recurrent state, so decode skips the growing attention over past tokens and the speedup grows with context (Table S13).

Deletion, measured as a skip (EXP 72). The `prune` of §3 zeroes the attention parameters, leaving the module in the graph with its KV slot and FLOPs and no serving payoff. Skipping the sublayer removes both and computes the same function, attention entering the residual stream additively.

Checked first: mean token NLL of skipped against zeroed agrees at 0.00×10^{0} (absolute difference below the 10^{-5} tolerance, exactly zero in bf16) on Llama-3-8B at k=8 and Qwen2.5-7B at k=7, so every quality number for `prune` describes the skipped artefact too (Table H.12 prices it).

Table H.12: Deletion-by-skip, measured. bf16, batch 1, per-sequence KV read from the cache tensors, decode timed over 16 steady-state steps after prefill; the three selector arms are within 0.2% of each other and are reported as one row. The two contexts the text prices are shown; the rest are in Table S15.

model (k)	context	KV MiB		decode tok/s	
		clean	deleted	clean	deleted
Llama-3-8B (8/32)	4k	512.0	384.0	41.65	46.01
	16k	2048.0	1536.0	22.13	26.62
Qwen2.5-7B (7/28)	16k	896.0	672.0	28.98	33.93

Cache saving is exactly k/L in both cases (25.0% and 25.0%), as it must be when the deleted layers hold nothing. The disjoint k-subset control agrees, layers 0–7 on Llama-3-8B instead of the selected eight return 1536.00 MiB at 16k, the same to the last digit. These come from this script's own decode loop against its own clean baseline and are *not* comparable to the linear-attention table (Table S13).

Two traps we fell into. The wrapper's returned `present_key_value` is stored *into* the `DynamicCache`, so a naive probe double-counts it (684.35 rather than 678.17 MiB, a 0.9% inflation), and the bf16 recurrent state differs by exactly that term, so the two errors are indistinguishable. `analysis/exp70_reconcile.py` separates them.

Why no concurrency number. Concurrency is a memory budget over this footprint whose divisor belongs to the serving stack: HuggingFace `transformers` reallocates the whole cache tensor at every decode step (`DynamicCache`), carrying a transient of roughly 0.8× the cache that a paged-attention server would not, so maximum batch at 16k on a 95 GiB card depends on the protocol, not the architecture (Table S14). Of the protocols priced there, one is a full batch search, one a single diagnostic, and one extrapolated from a per-sequence decode measurement at B=8 (1585 and 1265 MiB against footprints of 896 and 678), each measures a different thing, none the architecture. We report the per-sequence footprint, 896 → 678 MiB in all of them, which a reader with a serving stack can divide their own budget by. Quoting one ratio as *the* concurrency gain would report a property of `transformers`.

H.11 Generation: eight-shot GSM8K

Per-arm numbers for §7: greedy decoding, 200 problems, one fixed eight-shot exemplar block across arms, exact match on the final integer. Clean accuracy 76.5% (Qwen2.5-7B), 45.0% (Llama-3-8B), 71.5% (its Instruct variant), all far above the floor `eval_benchmarks.py` requires before a task may be reported.

						ours vs prior, paired	
model	ratio	clean	ours	deepest-k	depth prior	win/lose	p
Qwen2.5-7B	$L/4$	76.5%	1.3%	0.7%	1.3%	0/0	—
Qwen2.5-7B	$L/2$	76.5%	0.7%	1.3%	0.7%	1/1	1.00
Llama-3-8B	$L/4$	45.0%	**53.3%**	48.9%	12.2%	**46/9**	4×10^{-7}
Llama-3-8B	$L/2$	45.0%	8.9%	4.4%	7.8%	8/7	1.00
Llama-3-8B-Inst.	$L/4$	71.5%	70.6%	71.3%	4.9%	**97/3**	3×10^{-25}

Arm columns: retention against the model's clean accuracy. Last two: an exact McNemar test on per-item exact-match vectors (problems the corrected score solves and the depth prior misses, against the reverse, two-sided binomial on the discordant pairs, `analysis/gsm8k_paired.py`). The gap holds on Llama-3-8B, both variants, by a margin that is not close. Against deepest-k the same test ties every time (36/37, 28/24, 5/1, 2/1, 1/2. $p \geq 0.22$), the honest reading of a two-line difference on the Instruct row, and why this section is evidence about the *prior*, not the ranking. The Qwen $L/4$ row has *no* discordant pairs, corrected score and prior selecting identical layers: one artefact evaluated twice, the degeneracy §G.1 reports, not a tie between two selectors. On Llama-3-8B at $L/4$ the corrected score's completions are ordinary chains of thought ending in `####`, the prior's non-numeric token loops. Every item's raw completions are in the released JSON, so the extraction rule can be audited without re-running the model.

A fourth selector, Block Influence, was run at all three $L/4$ rows above. At Qwen2.5-7B it retains 2.0% of clean, three items of 200, against the corrected score's two. Its clean arm reads 75.0% rather

than the 76.5% above, five per-item flips on the same problems, and the retention is 2.0% against either denominator. $L/4$ is the point where Block Influence leads this column on Llama-3-8B, over the isotropic set. On Qwen2.5-7B no selector separates from the floor.

The directional set, in the same run. Re-evaluated in one process with Block Influence's and the isotropic sets (`gsm8k_llama38b_k8_samerun.json`; the clean arm matches the released one on all 200 problems), the directional read's set keeps 40.5% against Block Influence's 37.0% (28:21, $p = 0.39$) and the isotropic set's 24.0% (47:14, $p = 2.7 \times 10^{-5}$). The two CurvProbe sets share six layers; what the isotropic set loses is in the two it does not share, 11 and 18.

A TriviaQA short-answer probe on base Llama-3-8B stopped at the floor: clean exact match $0/156$ under the 5-shot Question/Answer format, a prompt/task mismatch, not a compression effect.

Table H.13: **Token NLL by position, 16k sequences.** Llama-3-8B, k=8 deletion, the same three arms as Table H.11, eight equal-width bins of 2k tokens. The intact model degrades from 1.89 to 9.25 nats across the sequence, so the right-hand bins are extrapolation for every arm; the ordering is set in the first 8k, the checkpoint's trained context, where the means are 1.785 (clean), 1.980 (CurvProbe), 2.751 (deepest-k) and 6.817 (depth prior).

tokens	clean	CurvProbe	deepest-k	depth prior
0–2k	1.890	2.105	2.914	6.412
2–4k	1.716	1.901	2.704	6.817
4–6k	1.802	1.982	2.700	7.014
6–8k	1.732	1.932	2.687	7.025
8–10k	4.591	4.729	5.571	7.259
10–12k	7.513	7.080	7.994	7.816
12–14k	8.518	7.298	7.863	7.770
14–16k	9.253	7.549	7.814	7.892
all 16k	4.627	4.322	5.031	7.251

The released file records `n_eval_seqs` $= 24$. The 16k stream supplies 7, the same request-versus-supply gap the Reproducibility Statement records for the 1024-token stream. We report the count the measurement used.

I THE FREE-PRIOR AUDIT AND WHAT SETS ITS RATE

I.1 THE FREE PRIOR, PRICED

§I.1 claims that one forward-only number says whether the free monotone prior is safe on a given checkpoint. Here is the construction, the source of every entry in Table I.3, and why the rule is derived rather than registered.

The generational split, per checkpoint. Table I.1 aggregates. The per-checkpoint rows, with each recipe's release year, organisation and normaliser group read from its configuration rather than assigned by us, are in the supplementary appendix. OPT-6.7B and OLMo-1-1B, added last, sit where their normaliser puts them. Gemma-2-2B joins Gemma-2-9B in the sign-inverted row. Registered before measurement: EXP89 (GPT-NeoX-20B, Yi-1.5-9B), EXP92 (OPT-6.7B) and EXP96, EXP99 (the OLMo-1 pair, Appendix A.1). The † cross-layer OLS interval on $\hat{c}$ contains the prior's $c_0 = 3$ but is *not* the interval the audit decides with. Qwen2.5-7B, Qwen2.5-0.5B and OPT-1.3B, re-measured with eight probes, kept the standard error within 6% of its one-probe value.

What it is not. The figures (1.68 against 5.32 over twenty checkpoints, exact permutation $p = 1.2 \times 10^{-5}$) count OPT-6.7B on the centred side, where its normaliser puts it. Without it the centred mean is 1.47. Bloom-7B1 is instructive: 7.1B, of the high group's scale, yet sitting at 0.19 with the other LayerNorm checkpoints. The scale, organisation and single-lab controls behind this paragraph are in the supplementary appendix.

What is registered, and what is not. The grouping was assigned after the first fourteen $\hat{c}$ values were known, exploratory in the sense Appendix A.1 reserves, but three things about it are not free parameters: the boundary $c = 3$ is the free prior's own, the separation is perfect so there is no threshold to tune, and the split is the most extreme of the 3003 two-group splits of these fourteen points. EXP89 turns it into a prediction: the rule and two never-measured checkpoints (one per side, chosen so that neither organisation nor scale can explain the outcome) are frozen with their snapshot hashes in `prereg_generation_split.json`.

