Multi-Signal Interpretability Identifies Task-Critical Circuits for Efficient Model Compression

Srikar Vardhan Mangadoddi
National Institute of Technology Silchar
Petavue
srikar@petavue.com

Abstract

Post-training quantization methods such as GPTQ and AWQ minimize reconstruction error without asking which weights matter for the task. We combine six interpretability signals to select a sparse skeleton of task-critical MLP neurons that is kept at 16-bit while the rest of the model is quantized. On Gemma-2-2B fine-tuned for SQL generation, the 0.54% skeleton with 3-bit MLPs and 8-bit attention retains **100% of baseline-correct outputs at 4.03× compression**, versus 75.2% for uniform 4-bit. On Llama-3-8B fine-tuned for Cypher generation, where uniform 4-bit loses 23 points, the 0.46% skeleton recovers 14.4 of them and a random skeleton of equal size recovers none. Three signals are individually effective and three are not, and the effective set differs between models, so the consensus rule provides robustness no single signal does. Selection matters where uniform quantization leaves headroom, which suggests measuring headroom, selecting by consensus, and protecting as a recipe for larger models. GPTQ error compensation degrades 3-bit layouts by 16 to 18 points regardless of protection.

1 Introduction

Fine-tuned language models are now the standard deployment unit: practitioners routinely adapt base models to tasks such as code generation, summarization, or SQL synthesis, and deploying them in resource-constrained settings requires compression. Post-training quantization methods such as GPTQ (Frantar et al., 2023) and AWQ (Lin et al., 2024) lower weight precision effectively, but they minimize reconstruction error without asking *which neurons matter for the task.*

Fine-tuning creates task-specific computational pathways, circuits in the loose sense of neuron subsets critical for a learned capability (Olah et al., 2020; Elhage et al., 2021), and these changes are sparse and localized (Panigrahi et al., 2023; Prakash et al., 2024). If we can identify this circuit, we can protect it at full precision and compress everything else. TaCQ (Xiao et al., 2025) takes a step in this direction with weight-perturbation and gradient signals at weight granularity. We ask a simpler question first: does the choice of protected neurons matter at all, and if so, which signals identify them?

We propose **CircuitTier**, which combines six interpretability signals (Figure 1) to select a sparse skeleton of task-critical MLP neurons that is kept at 16-bit while the remaining weights are quantized. The signals have low top-k overlap (mean Jaccard of top-1,295 neurons: 0.012), so each identifies neurons the others rank as unimportant.[1] On Gemma-2-2B (Gemma Team et al., 2024) fine-tuned on TinySQL (Harrasse et al., 2025), the 0.54% skeleton at 16-bit with 3-bit MLPs and 8-bit attention retains 100% of baseline-correct outputs at 4.03× compression, versus 75.2% for uniform 4-bit. On this model the precision layout does the work: the same retention is reached with no skeleton, because low-bit MLPs with 8-bit attention leave nothing to recover. On Llama-3-8B (Grattafiori et al., 2024) fine-tuned for Cypher generation (Ozsoy et al., 2024), uniform 4-bit loses 23 points and the skeleton matters: protecting the 0.46% our signals select recovers 14.4 points, while a random skeleton of the same size recovers none. Selection matters where uniform quantization leaves headroom, which suggests measuring headroom, selecting by consensus, and protecting as a recipe for larger and harder-to-quantize models.

Our contributions are:

1. A framework that selects a sparse skeleton from six interpretability signals at neuron granularity, reaching 100% task retention at

[1] Code, results, and an interactive 3D circuit explorer are available at https://github.com/M-SRIKAR-VARDHAN/circuittier.

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 140–154
November 6-10, 2026 ©2026 Association for Computational Linguistics

4.03× on Gemma-2-2B.

2. Evidence that the identity of protected neurons matters when the compressed model has something to lose: on Llama-3-8B the multi-signal skeleton recovers 14.4 points over uniform 4-bit while random protection recovers 0.0 (n=894). Three signals are individually effective and three are not, the effective set differs between models, and the consensus rule matches the best single signal on both without knowing which it is.