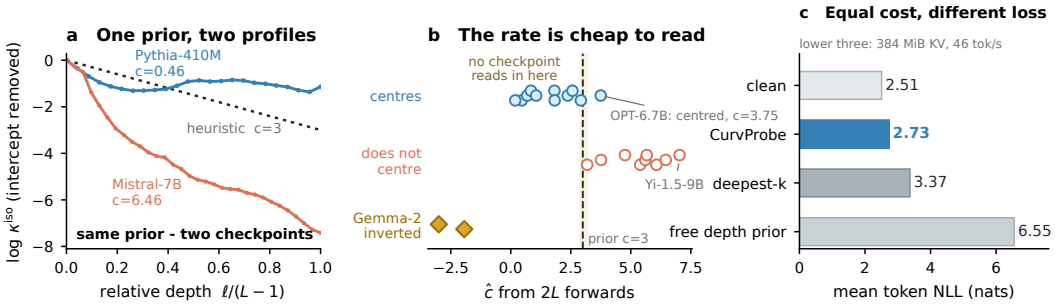


Figure I.1: **The depth prior as a claim about curvature. (a)** One exponential prior against two measured profiles, intercept removed so the slope *is* the rate: fair on Mistral-7B ($\hat{c} = 6.46$), wrong by a factor of six on Pythia-410M (0.46). **(b)** That rate for twenty-two checkpoints, each $2L$ forward passes, no operation and no backward pass. Nothing reads between 2.92 and 3.18, where the prior's constant sits, and rows group by whether the normaliser centres, which is what the rate tracks (§I.1). OPT-6.7B is the one centred checkpoint above the constant and the one place configuration and measurement disagree (§I.3). **(c)** The equal-cost comparison of Figure 2b under the separate WikiText-2 protocol: mean token NLL 2.73 for the profile's eight sublayers, 3.37 for deepest-k, 6.55 for the free depth prior, on a clean 2.51 (perplexities 15.34, 29.12, 700.63, 12.27).

Table I.1: **Curvature decay rate by normalisation.** $\hat{c}$ from $\log \kappa_\ell^{\text{iso}} = a - \hat{c}\,\ell/(L-1)$, measured on the clean checkpoint at $2L$ forward passes, the grouping read from each model's configuration rather than assigned by us. The free depth prior's constant, c=3, separates nineteen of the twenty, OPT-6.7B excepted. The centred group includes OLMo-1, whose LayerNorm has no learnable parameters and which carries a gated MLP (the checkpoint separating the normaliser from the gated block), and OPT-6.7B, the one centred checkpoint past the boundary (§I.3). The audit abstains on *two*, OPT-6.7B and Qwen2.5-7B, under the batch bootstrap of `chat_uncertainty.py`, the interval every abstention decision here is made with; the wider cross-layer OLS interval, used by no decision, would put Qwen2.5-0.5B and Qwen2.5-7B on the line instead, the two disagreeing about *which* checkpoints are unresolved. The Gemma-2 checkpoints are listed apart because their curvature *rises* with depth.

normalisation	n	$\hat{c}$ range	mean
centred (LayerNorm)	11	0.19–3.75	1.68
— the depth prior's $c = 3$ falls here, in the gap —			
uncentred (RMSNorm)	9	3.18–7.03	5.32
sign-inverted	Gemma-2-2B, Gemma-2-9B	-1.94, -3.00	—

checkpoint	recipe era	why this one	frozen prediction
GPT-NeoX-20B	2022	20B, three times larger than any checkpoint now below the gap	$\hat{c} < 3$
Yi-1.5-9B	2024	a lineage and organisation absent from the derivation set	$\hat{c} > 3$

Outcome: both confirmed. GPT-NeoX-20B reads $\hat{c} = 2.569$ and Yi-1.5-9B $\hat{c} = 7.030$, on the predicted sides, both recorded in the registration file. Run twice on different accelerators, Yi-1.5-9B's two κ^{iso} profiles differ by at most 8.7×10^{-8} and give the same $\hat{c}$ to four decimals. Adding both tightens the empty interval from $(2.37, 3.18)$ to $(2.57, 3.18)$, still containing the prior's $c = 3$, and takes the exact permutation p from 3.3×10^{-4} to 8.7×10^{-5}, and on the full corpus to 1.2×10^{-5}.

The record outside the derivation set. Under a fixed inclusion rule: the fourteen the split was chosen on are separated by construction and `prereg_generation_split.json` names them, so the complement is not ours to choose after the fact. The normaliser label applied to the other eight, and six of eight land on the predicted side. Three of the four (*) frozen with a written prediction before measurement hold, and both failures are §I.3 and Appendix I.7. This weaker number, not the corpus-wide nineteen of twenty, should carry the reading. Gemma-2 is unscored, the sign-inversion criterion is recorded as `sign_inverted_case` before either registered checkpoint was measured, and counting both Gemma-2 checkpoints as uncentred failures would make it six of ten and the exact test $p = 0.055$.

checkpoint	normaliser	predicted	$\hat{c}$
OLMo-1-7B	centred	< 3	0.69 ✓
OLMo-1-1B	centred	< 3	0.84 ✓
GPT-NeoX-20B*	centred	< 3	2.57 ✓
OPT-1.3B	centred	< 3	2.92 ✓
OPT-6.7B	centred	< 3	3.75 ×
Yi-1.5-9B*	uncentred	> 3	7.03 ✓
OLMo-2-7B*	uncentred	> 3	3.71 ✓
OLMo-2-1B*	uncentred	> 3	1.68 ×

A measurement-code control. Eleven of the seventeen profiles predate a fix to the evaluation batching (fragments scored without a leading document-start token, inflating NLL severely) and six, including both registered checkpoints, follow it. The fix is a no-op without a token to prepend, so the boundary comparison is clean without re-measurement. Where the effect can exist we measured it: GPT-J-6B gives $\hat{c} = 2.365$, 2.373 and 2.333 with the token and without it, a spread of 0.04 against a fitted standard error of 0.12, and Pythia-410M moves by -0.002. The fix neither creates the era split nor removes it. The tokenizer partition behind this control is in the supplementary appendix.

The confound the registration does not settle. $\hat{c}$ is strongly collinear with the growth rate of the substitution displacement $\|\epsilon_\ell\|$ ($\rho = +0.846$ over the fifteen checkpoints with a substitution grid), and regressing either out removes the group separation, so the two cannot be told apart at this sample size. Pre-LN variance growth (Sun et al., 2025) is a plausible common cause. The era split is not a restatement of that growth: the growth rate does *not* separate the eras (pre-LLaMA $[-0.08, 2.38]$ against LLaMA-era $[1.91, 3.77]$, overlapping) while $\hat{c}$ does. We state the finding as a change in the joint depth-dependence of the two factors whose product is damage, not as a claim about which moved.

GPT-NeoX-20B is the load-bearing half: the low group tops out at 7.1B, and a 20B older-recipe checkpoint below 3 beside a 0.5B newer-recipe one at 3.76 rules scale out by construction. If either prediction fails we report it in Appendix A.1 and withdraw the generational reading, keeping the audit, which is scored per checkpoint against a measured $\Delta\rho$ and never consults the label.

Construction. The prior is $\exp(-c\ell/(L-1))$ with $c = 3$: leave-one-out over the eleven models gives $c \in [3.2, 3.60]$, rounded *down* so the unsafe call widens rather than narrows. No published method fixes the exponent, so the prior is ours, calibrated on the older checkpoints and held fixed. Its claim is that a rule of this form is wrong on the uncentred checkpoints, not that a published algorithm hard-codes 3. Being a claim about a rate, it is compared against the checkpoint's own, and the realised $\Delta\rho$ is $\rho(\frac{1}{2}\|\epsilon\|^2\kappa^{\text{iso}}, a) - \rho(\frac{1}{2}\|\epsilon\|^2 \exp(-3\ell/(L-1)), a)$ over the five operations of §3. A checkpoint is labelled unsafe when $\Delta\rho > 0.05$, the margin frozen in EXP88 before any held-out checkpoint was measured and below the largest win the prior ever takes (0.082, on GPT-J-6B).

What the audit scores. Seventeen checkpoints pass the probe's validity check. The rule calls fifteen, abstains on two, and untuned is correct on 12 of the fifteen, three of them among the six not used in its derivation. Of the other three, OLMo-1-1B and OLMo-1-7B, registered before measurement to test the safe side (EXP96, EXP99), both refuted it, which with GPT-2-medium makes the safe side $4/7$, hence the one-directional claim of §I.1, while the unsafe direction is $8/8$; the third, OPT-6.7B, is one of the two abstentions. Refitting the threshold leave-one-out is worse than not fitting it, and rank agreement between κ^{iso} and the prior is near chance ($12/17$): $\text{Spearman}(\kappa^{\text{iso}}, \exp(-3\ell/(L-1)))$ runs from 0.30 to 1.00 over the seventeen and clears 0.87 on eleven of them, and Spearman discards the rate by construction.

Bloom-7B1 was scored late, and it is a miss. The decision rule of §I.1 sends a checkpoint whose quadratic probe collapses at the validity check to direct measurement without a profile, and Bloom's first run collapsed there (remainder 7.6). The collapse was a harness bug, not ALiBi (Appendix F.7). On the fixed harness Bloom passes (0.003) and reads $\hat{c} = 0.19$, a call of *safe*, against a true $\Delta\rho = +0.660$. Folded in, the rule is $12/16$ determinate.

The asymmetry. Over the eleven unsafe checkpoints the rule calls, trusting the prior costs 0.452 in mean ρ against 0.053 for the probe on the four safe ones, a ratio of 8.6 that argues for running it by default. On the held-out extremes of Gemma-2-9B, Mistral-7B-v0.3 and LLaMA-13B, none used to build the rule, the prior reaches $\rho = -0.397$, -0.220 and -0.135 (worse than not looking at the model in every case) against $+0.706$, $+0.606$ and $+0.577$ for the corrected score.

Status, stated plainly. $\hat{c}$ was chosen after the eleven derivation $\Delta\rho$ values were known, the two-sided form after Gemma-2-9B. Neither is a fished feature, the prior hard-coding $c = 3$ and assuming a decreasing sign. Gemma-2-9B and Mistral-7B-v0.3 are out-of-sample but were already measured when the rule was written, confirming without registering it. LLaMA-13B registers

it: the rule, its constants, the operation set, the margin and the snapshot hash were frozen in `prereg_llama13b_gate.json` before measurement, and the prediction held ($\hat{c} = 5.39$, unsafe. Measured $\Delta\rho = +0.711$), EXP88 in Appendix A.1.