3. Two negative results: GPTQ error compensation degrades 3-bit layouts by 16 to 18 points regardless of protection while helping at 4-bit, and zeroing the neurons all six signals rank lowest costs 8 points.

2 Related Work

Post-training quantization methods reduce weight precision without retraining: GPTQ (Frantar et al., 2023) uses second-order information, AWQ (Lin et al., 2024) protects salient weights via activation magnitudes, and SpQR (Dettmers et al., 2023) isolates outlier weights. Mixed-precision methods such as HAWQ (Dong et al., 2019) assign bit-widths per layer via Hessian sensitivity, while OWQ (Lee et al., 2024) and SqueezeLLM (Kim et al., 2024) operate per weight group. These methods use one or two mathematical criteria at weight or layer granularity and are task-agnostic by design.

On the interpretability side, LLM-Pruner (Ma et al., 2023) uses gradients to identify removable structures and ShortGPT (Men et al., 2024) removes layers by importance. Pruning-at-initialization methods such as SNIP (Lee et al., 2019), GraSP (Wang et al., 2020), and SynFlow (Tanaka et al., 2020) score parameters of an untrained network to select a trainable subnetwork; we instead operate on an already fine-tuned model(Vardhan and Teja, 2026) and target the circuit introduced by fine-tuning, using signals that presuppose a base checkpoint. Circuit discovery methods (Conmy et al., 2023; Syed et al., 2023) locate the components responsible for a behavior; we use one of them, attribution patching, as a signal.Analyses of what fine-tuning changes report that it strengthens existing mechanisms rather than building new ones (Prakash et al., 2024), that its effect is localized to a small fraction of parameters (Panigrahi et al., 2023), and that it can be reverted by editing a sparse set of components (Jain et al., 2024); these motivate looking for a sparse protected set at all. Most closely related, TaCQ (Xiao et al., 2025) combines weight magnitude, gradient, and quantization-effect terms into a single per-weight saliency score. Its components derive from the same backward pass and are correlated by construction, with binary protection at weight granularity. We use six independently computed signals at neuron granularity, compare fine-tuned and base model weights and activations to isolate task-specific circuits, and, unlike prior work, test whether the selected neurons matter against random selection of equal size.

3 Method

Our approach rests on a simple premise: fine-tuning creates task-specific circuits. By comparing the fine-tuned model against its base checkpoint, we identify which neurons the model relies on for the target task and protect them during compression, in three stages (Figure 1): signal extraction, tier classification, and mixed-precision quantization. Throughout, a *neuron* is a single MLP intermediate dimension: one row of the gate and up projections and the corresponding column of the down projection, the atomic unit at which we assign precision.

3.1 Signal Extraction

We compute six interpretability signals for all 239,616 MLP neurons and 208 attention heads (Table 1). *EAP* zero-ablates each neuron and records the change in the target-token logit, on 50 analysis samples. *Gradient saliency* accumulates the absolute loss gradient over each neuron's weights on 20 samples. *Weight magnitude* is the mean absolute value of the neuron's weights and needs no data. *Weight delta* is the mean absolute change in those weights between the base and fine-tuned checkpoints. *Activation delta* is the change in the neuron's maximum activation between the two models over 500 samples. *Edge importance* sums the patching scores of all attention-head edges into the neuron, capturing connectivity the other five per-neuron signals miss. Signals (4) and (5) require both checkpoints, isolating task-specific computation from general-purpose behavior; the method therefore assumes access to the pre-fine-tuning weights. Full definitions are in Appendix B.