I.2 No threshold would have served the audit better

The audit's two constants are the depth heuristic's own (the 3 its exponent, the 0 its monotonicity) and the sweep of the supplementary appendix fits the upper one over the whole plausible range against the same fifteen determinate checkpoints: its maximum is 12, attained by $c_0 = 3$, so an oracle choosing the threshold with the answers in hand does no better than the heuristic's own exponent. The lower constant is similarly insensitive: any value in $[-1.0, 0]$ gives $12/15$, the score falling where the rule starts calling slowly-decaying LayerNorm checkpoints unsafe. Fit quality does not predict the misses, and the audit's errors are entirely one-sided: over the fifteen checkpoints it calls, eight true positives, four true negatives, *three* false negatives and *no* false positives (one-sided Fisher $p = 2.6 \times 10^{-2}$), against $11/15$ for calling everything unsafe.

The two-sidedness argument (the prior's own 3 and 0, and the Gemma-2-9B profile that added the direction half at $\hat{c} = -3.00$, 41 of 42 layers positive, $\rho = -0.397$), the per-checkpoint table behind Table I.1, the leave-one-lineage-out fits, the lineage-internal permutations, the training-recipe discussion, the threshold sweep with its per-c_0 counts and fit-quality comparison, and the per-operation GPT-J-6B and Pythia-160M readings are in the supplementary appendix.

I.3 OPT-6.7B: the checkpoint the architecture rule gets wrong

Not degenerate. 31/32 layers give $\kappa^{\text{iso}} > 0$ (GPT-2-medium and Qwen3-8B also have one), fit $R^2 = 0.89$, median relative third-order remainder 0.008, far inside the probe's validity band. $\hat{c} = 3.748$; fitted standard error 0.24 gives $[3.27, 4.22]$, carrying only cross-layer OLS residual. Resampling calibration batches gives $[2.64, 4.14]$, not excluding the boundary: $P(\hat{c} > 3) = 0.91$ (`analysis/chat_uncertainty.py`).

Not a scale effect.

family	$\hat{c}$ by size	rises with scale
OPT	1.3B 2.92, 6.7B 3.75	**yes**
Pythia	160M 1.82, 410M 0.46, 1.4B 1.06	no
Qwen2.5	0.5B 3.76, 7B 3.18	no
Llama	1.1B 5.61, 8B 4.76, 13B 5.39	no
Gemma-2	2B -1.94, 9B -3.00	no
OLMo-1	1B 0.84, 7B 0.69	no

A scale law must hold across families; one family of six moving is that family's property, which we cannot identify.

Not an architectural difference within the family either. The two OPT configs agree on every field that matters: `do_layer_norm_before`, ReLU, elementwise-affine LayerNorm, no removed final norm; only depth (24 vs 32) and width differ. No escape: OPT-6.7B is a LayerNorm checkpoint past the boundary.

The registered test. EXP92 froze two predictions before the damage grid: the architecture rule (LayerNorm, so the free depth heuristic is safe, $\Delta\rho \leq 0.05$) and the prior audit ($\hat{c} = 3.75 > 3$, unsafe, $\Delta\rho > 0.05$). Corrected score beats the heuristic on every operation ($+0.069$, $+0.151$, $+0.028$, $+0.117$), mean $\Delta\rho = +0.091$: audit right.

One operation missing, verdict unchanged. The low-rank cell did not complete: the substitution script recomputes an SVD per evaluation batch, pathological at 32 layers of a 27 GB fp32 model from layer 10 on. Cause: weight-tensor accumulation across the per-batch loop, not the SVD (8 seconds in isolation), allocator fragmentation or thread leakage. Four operations give mean $+0.091$; the fifth must fall below -0.114 to pull it under threshold, vs measured range $+0.028$–$+0.151$. No operation here is that far from its neighbours.

I.4 A mechanism we proposed, and why it cannot be right

We record the one candidate tested for why centring tracks the decay rate, and retract the first argument against it.

The candidate. RMS normalisation leaves the residual stream's DC component (projection onto the all-ones direction) in place; LayerNorm subtracts it per block, so DC accumulates with depth only under RMSNorm. Loss nearly flat along it: a growing DC share puts a growing fraction of an isotropic probe into a flat direction, κ^{iso} falling faster with depth, so higher DC accumulation, higher $\hat{c}$.

Why it cannot work. An earlier version argued by counting: $\kappa^{\text{iso}} = \text{tr}(H_\ell)/d$ averages curvature over d orthogonal directions, DC among them, so flattening it removes at most $1/d$ of the trace, not a factor of three. **That argument is false**, retracted: flattening a unit direction v removes $v^\top H v / \text{tr}(H)$, $1/d$ only when v carries an average share, all of it for $H = \lambda v v^\top$; one direction can dominate a trace, so counting alone bounds nothing. The hypothesis nonetheless predicts DC share grows with depth, faster under RMSNorm.

The measurement agrees, twice over. On GPT-2-medium (LayerNorm) and TinyLlama (RMSNorm) DC share is about 2% of residual norm at every depth in both, growing the wrong way: LayerNorm accumulates faster (logit slope $+0.53$ against $+0.38$). Premise fails too: GPT-2's LayerNorm has a learnable bias re-injecting DC after centring, so "LayerNorm removes the DC component" is false of LayerNorm as usually parameterised.

What this leaves standing, and what it does not. This hypothesis is refuted on its own predictions, in two ways; the search is *not* constrained: the earlier claim that no mechanism acting on a bounded number of directions could set the depth profile of $\text{tr}(H_\ell)/d$ followed only from the retracted counting, and a few dominant directions would do it. We report no measured bound on spectrum concentration, which narrowing to macroscopic-spectrum mechanisms would need. The mechanism behind the split stays open; §I.1 does not rest on it.

I.5 Removing the normaliser's centring, inside one model

§I.1's split of twenty checkpoints by whether the normaliser subtracts the mean, and the OLMo-1/OLMoE pair isolating that feature against a gated MLP, rotary embeddings and block structure, are all between-model comparisons; here centring is manipulated inside one trained checkpoint.

The path. LayerNorm and RMS normalisation are endpoints of $(x-\alpha\mu)/\sqrt{\mathbb{E}[(x-\alpha\mu)^2]+\epsilon}\cdot\gamma+\beta$: α=1 LayerNorm, α=0 RMS normalisation retaining the LayerNorm bias. Wrapping every pre-attention and pre-MLP norm we sweep α, measuring κ^{iso} and clean loss per setting; the α=1 run reproduces the untouched model to $|\Delta\text{NLL}| = 0$ on both checkpoints, the script aborting if not.

	GPT-2-medium (24 layers)			Pythia-160M (12 layers)		
α	$\hat{c}$	s.e.	ΔNLL	$\hat{c}$	s.e.	ΔNLL
1.00	2.074	0.157	+0.0000	1.904	0.183	+0.0000
0.75	2.186	0.134	+0.0016	1.926	0.177	+0.0387
0.50	2.510	0.061	+0.0080	1.996	0.172	+0.0802
0.25	**3.148**	0.100	+0.0284	2.063	0.186	+0.1209
0.00	**3.842**	0.178	+0.0801	2.123	0.187	+0.1632

What GPT-2-medium shows. $\hat{c}$ monotone in α, crossing $c_0 = 3$ at α=0.25 and moving 1.77 for 0.08 nats of loss — 22 units of rate per nat of degradation. A checkpoint at 2.07 with centring intact reads 3.84 without it, inside the range every RMSNorm model in the corpus occupies; the measurement is of a working model, 3.16 nats against a clean 3.08.

What Pythia-160M shows, and does not. The direction replicates, the magnitude does not: $1.90 \to 2.12$, never crossing, at 0.16 nats — 1.3 units of rate per nat, seventeen times less efficient than GPT-2-medium. Whether depth (12 against 24 layers), scale or the parallel attention/MLP block accounts for that we do not know and do not claim; the second checkpoint supports the sign, the magnitude the first.

The inverse, on a checkpoint that never trained with centring. If centring sets the rate, imposing it on an RMS-normalised checkpoint should lower $\hat{c}$; it does not. Sweeping α upward on Qwen2.5-0.5B ($\hat{c} = 3.73$, 24 layers; α=0 the identity, reproduced to $|\Delta\text{NLL}| = 0$):

α	0.00	0.50	1.00
$\hat{c}$	3.730	3.727	3.733
ΔNLL	+0.0000	+0.0035	+0.0174
κ profile vs α=0	1.000	0.997	1.008

The loss rises monotonically, so the model is changed, but $\hat{c}$ moves by 0.004 and the curvature profile by under 1%: the mirror of the GPT-2-medium result does not hold.

Was the inverse a real intervention? A null is informative only if the manipulation happened. With $x = \mu\mathbf{1} + x_{\text{ac}}$, the sweep removes the DC component, of size $\sqrt{\rho}$ relative to the normaliser output, $\rho = \mu^2/\mathbb{E}[x^2]$. Measured at every norm input (EXP95, registered before measurement with a two-sided criterion, both sides failing): $\rho = 1.5\times10^{-4}$, 7.6×10^{-4}, 4.2×10^{-4} for GPT-2-medium, Pythia-160M, Qwen2.5-0.5B — perturbations of 1.2%, 2.8%, 2.1% of each norm output. Small and

comparable, and the registered hypothesis that DC accumulates under LayerNorm and not RMS normalisation is refuted: the largest responder has the *smallest* ρ.

What is left, and what it is not. Loss cost per percent of perturbation: 0.065 and 0.059 nats for the two LayerNorm-trained checkpoints, 0.009 for the RMS-trained one — the same relative change to the normaliser is roughly seven times cheaper in the model that never trained with centring. An observation, not a result, found after the registered criterion failed, with LayerNorm confounded with centring *removed* and RMS with centring *added*, which three checkpoints cannot separate.

Extrapolating past the identity point (EXP97). That confound — LayerNorm checkpoints only ever lost DC, the RMSNorm checkpoint only ever gained it — is separable within one model by sweeping past its identity point, not only up to it. With $x = \mu\mathbf{1} + x_{\text{ac}}$, surviving DC at α is $\mu(1-\alpha)$ on a LayerNorm-trained checkpoint: α=0 restores a full $+\mu$ (already swept, §I.1), α>1 a partial DC of the *opposite* sign. If $\hat{c}$ responds to departure from the trained state regardless of sign, both arms move; if to the DC change's sign, only the removal arm. Registered before measurement (EXP97, two hypotheses at equal specificity, pre-declared inconclusive band):

	identity	overshoot	$\Delta\hat{c}$
GPT-2-medium (LN), $\alpha : 1.0 \to 1.5$	2.074	2.256	$+0.182$
Qwen2.5-0.5B (RMS), $\alpha : 0.0 \to -0.5$	3.730	3.738	$+0.009$

Neither registered prediction holds: $H_{\text{departure}}$ required a move over 0.3, $H_{\text{direction}}$ one under 0.1; 0.182 sits inside the pre-declared inconclusive band, Qwen2.5-0.5B not contradicting it (same sign, smaller). Reported as designed — inconclusive — threshold, checkpoints and grid not revisited. Two further facts about the raw measurement: the RMS-trained checkpoint moves by under a hundredth of a unit in either direction pushed, three sweeps running; the LayerNorm overshoot arm reaches only $0.5 \times |\mu|$ of DC, half the identity-to-zero arm's full $1 \times |\mu|$, so the two endpoint deltas are not equal-sized perturbations. At matched $|\text{DC}| = 0.5 \times |\mu|$ the removal arm's midpoint (§I.1, α=0.5: $c = 2.510$, $\Delta = +0.436$) moves the rate roughly $2.4\times$ more than the addition arm at the same magnitude ($+0.182$) — an asymmetry by sign at equal perturbation size the registered endpoint comparison was not built to isolate.

What this cannot show. Removing centring at inference from a model *trained* with it is not training without it. The sweep shows inference-time depth geometry responding to centring in the checkpoints that trained with it, enough to move one across the boundary the compression literature's heuristic sits on; the inverse result shows the limit is not removed — RMS-trained Qwen2.5-0.5B does not acquire the dependence when centring is imposed, so whatever sets its rate is fixed during training, not by the inference-time norm. Settling that needs matched training runs, a training study not this paper; the mechanism is open.

I.6 Norm placement, and why Gemma-2 is set apart rather than explained

§I.1 sets Gemma-2-2B and Gemma-2-9B aside: RMSNorm checkpoints whose curvature *rises* with depth ($\hat{c} = -1.94$ and -3.00). Table I.2 eliminates eight candidate causes but not the number or placement of a block's normalisers, which is what Gemma-2 changed.

How it is read. For each family we parse the decoder-layer class in the installed `transformers` source and collect the norm modules its `__init__` assigns (`81_norm_placement.py`, output `norm_placement.json`). Counts are *declared* modules; a parallel block's count is not independent — GPT-J shares one norm between attention and MLP — restating Table I.2's block row.

norms per block	checkpoints	$\hat{c}$ range
1 (parallel)	GPT-J-6B	2.37
2	Pythia $\times$3, GPT-NeoX-20B, OPT $\times$2, OLMo-1 $\times$2, Bloom-7B1, Qwen2.5 $\times$2, Qwen3-8B, TinyLlama, Llama-3-8B, LLaMA-13B, OLMoE, Mistral-7B, Yi-1.5-9B	0.46–7.03
3	GPT-2-medium (one is cross-attention, unused here)	1.83
4	**Gemma-2-2B, Gemma-2-9B**	$\mathbf{-1.94}$, $\mathbf{-3.00}$

What this licenses, and what it does not. Norm count co-varies perfectly with the sign of $\hat{c}$ across all twenty-two checkpoints: the only family with four norms per block is the only one that inverts, a *candidate* explanation confounded with every other Gemma-2 design decision on these twenty-two.

Why this is the right form for the scope condition. A reviewer objected that setting Gemma-2 aside on a measured inversion gives no rule executable on a new architecture before measuring it. That is right about the sign criterion; this table answers it: the norm count is parsed from the decoder-layer class, not measured (no weights, no instantiation, no GPU), so our scope condition is *blocks with at*

most three declared normalisers, decidable from config and source before any forward pass. Three qualifications: it postdates Gemma-2's inversion, so it is not registered; it rests on one family, so as an explanation it is confounded, which does not matter for a *scope* condition that need only be checkable and falsifiable; and it is refuted by a four-normed checkpoint from another family that does *not* invert, as by a three-normed one that does. Under it §I.1's record is the six of eight above, Gemma-2 out of scope rather than counted wrong, checkable against the next released model.

What OLMo-2 settles, and what it does not. EXP102 (Appendix I.7) measures a second family that moved its normalisers further than Gemma-2: Gemma-2 *keeps* a pre-attention norm and adds post-norms (four per block); OLMo-2 has *no* pre-attention norm and puts both of its two after the sublayers ($h = x + \mathrm{PostAttnNorm}(\mathrm{attn}(x))$), reading $\hat{c} = 3.71$ on the side its normaliser predicts. Placement *per se* is eliminated: moving norms after the sublayers does not invert the sign, so inversion is not a property of post-placement. The *sandwich* (pre-norm and post-norm at once, four per block) is untouched, OLMo-2 having two, not four (Table I.2). Separating the sandwich needs a four-normed checkpoint from a second family, or Gemma-2's sandwich norms disabled and $\hat{c}$ re-read; neither is run here, and the Gemma-2 inversion stays unexplained.

Two further readings cut against convenience. Its norm-count row is 2, the same bucket as fourteen pre-norm checkpoints spanning 0.46–7.03: the count table gains a member, not a discriminant. And the withheld OLMo-2-1B (Appendix I.7) is also post-norm and positive (1.68), so the sign reading does not depend on the checkpoint whose rate we decline to publish — a sign is coarser than a rate.

I.7 EXP102: OLMo-2 completes the AI2 square

Why. §I.1's controlled pair, OLMo-1-7B (LayerNorm, dense, 0.69) against OLMoE (RMSNorm, mixture-of-experts, 5.66), shares laboratory, gated MLP, rotary embeddings and pre-norm placement but not density, so its gap reads as dense-versus-MoE rather than normaliser. OLMo-2-7B is AI2, dense and RMSNorm — the missing cell.

Registration. `prereg_exp102_olmo2.json`, frozen before any OLMo-2 checkpoint was measured, states three equally specific outcomes: $\hat{c} > 3$ (normaliser sets the rate), $0 \le \hat{c} < 3$ (density sets it, confounding §I.1's pair), $\hat{c} < 0$ (placement sets the sign, as if Gemma-2's inversion generalised). It fixes the protocol, commits in advance to reporting whichever outcome is selected, and leaves boundary, group definition and checkpoint list alone.

Protocol, and the one asymmetry inside it. Both checkpoints were measured with `09_transfer.py` in fp32 at the registered settings: four isotropic probes per layer per batch, thirty-two windows of 1024 tokens, seed 0, $t \in \{0.5, 1.0\}$, third-order probe field of Appendix I.8 off. Batch size is unregistered and differs: the 1B ran the default two windows per batch (sixteen batches), the 7B one (thirty-two), two 7B windows in fp32 not fitting on a 48 GB card. Probes are drawn per (batch, layer): the 7B draws $4 \times 32 = 128$ per layer, the 1B $4 \times 16 = 64$, and the batch bootstrap behind the interval above resamples thirty-two units against sixteen. Neither κ^{iso} nor $\hat{c}$ is defined via batch size (both average over the same thirty-two windows), but the 1B's interval is the wider partly for this reason, not only its poorly conditioned probe.

Two gates before any number was read. The `olmo2` adapter, written for this experiment, does not crash when wrong: it returns plausible numbers that are not the model's. `76_adapter_roundtrip.py` rebuilds each block's attention output from the adapter's own q, k, v and output projection, comparing against the module's forward at three depths: relative error 0.0 on both checkpoints, i.e. bitwise reproduction. Separately, the reconciled measurement stack carrying the new adapter re-measures Qwen2.5-7B at $\hat{c} = 3.1791$ against its known 3.179.

Result. OLMo-2-1124-7B reads $\hat{c} = 3.7098$ over 32 layers ($R^2 = 0.928$; 95% CI $[3.49, 3.86]$ by the batch bootstrap of `chat_uncertainty.py`, resampling batches with all layers held together; the cross-layer OLS interval is the wider $[3.34, 4.08]$). It does not contain 3, so the reading is determinate and calls the checkpoint unsafe for a c_0=3 prior — alone, outcome H1, but the registration did not let us take it alone.

	LayerNorm (centres)	RMSNorm (does not)
dense	OLMo-1-7B 0.69	OLMo-2-7B 3.71 *and* OLMo-2-1B 1.68
mixture-of-experts	—	OLMoE 5.66

Holding density fixed and swapping the normaliser moves the rate $0.69 \to 3.71$ (by 3.02) *across* the boundary; holding the normaliser fixed and swapping density moves it $3.71 \to 5.66$ (by 1.95), both on one side. Density is not inert but does not decide the call: the answer to the dense-versus-MoE objection, in numbers.

The second checkpoint, and why this row is unresolved. OLMo-2-0425-1B, measured under the same registration, reads $\hat{c} = 1.68$, withheld when EXP102 was written because its remainder along ϵ_ℓ is 0.59 against the 0.50 gate `75_curvature_index.py` then applied. A reviewer noted that a gate fitted along one operation cannot license or refuse a *rate*, and that we read it as bearing on the rate here but not for Bloom-7B1; the reviewer was right.

EXP105 (Appendix I.13) replaces the borrowed gate with an operation-free one, registered before it was run. Under it OLMo-2-0425-1B passes at 0.23, so its rate is publishable and we publish it: 1.68, inside $[0, 3)$, outcome **H2**, where OLMo-2-1124-7B's 3.71 is outcome H1.

EXP102's falsification clause, written before any OLMo-2 checkpoint existed on our disk, says what happens next: *if OLMo-2-1B and OLMo-2-7B disagree about which hypothesis they select, both are reported and the row is called unresolved.* They disagree, so **the row is unresolved**: the square's missing cell is two values, 1.68 and 3.71, on opposite sides of the boundary, so it does not separate normaliser from density and the dense-against-mixture-of- experts confound in §I.1's pair stands. We had reported a resolved H1 on the strength of the 7B alone; that reading is withdrawn.

What still stands. The *sign* reading does not depend on which leg you trust: both post-norm checkpoints are positive (1.68 and 3.71), so moving the normalisers after the sublayers does not invert the profile — what Appendix I.6 needs, now resting on two published rates. And $\hat{c}$ separating 3.71 from OLMo-1-7B's 0.69 inside one laboratory is unaffected; what the 1B removes is the claim that density was controlled while that happened.

Two limitations. Along ϵ_ℓ this family sits closer to the validity boundary than any other we index: OLMo-2-7B at 0.4275, against TinyLlama's 0.379 for the next worst; the operation-free gate of Appendix I.13 clears both checkpoints comfortably (0.15 and 0.23), but on the stricter eps-based reading these are the highest numbers in the corpus after Bloom-7B1. Second, neither checkpoint joins the twenty behind §I.1's split (a registered prospective test against a pre-existing twenty), and folding the 7B in alone would move the exact permutation p from 1.19×10^{-5} to 1.42×10^{-5} — a small weakening reported here rather than left to the choice of framing.

I.8 EXP104: THE VALIDITY CHECK WITHOUT AN OPERATION IN IT

The gap. The check routing Bloom-7B1 out of the audit is the relative third-order remainder $|a - (A + B)|/|a|$. A and B are fitted along $\varepsilon = y_{\text{lin}} - y_{\text{sm}}$, one operation's linearisation error; by the t=1 identity of Appendix C.1, $\phi(1) = a_\ell$ exactly, so the numerator is that operation's true single-layer damage. The gate consumes the kind of quantity the audit anticipates: per-layer damage, one t, one operation — not a compressed model end to end, but not nothing; calling it operation-free is wrong. Operation-free is $\hat{c}$.

The fix, and a first-attempt error. `09_transfer.py --probe-resid` fits the same remainder along the *isotropic* probe u: a random direction at matched norm, the one κ^{iso} uses, no substitution. Our first report divided it by a_ℓ, damage along ϵ_ℓ: two directions. The checks looked $15\times$ apart on scale, so we wrote the 0.50 gate could not carry over. **Both withdrawn.** Correct analogue of $|a - (A + B)|/|a|$: $|R_3^{\text{iso}}|/|a^{\text{iso}}|$, numerator and denominator both along u; `09_transfer.py` now stores a^{iso} to form it. That attempt's three checkpoints predate the field and cannot be corrected in post-processing; their numbers are withdrawn, not restated.

I.9 EXP107: ONE ARCHITECTURE, TRAINED TWICE

Registration. Frozen before any training run, with H1 (the uncentred run's $\hat{c}$ exceeds the centred run's by more than 1.0), H0 (they differ by less than 1.0) and H2 (they differ by more than 1.0 the other way) at equal specificity, and this commitment: *the final checkpoint decides. We will not select the checkpoint that separates best, will not change the* 1.0 *margin after seeing the trajectories, and will not re-run at a different scale and report only the run that worked.* One pair was trained. It is the one reported.

Construction. GPT-NeoX, 8 layers, hidden 512, 8 heads, parallel residual, rotary fraction 0.25, vocabulary 50277, $76{,}703{,}744$ parameters. The second build replaces all 17 `nn.LayerNorm` modules with `RMSNorm` of the same weight shape and `eps` *under the same attribute names*, leaving `model_type`, block structure and every projection untouched, so `09_transfer.py` measures both through the existing `gpt_neox` adapter with no new code path. The $8{,}704$ parameters by which the uncentred model's $76{,}695{,}040$ differ are the 17×512 normaliser biases, the treatment, not a confound. Both runs use seed 0, learning rate 3×10^{-4}, batch 16, sequence 512, $20{,}000$ steps and *one materialised list of batches replayed for both*.

step	loss (LN)	loss (RMS)	$\hat{c}$ (LN)	$\hat{c}$ (RMS)	Δ
39	9.6396	9.6397	-0.13	-1.61	*gate fails*
136	7.0553	7.0577	11.83	11.47	*3 layers fit*
473	6.1911	6.1903	1.17	0.46	$r^2 \leq 0.15$
1648	5.6230	5.6238	0.99	1.15	$+0.16$
5742	4.8424	4.8287	1.50	1.39	-0.11
20,000	4.5902	4.5918	**1.56**	**1.44**	$\mathbf{-0.12}$

Which rows are readable. The first three are not: at step 39 the uncentred run fails the rate gate of Appendix I.13 on 3 of 8 layers. At step 136 only 3 layers have positive κ^{iso}, so the near-perfect r^2 is three points and a line. At step 473 the profile is not yet exponential ($r^2 = 0.15, 0.05$). The last three pass the gate on every layer (worst ratio 0.14), fit at $r^2 \geq 0.83$ and agree, and the registration named the last one regardless.

Outcome: H0. $\Delta = -0.12$ at the final checkpoint, standard error 0.33, 95% CI $[-0.78, +0.54]$: $|\Delta| < 1.0$, so H0. The interval is narrow enough to matter (the corpus separation of $+3.57$ and the registered margin both sit outside it) and the direction is very slightly *opposite* to the corpus, at a thirtieth of its size.

What we take from it, and what we do not. The registered scale caveat stands: 164M tokens on a 77M-parameter model, one-thirteenth of the earliest measured point on Pythia-160M's released trajectory, so no null at scale is proven and the mechanism is not refuted. We claim only prediction: centring *predicts* $\hat{c}$ across released checkpoints (all §I.1's audit consumes, the measured $\hat{c}$ and never the label) the causal reading withdrawn from §I.1, the abstract and Table I.2. One asymmetry: the centred arm reproduced its group (1.56 against a group mean of 1.68), the uncentred arm did not (1.44 against 5.32), so if the effect is real the uncentred group acquires it over a budget this run does not reach rather than the normaliser imposing it from the start.

Table I.2: **What else could set $\hat{c}$, and why it does not.** Every row is decided by checkpoints already in this corpus; two rows (scale, depth) rest on checkpoints registered before measurement (EXP89). The last row is what survives.

candidate	what rules it out
rotary embeddings	Pythia, GPT-J, GPT-NeoX and OLMo-1 use rotary and sit low
parallel vs. sequential block	GPT-2-m, Bloom-7B1, OPT-1.3B, OLMo-1 are sequential and sit low
gated feed-forward block	OLMo-1-7B is gated and reads 0.69; gated MLPs sit on both sides
training tokens	more training moves $\hat{c}$ *down* within one run (Pythia-160M: 3.08 at step 1k, 1.5–1.9 thereafter); the uncentred group sits *up*
scale	GPT-NeoX-20B (20B) reads 2.57, below OLMoE-1B-7B; five of six families do not rise with scale
depth	the same pair inverts: 44 layers below 16
organisation	five organisations below the gap, six above; Meta and AI2 on both sides
release date	LLaMA-13B (Feb. 2023) reads 5.39; Pythia, same year, 0.46–1.82
norm placement	OLMo-2 puts both norms after the sublayers; neither checkpoint inverts (1.68, 3.71; App. I.6)
whether the normaliser centres	**not eliminated; isolated by the OLMo-1/OLMoE pair, which does not hold density fixed. Matched training at 77M does not reproduce it (EXP107): the surviving predictor, not a demonstrated cause**

I.10 EXP109: MEASURING AN EXPLANATION WE HAD ONLY ASSERTED

What forced this. Appendix H.2 attributes the selector collapse, under a good quantiser, to error-compensated reconstruction and scale search being more curvature-generic than naive rounding; no result file carried GPTQ or AWQ anisotropy, so we registered the claim (`prereg_exp109_quantiser_genericity.json`) and measured it.

The statistic is the paper's own. spread is the interquartile range over layers of $\kappa_\ell / \kappa_\ell^{\text{iso}}$ that `14_aniso_aggregate.py` reports for every operation; lower is more curvature-generic, $\tau = 0.775$ the EXP79 threshold. Pooled Spearman with $\rho(\|\epsilon\|^2, a)$: 0.697, 95% CI $[0.656, 0.735]$, $n = 1090$; control and quantisers share run, batches, hook, probe, restore.

	RTN (control)	GPTQ	AWQ	$\tau = 0.775$
Qwen2.5-0.5B	0.479	0.408	0.373	all generic
Pythia-1.4B	0.734	0.730	0.813	AWQ above τ
Qwen2.5-7B	0.553	0.536	0.526	all generic
median vs control	—	-0.016	-0.026	

Outcome: H1, and it does not mean what we wanted it to. Both medians fall and five of six cells move the predicted way, the registered criterion, but the gap between operations where the correction

Table I.3: **One forward-only number says when the free depth heuristic is *un*safe on a checkpoint, and does not say when it is safe.** $\Delta\rho$ is the realised gain of the corrected score over the heuristic, meaned over five operations; a checkpoint is truly unsafe when $\Delta\rho > 0.05$, the margin registered in EXP88 and frozen before any held-out measurement, just below the largest margin by which the heuristic ever wins (0.082). Eleven checkpoints derived the rule; Gemma-2-9B and Mistral-7B were measured but not consulted when it was written; LLaMA-13B, OPT-6.7B, OLMo-1-1B and OLMo-1-7B were registered before measurement (EXP88, EXP92, EXP96, EXP99), the last two to test the safe side, which both refuted. Bloom-7B1 was routed out on its first run by a probe collapse that was a harness bug (Appendix F.7); its row is the fixed harness. Per-checkpoint values in Appendix I.1.

checkpoint	$\hat{c}$	ρ(heuristic)	ρ(ours)	$\Delta\rho$	audit
five RMSNorm checkpoints	3.76–6.07	+0.150–+0.477	+0.397–+0.800	+0.247–+0.502	unsafe, correct
four LayerNorm checkpoints	0.46–2.37	+0.577–+0.828	+0.512–+0.771	−0.065–−0.034	safe, correct
GPT-2-medium	1.83	+0.173	+0.417	+0.244	safe, **missed**
held out: Gemma-2-9B	−3.00	**−0.397**	**+0.706**	+1.102	unsafe, correct
held out: Mistral-7B-v0.3	6.46	**−0.220**	**+0.606**	+0.827	unsafe, correct
registered: LLaMA-13B	5.39	**−0.135**	**+0.577**	+0.711	unsafe, correct
no call: OPT-6.7B*, Qwen2.5-7B	3.18–3.75	+0.459–+0.628	+0.536–+0.719	+0.076–+0.091	*abstains*‡
registered: OLMo-1-1B	0.84	+0.455	+0.511	+0.055	safe, **missed**†
registered: OLMo-1-7B	0.69	+0.740	+0.889	+0.150	safe, **missed**†
fixed harness: Bloom-7B1	0.19	−0.119	+0.541	+0.660	safe, **missed**

*the one LayerNorm checkpoint whose *measured rate* is unsafe, and the one place where reading the configuration and measuring the rate disagree (§I.3). The rule itself returns no call here, see ‡. ‡no call returned: $\hat{c}$'s interval straddles c_0. Both measured unsafe. †OLMo-1-1B clears the margin by 0.005, the closest of any scored checkpoint. OLMo-1-7B by 0.100 from $\hat{c} = 0.69$, far inside the safe interval (Appendix I.2).

helps and inert ones is 0.603 (`linear` 0.67–1.53 across these models, int3 rounding 0.43–0.73), the 0.016 shift under three percent of it; Pythia-1.4B's AWQ cell moves +0.079 the wrong way, the table's largest move. A difference this size cannot explain a collapse governed by one twenty times larger; claim withdrawn.

What the run explains, which we had missed. All three round-to-nearest controls are *already* below τ: the correction adds little under int3 because rounding at this bit width is curvature-generic to begin with, not because GPTQ improves on rounding's genericity — Table E.7's other side, `quant4` and `quant3` being operations where magnitude score alone reaches $\rho = 0.93$ and 0.95. Deployed quantisers move it slightly further, changing nothing structural. This account is better because measured, and it predicts where the correction *is* worth paying for regardless of quantiser: spread above τ, known before deployment.

I.11 EXP108: THREE ORGANISATIONS THE READING HAD NEVER SEEN

Why. A reviewer objected, rightly, that the eight checkpoints outside the split's fourteen are not eight independent families and that the scope was fixed after seeing curvature. EXP108 freezes the scope, label rule, boundary and gate, applying them to three organisations absent from the twenty-two (`prereg_exp108_unseen_families.json`, frozen before measurement). Selection, in advance: organisations outside the corpus; a `model_type` in the adapter registry, so no attention code is written; both boundary sides represented, so it cannot pass one-sidedly.

checkpoint	organisation	normaliser	gate	$\hat{c}$	R^2	frozen prediction	
RedPajama-INCITE-3B	Together	centred	0.060	1.64	0.23	$\hat{c} < 3$	✓
Falcon3-7B-Base	TII	uncentred	0.031	2.67	0.37	$\hat{c} > 3$	×
SmolLM2-1.7B	HuggingFaceTB	uncentred	0.077	6.74	0.83	$\hat{c} > 3$	✓

Outcome: WEAKENED, miss real. Two of three, no abstentions: all clear the gate comfortably (0.031 to 0.077 against 0.50). Falcon3-7B is uncentred at $\hat{c} = 2.67$, the first uncentred model here below the prior's constant: by the registration's terms, a straight refutation. We do not reclassify it, move the boundary, or add a fourth checkpoint; all forbidden in advance.

Cost. *Group* difference held: folding all three in takes the exact permutation p from 1.19×10^{-5} to 9.6×10^{-6}, SmolLM2 and RedPajama-INCITE separating the groups more than Falcon3 closes them; means 5.21 uncentred (n=11), 1.74 centred (n=12). *Side* claim weakened: the uncentred group spans 2.67 to 7.03, straddling $c_0 = 3$, having begun at 3.18. The empty interval survives (no checkpoint reads between 2.92 and 3.18; Falcon3 sits below), so the free prior's constant keeps its gap; only the claim that a normaliser label names a side weakens, to 8 of 11 out of sample.

Fits. Neither new checkpoint is discountable on fit without discounting corpus ones: RedPajama-INCITE's $R^2 = 0.23$ is the higher of the three, above Pythia-410M's 0.20 and Bloom-7B1's 0.15, Falcon3's 0.37 beside OLMo-1-7B's 0.38, all published with rates. Low R^2 means the profile departs from a single exponential, the limitation Appendix I.16 prices with three further rate summaries, not grounds for setting one aside either direction. SmolLM2-1.7B, measured twice identically, differs by $\leq 2.5 \times 10^{-5}$ relative in per-layer κ^{iso}, with $\hat{c}$, R^2 equal to three decimals.

I.12 EXP110B: TURNING THE AUDIT'S KNOB INSIDE ONE MODEL

What forced this. §I.1 scores the association between $\hat{c}$ and the free prior's cost *across* checkpoints, where lineage, corpus, scale and normaliser move together with $\hat{c}$ and none is fixed. Deployer needs the within-model reading — weights, corpus and batches fixed, does the depth prior cost more at a higher rate? — and the corpus is silent, so we built a knob. Interpolating the normaliser,

$$\mathrm{N}_\alpha(x) \;=\; \frac{x - \alpha\mu}{\sqrt{\mathrm{mean}\left((x-\alpha\mu)^2\right) + \epsilon}} \odot \gamma + \beta, \tag{18}$$

recovers LayerNorm at $\alpha = 1$, RMSNorm at $\alpha = 0$, $\hat{c}$ moving continuously between them in one trained checkpoint.

The registration was amended once. `prereg_exp110` chose OPT-1.3B — corpus records $\hat{c} = 2.92$, just under the constant — so lowering α would carry it across from the safe side; a gate on the α=1 row alone, before any second α, broke that design: under this protocol OPT-1.3B reads 3.195 with s.e. 0.327, consistent with 2.92 at 0.85 s.e. but already *above* 3.0, interval straddling it. EXP77's sweeps put $\alpha = 1$ near a *minimum* of $\hat{c}$ (removing centring raises it strongly, imposing more slightly), so from 3.195 no route back down through the threshold: crossing prediction unsatisfiable, and `prereg_exp110b` drops the claim to the monotone relation the crossing rests on. Gate saw only the treatment variable and the per-layer damage range — no $\Delta\rho$ at any other α existed — host not swapped, swapping after an outcome being what the registration prevents.

Frozen before the sweep. H1: $\mathrm{Spearman}(\hat{c}, \overline{\Delta\rho})$ over the five α at least $+0.7$ *and* $\overline{\Delta\rho}$ at α=0 exceeding α=1 by more than the registered 0.05. H0: $|\mathrm{Spearman}| < 0.4$, §I.1's claim narrowing in the main text to the between-checkpoint association it is scored on. H2: $\mathrm{Spearman} \leq -0.4$, reported as a refutation. Two guards: an operation whose per-layer damage spans under 0.05 nats carries no ranking information — reported, excluded from the mean; a $\hat{c}$ moving only once the clean loss has degraded past 0.1 nats measures a broken model — reported as uninformative, not support. The second matters here, α damaging the network as well as bending its profile.

	OPT-1.3B (treatment)				Pythia-160M (control)	
α	$\hat{c}$	s.e.	ΔNLL	$\overline{\Delta\rho}$	$\hat{c}$	$\overline{\Delta\rho}$
1.00	3.195	0.327	0.0000	−0.016	1.904	−0.105
0.75	2.548	0.284	0.0413	+0.065	1.926	−0.116
0.50	2.672	0.376	0.0885	+0.040	1.996	−0.067
0.25	3.950	0.202	0.1791	+0.061	2.063	−0.106
0.00	6.807	0.128	0.5423	+0.501	2.123	−0.090
Spearman +0.300; α=0 minus α=1: +0.517					+0.300; +0.015	

Outcome: H0. $\mathrm{Spearman} = +0.300$, inside the registered null band: H1 and H2 both refuted. α not a monotone knob on $\hat{c}$ — lowering it from 1 first *lowers* the rate to 2.548, the rise the design was built on arriving only at $\alpha \leq 0.25$, where the clean loss has degraded 0.18 and 0.54 nats. 84% of the movement in $\overline{\Delta\rho}$ lies past the 0.1-nat guard — the case that guard was written for — so the large $+0.517$ endpoint difference is not support. Three α remain inside the clean band, $\hat{c}$ spanning 2.55 to 3.20: too narrow a dose for a response, association negative over those three. Control behaves: $\hat{c}$ moves only $1.90 \to 2.12$, $\overline{\Delta\rho}$ flat and negative, so no part of the treatment arm is α acting on $\Delta\rho$ by a route bypassing $\hat{c}$. §I.1's claim stands where it is scored, on checkpoints; not extended to within-model causation.

The void rule lives in the scorer, not the measurement script. `83_audit_doseresponse.py` averages all five operations; `analysis/exp110b_verdict.py` applies the registered 0.05 exclusion and prints both means, moving the α=1 row from $+0.022$ to -0.016 (`quant4` spans 0.017 nats there) and no verdict — Spearman $+0.300$ either way.

What the run establishes instead. OPT-1.3B is one of the four checkpoints carrying a curvature profile and not the substitution grid (§3); EXP110b's five operations sit outside the grid's counts. Closer to the constant than any checkpoint on the scoreboard, and three readings exist, all at 32×1024 tokens:

reading	probes	$\hat{c}$	OLS s.e.
`transfer_opt13b_r8.json` (the corpus value, Table I.1)	8	2.918	0.350
`transfer_opt13b.json` (superseded by the row above)	1	3.115	0.340
`audit_dose_opt13b_full.json`, α=1	—	3.195	0.327

The three readings span 0.28 and straddle c_0 — a boundary checkpoint whose noise is the calibration draw, not the cross-layer residual. Figure I.1(b)'s gap is bounded below by the published 2.92: no rate the paper uses falls inside $(2.92, 3.18)$ — EXP110b's 3.195 included — but the superseded one-probe reading does, so the gap concerns twenty-two published rates, not the impossibility of a checkpoint reading there. Damage decides: $\overline{\Delta\rho} = -0.016$, inside the margin, so the checkpoint is *safe*, the two readings above 3 would each have been this paper's first false alarm, and the corpus's own reading is the one below. The case supports the rule's form, not any number in it: a threshold on a point estimate is not stable at 0.28 of spread, and the three-way rule declines where that spread lives.
OPT-1.3B is not folded into the scoreboard — measured under this registration, it lacks the per-batch κ^{iso} series the batch bootstrap of `chat_uncertainty.py` decides every abstention with, which none of the three files carries. The selection is on the predictor and predates the outcome: the host was chosen for its published 2.92, before its damage existed.
Pythia-160M, a scored row of Table I.3, re-measured under different batches and a different script: $\hat{c} = 1.904$ against the scoreboard's 1.818, $\overline{\Delta\rho} = -0.105$ against -0.057 — same safe call, neither reading within 0.10 of the criterion, `prune` the operation carrying it in both.

I.13 EXP105: A VALIDITY GATE FOR RATE CLAIMS, WITH NO OPERATION IN IT

What forced this. $|a-(A+B)|/|a|$ is fitted along a particular operation, so it may route Bloom-7B1 out of the *audit* (a damage claim) but may not withhold OLMo-2-0425-1B's *rate* while Bloom-7B1's rate stays in the normaliser split. One number was serving two jobs; a reviewer was right, and EXP105 replaces the borrowed gate for rate claims with an operation-free one.
What the audit costs to *run*, as opposed to *validate*. A reviewer asked what decision curvature reading improves when an operation's per-layer damage is unmeasured, noting that the R_3 gate consumes a_ℓ; both right, of different things. *Validating* the audit required damage — $\Delta\rho > 0.05$ scored against measured a_ℓ on five operations per checkpoint — but *running* it does not: the deployed procedure is three operation-free steps:

1. Check the scope condition — at most three declared normalisers per block (Appendix I.6) — from the model source; no forward pass.
2. Read κ^{iso}_ℓ with $2L$ forward passes on the clean checkpoint and fit $\hat{c}$: no operation, no backward pass, 207.7 s at Pythia-410M.
3. Apply the isotropic gate given below, from the same probes, with no operation: if it passes, $\hat{c} > 3$ or $\hat{c} < 0$ says the exponential prior is unsafe; otherwise no rate is published.

Before touching a layer the practitioner decides whether to trust a monotone exponential depth prior on *this* checkpoint or to pay for measurement; where it fires it is right every time $(8/8)$, and its value is firing before the budget is spent, not being cheaper than measuring (at matched budget it is not, Appendix D.10).
The R_3 gate of Table I.3 routes Bloom-7B1 out of the *scoring* and is operation-based because scoring is a damage claim; a deployment without a chosen operation would use step 3, which Bloom-7B1 passes (0.476), so the deployed rule keeps a checkpoint the scored table sets aside. The rule is one-directional: it does not certify the safe side, on which two registered predictions were spent (EXP96, EXP99).
The gate, frozen before measurement (`prereg_exp105_isogate.json`):

$$\operatorname*{median}_{\ell} \frac{|R^{\text{iso}}_{3,\ell}|}{|a^{\text{iso}}_{\ell}|} > 0.50 \quad \Longrightarrow \quad \text{no rate is published.}$$

The threshold is inherited, not chosen: 0.50 is what `75_curvature_index.py` applied since before EXP102, the third-order remainder not exceeding half the damage the quadratic model is asked to explain. That form names no direction, so the number carries from ϵ_ℓ to u without re-tuning; three outcomes were registered at equal specificity, including the costliest.

	isotropic (the gate)	along ϵ_ℓ	$\hat{c}$
Bloom-7B1 (ALiBi)	0.476	0.916	+0.97
OLMo-2-0425-1B	0.230	0.586	+1.68
OLMo-2-1124-7B	0.145	0.428	+3.71

What it returned. All three pass, both consequences registered in advance. The first favours us: Bloom-7B1 fails the operation-based check but passes the operation-free one, so the two measure different things, as we asserted when keeping its rate in the split — a quadratic model can fail to predict one operation at a layer while the isotropic curvature there stays readable — and §I.1's numbers are unchanged. The second goes against us: OLMo-2-0425-1B passes at 0.230, so we publish its rate, $\hat{c} = 1.68$, putting EXP102's two legs on opposite sides of the boundary, and by that registration's falsification clause the row is unresolved (Appendix I.7).

Two things to hold against this. Bloom-7B1 clears the gate by 0.024 — a pass under a pre-measurement threshold, reported as one; a reader who set the gate below 0.48 gets the opposite answer, the split losing a checkpoint at the cost computed in advance (n=10, mean 1.83, $p = 3.2 \times 10^{-5}$). Second, the gate has been applied to three checkpoints, not all twenty: the registration makes screening the other nineteen conditional on Bloom-7B1 *failing*, and it passed, so another checkpoint might fail a gate we have not applied to it — stated rather than describing the corpus as screened.

I.14 Does the inversion window predict the inversions?

Proposition 10 is the audit's bridge: trust in a fixed-rate prior turns on the rate mismatch $|\hat{c} - c_0|$, not profile similarity.

What it does not say. A wide window is an opportunity for inversion, not an inversion: a pair (ℓ, m) inverts only when $\log r$ falls *inside* it, so the operation's magnitude spectrum sets how many turn over — at $r = 1$ (all layers perturbed equally) $\log r = 0$ lies below $\min(\hat{c}, c_0)\Delta$ for every positive Δ: nothing inverts. Hence $\hat{c} > c_0$ does not logically imply a misranking heuristic: the audit rests on the measured incidence below, not on a proposition assuming an exactly exponential profile that none measured has (R^2 from 0.15 to 0.98).

The comparison. Per layer pair (ℓ, m) in each model–operation cell, *predicted* inversion is the proposition's criterion $\min(\hat{c}, c_0)\Delta < \log r < \max(\hat{c}, c_0)\Delta$ from the fitted $\hat{c}$ and measured $\|\epsilon_\ell\|$ only; *actual* inversion is whether the corrected score (raw κ^{iso}, not the fit) and the free prior order that pair oppositely — the gap being the cost of the exponential idealisation.

predictor of the realised inversion rate	Spearman	mean |error|
each checkpoint's own $\hat{c}$ (Proposition 10)	$+\mathbf{0.787}$	$\mathbf{0.069}$
one universal constant, the corpus mean $\hat{c} = 2.86$	$+0.075$	0.270

Over 89 cells (eighteen checkpoints, five operations) the idealisation costs about seven points of inversion rate; within that, the proposition predicts the profile.

Why row two matters. The counterfactual this paper argues against is replacing each checkpoint's $\hat{c}$ by the corpus mean: one constant for every model, what a fixed-rate depth prior assumes and what 2.86 almost is ($c_0 = 3$). Prediction is then at chance, error 3.9 times larger. The rate is what the inversion window is a function of, not a statistic merely correlating with damage, and must be measured per checkpoint.

A door closed. The audit thresholds this quantity into a binary call that §I.1 reports reliable one way only. Deciding on the continuous quantity we tried: neither thresholding the predicted inversion rate nor $|\hat{c} - c_0|$ beats the gate, both best thresholds degenerate, recovering the majority baseline, not separating the classes, the four safe checkpoints (0.127, 0.147, 0.257, 0.282) interleaved with unsafe ones. The window predicts *how many* pairs invert; safety is whether they cost rank correlation, and many that invert do not matter. So $|\hat{c} - c_0|$ is the stronger *description* — it predicts how often the prior mis-orders layers whether or not the threshold at 3 resolves a checkpoint — not a better *decision rule*. Computed by `82_window_prediction.py` from the released `substitutions_*.json`; output `window_prediction.json`, one row per cell.

I.15 The two normalisers separate during training

§I.1's dose-response is inference-time and cannot show that training *under* a normaliser produces the rate; released checkpoints can, without training, via the $2L$ forward passes the profile costs.

Registered first. EXP98 froze one prediction: an RMS-normalised checkpoint's $\hat{c}$ rises over training by at least 1.0, where Pythia-160M's falls; the opposite reading (rate set early, unmoved by training) stated at equal specificity, with pre-declared inconclusive band 0.5–1.0. TinyLlama-1.1B: already in the corpus, not added for this.

	Pythia-160M (LayerNorm)	TinyLlama-1.1B (RMSNorm)
earliest measured	3.078 *(step 1k, ≈2B tok)*	3.837 *(105B tok)*
overshoot	1.541 *(step 79k)*	6.913 *(503B tok)*
latest measured	1.911 *(step 143k)*	5.577 *(3T tok)*
its group's mean	1.75	5.32
final distance from it	+0.16	+0.26

Net change -1.17 (Pythia), $+1.74$ (TinyLlama): H1 confirmed on its registered criterion. Both overshoot their group mean, settling within 0.26; neither monotone (TinyLlama peaks at 6.913 then declines). Registered band's *no clear direction* clause excludes this run: ends 1.74 above start, every later point above the first.

What this is not. Two families, not one architecture trained twice: data, tokeniser, optimiser, scale, sampled token count differ (2B vs 105B earliest). Only travel direction is a claim, not where each family *starts*. Both earliest points fall between 3 and 4, near the depth heuristic's constant (observation, not result: sampling cannot support it). Closing it needs an RMSNorm family at Pythia's token counts, which this paper does not run. `80_training_trajectory.py`, registered `prereg_training_trajectory.json`, output `trajectory_tinyllama.json`.

I.16 Does the split depend on the summary statistic?

$\hat{c}$ summarises a profile as a fitted exponential's slope, but several profiles are not exponential: OLMo-1's profile drops only in its last two layers, and R^2 runs 0.15–0.98. We recompute it under four summaries of the same profiles plus a deliberately failing control, on the eighteen checkpoints then measured (nine per group), excluding later OPT-6.7B and OLMo-1-1B.

summary of the profile	LayerNorm (9)	RMSNorm (9)	separated	exact p
OLS slope ($\hat{c}$, used in the paper)	0.46–2.92	3.18–7.03	yes	2.1×10^{-5}
head-to-tail log drop	0.98–3.52	4.34–11.04	yes	2.1×10^{-5}
half-split gap (mean log, halves)	0.08–1.60	1.48–3.28	no	6.2×10^{-5}
Theil–Sen slope (median pairwise)	0.34–3.40	2.64–5.83	no	8.2×10^{-5}
Spearman, κ^{iso} vs. depth *(control)*	0.29–1.00	0.93–1.00	no	7.0×10^{-3}

What this settles and what it qualifies. The group difference is no artefact of the exponential fit: every rate-like summary separates the groups at $p \leq 8.2 \times 10^{-5}$, Theil–Sen robust to those non-exponential shapes. Summary-specific is the *perfect* separation: two of four give it; half-split gap and Theil–Sen overlap near the boundary by $0.12/0.76$. $18/18$ is thus a property of the fitted rate, the group difference of the profiles; §I.1's per-checkpoint claims (audit, Table I.1's ranges) are stated in $\hat{c}$ throughout.

The control behaves as theory requires. κ^{iso}–depth Spearman is rate-invariant by construction, so Proposition 10 says it cannot see the inversion-window quantity; two orders of magnitude weaker than the rest, the statistic whose audit failure §I.1 reports ($12/17$, near chance). Proposition, null control and table agree on the rate, not the shape.

I.17 Is the profile inherited? Lineage versus scale

The raw comparison is uninformative, and that is the point. Every κ^{iso} profile falls with depth, dominating raw profile-to-profile correlation: $+0.734$ within a training lineage against $+0.747$ across our eleven checkpoints — *higher* across. So §I.1's question must be asked of the detrended profile, where §5 locates its advantage over a depth prior.

The residual is lineage-specific. Regressing $\log \kappa^{\mathrm{iso}}$ on relative depth and correlating residuals gives $+0.74$ within lineage against $+0.23$ across ($p = 0.0007$ by permutation of lineage labels, 20,000 draws), holding across the grid ($+0.55$, $+0.52$, $+0.48$ at 12, 16, 24 points; $p = 0.0011$, 0.0007, 0.0014). Subtracting the pooled mean profile instead — removing the common *shape*, not just an exponential trend — gives $+0.53$ against $+0.13$, $p = 0.054$, reported as marginal.

Grouping checkpoints into lineages is a judgement call, so we report all four defensible ones. Pythia merged with GPT-J: seven lineages, $p = 0.0004$; separated (as we report), eight, $p = 0.0007$; Qwen2.5/Qwen3 separated, nine, $p = 0.0002$; TinyLlama/Llama-3 separated, ten, $p = 0.003$. The primary split survives every grouping; only secondary controls move, *which* weakens depending on the grouping — at ten lineages the confound control below falls to one within-lineage pair, at eight the pooled-mean control is marginal — so none was chosen for its p-value.

The obvious confound, removed. Three of the seven within-lineage pairs are Pythia-160M/410M/1.4B, so the effect could be one family's idiosyncrasy. Dropping it leaves four within-lineage pairs against twenty-four across, and the split survives: $+0.63$ against $+0.03$, $p = 0.005$.

It is not a scale effect. The within-lineage pairs span 1 to $16\times$ in parameter count and hold across it — Qwen2.5-0.5B with Qwen2.5-7B at $+0.69$ ($15\times$ apart), TinyLlama-1.1B with Llama-3-8B at $+0.71$ ($7\times$) — while Qwen2.5-7B and Llama-3-8B, matched in scale and separated only by recipe, fall in the across-lineage population.

A hypothesis this refutes. It is tempting to close the loop with §5: the free depth prior wins exactly the four EleutherAI checkpoints, perhaps because the profile has little structure beyond depth there. It does not — the fraction of profile variance left after detrending does not separate the four the prior wins from the seven it loses (Spearman -0.19; medians 0.215 and 0.216), and OLMoE and TinyLlama, with the *least* residual structure, are among those the profile wins most.

Resemblance is not substitutability, and we checked. Borrowing a sibling's profile and selecting with it does not hold: scoring deletion at $k = L/4$ with a donor profile resampled to the target's depth, three of four same-lineage transfers keep essentially all the advantage (102–111%), the fourth destroys it (Qwen2.5-7B's profile on Qwen2.5-0.5B costs 3.17 nats against 0.92 for deepest-k), and two *cross*-lineage transfers work fine (OLMoE-1B-7B to Llama-3-8B, 114%). Donor transfer is not predicted by lineage; the one clean regularity is that GPT-2-medium, whose own profile already fails (§5), is a bad donor. The lineage effect concerns the *shape* profiles share, and one profile per checkpoint remains the price (Appendix D.10).

What this does not establish. "Lineage" here confounds architecture, tokeniser, pretraining corpus and optimiser recipe, and nothing above separates them: seven within-lineage pairs over three multi-member lineages, four once Pythia is removed. We report an effect that earns a targeted experiment — two checkpoints sharing a recipe but not a corpus would settle it — not an established mechanism. `analysis/lineage_residual.py` and `analysis/profile_transfer.py` reproduce every number here, including both refuted hypotheses.

J LIMITATIONS

Every row below is a boundary we measured rather than a caveat we are hedging, and the size of the effect is the number rather than an adjective. We collect them in one place because a reader deciding whether to use the criterion needs the failure surface, not the successes, and because each of these was found by a run that was written to be able to fail.

Table J.1: **Where the criterion does not pay, with the size of each boundary.** The last column names the section that derives the number. ‡the rows where something we measured beats the corrected score, rather than where the corrected score is merely weak.

boundary	what we measured	size of the effect
error-compensated quantisation	with GPTQ or AWQ deployed, every selector lands in a band that also contains the free depth prior: the correction is inert here rather than wrong	at most 0.14 nats of spread, 0.11 under SparseGPT 2:4 (App. H.2, H.3)
budgets past $L/2$	the per-layer ranking stops deciding the outcome, and generation degrades faster than perplexity	GSM8K 0.505 against 0.715 clean at k=8 of 32 (§6, Table 1)
strong anisotropy	on Gemma-2-9B under deletion the global correlation stays positive but moderate, and the deciding layers sit in the tail	$\rho = 0.49$–0.55, with 23 of 89 cells decoupled (App. F.1)
layer interactions dominate	on GPT-2-medium, and at $L/4$ on Qwen2.5-7B and Gemma-2-9B deletion, even *direct* per-layer measurement trails a set chosen jointly; a search restricted to the profile's $2k$ cheapest layers recovers it	$+0.26$ and $+0.59$ nats behind deepest-k on GPT-2-medium (App. G.2); joint damage 18–20$\times$ the single-layer sum, the restricted search ahead of Block Influence at 41–47% of a full search's evaluations (App. G.9)
one operation, fresh checkpoint	reading the profile first costs $2L+1$ passes against measurement's L	207 against 106 s at K=1 on Qwen2.5-7B (App. D.10)
a published selector, same layers‡	Block Influence, given the same base layer sets, leads on Instruct GSM8K and is nominally ahead on Instruct LAMBADA without separating ($p = 0.45$), and trails on ARC-Easy and MMLU; on Qwen2.5-7B the reasoning lead does not carry, and no selector separates above the floor; the directional read's set leads it on Instruct LAMBADA and is level on base GSM8K	0.650 against 0.505, clean 0.715, and 0.462 against 0.444 (Table 1); 2.0% against our 1.3%, deepest-k 0.7% (App. H.11)
a competition inside a budget‡	at k=7 of 28 on Qwen2.5-7B, converting the chosen layers to linear attention selects the same layers as the free prior, so the cell is retired	no separation beyond $\|\epsilon\|^2$ (App. G.8)
a corpus shift	thirty-nine of forty cells transfer to held-out C4; the fortieth loses on Pythia-410M pruning	$+0.271$ nats, and it is the only loss (App. G.6)

Beyond the table, four things we did not test at all, and would rather name than let a reader assume. **Scale.** The largest checkpoint on which an operation is measured anywhere in this paper is LLaMA-13B. Seven of the eighteen grid checkpoints are under 1.5B, and no operation above 13B is measured, so the profile's behaviour at frontier scale is an extrapolation we do not make. (The rate audit of §I.1 reaches 20B, but by reading $\hat{c}$, not by compressing.) **Long-context tasks.** We check that the quality ordering survives at 16k tokens (App. H.9) and we do not run a retrieval or needle-style benchmark, so the deployment claim is the weak one: the saving is bought at the lengths where the cache matters, and the ordering does not change. **Training-time compression.** The profile is read from a finished checkpoint. Compression that changes the weights while they are still moving is outside what we measured. **Anisotropy beyond the scalar.** We measure that H_ℓ varies in more directions than $\kappa_\ell^{\mathrm{iso}}$ can carry (App. F.9) and then use the scalar anyway, because the anisotropic object did not order layers better in the one comparison we ran. Where those two facts will part company is the open question this work leaves behind, and it is a question about the object, not about the estimator.