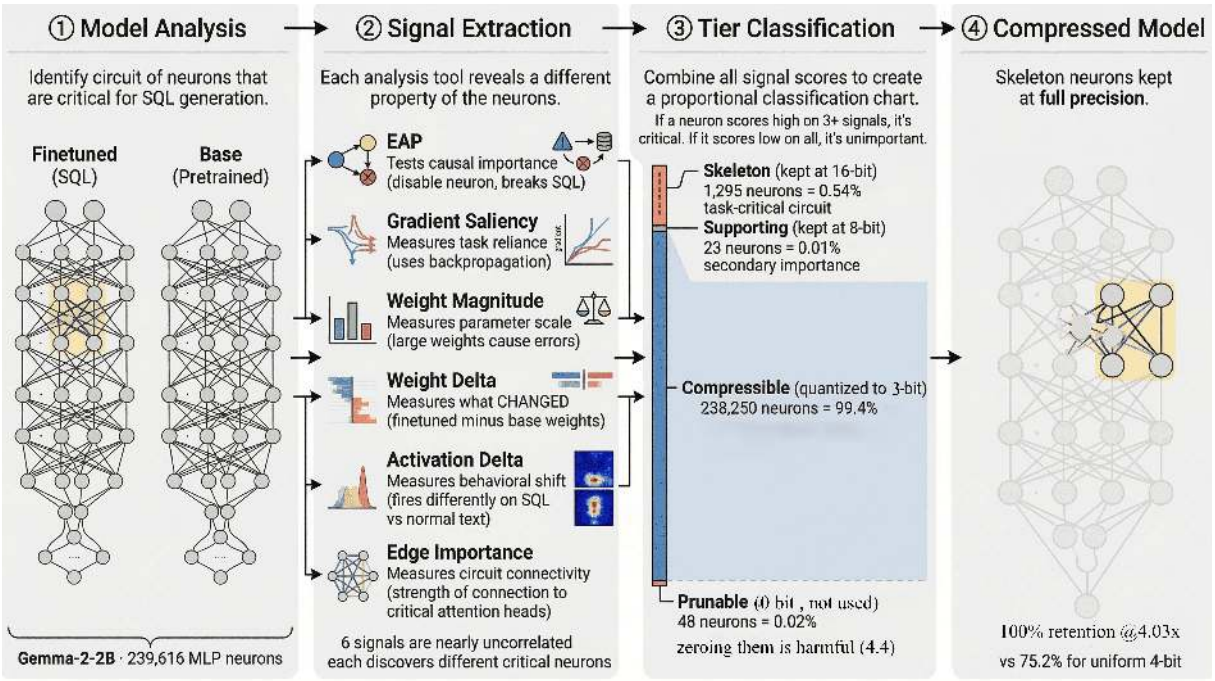


Figure 1: We analyze the fine-tuned and base models using six interpretability signals, each revealing a different aspect of neuron importance. Signals are combined via consensus rules to classify the 239,616 MLP neurons of Gemma-2-2B into precision tiers; the main configuration keeps the skeleton tier at 16-bit and quantizes all remaining MLP weights to 3-bit and attention to 8-bit.

#	Signal	Measures
1	EAP (Syed et al., 2023)	Causal importance
2	Gradient Saliency	Loss sensitivity
3	Weight Magnitude	Parameter scale
4	Weight Delta†	Fine-tuning change
5	Activation Delta†	Behavioral shift
6	Edge Importance	Circuit connectivity

Table 1: Six interpretability signals, each min-max normalized to $[0, 1]$. †Requires the base checkpoint. Mean pairwise Pearson $|r|$=0.098 (Magnitude and Weight Δ: r=0.85); mean Jaccard overlap of top-1,295 neurons 0.012. Definitions in Appendix B.

3.2 Tier Classification

For each neuron's normalized profile $\{s_1, \ldots, s_6\}$: **Skeleton (16-bit):** $\geq$3 signals exceed τ=0.3, or EAP $\geq \tau$ with Gradient or Activation $\Delta \geq \tau$ (1,295 neurons, 0.54%). **Supporting (8-bit):** $\geq$1 signal exceeds τ with a causal signal ≥ 0.2 (23 neurons). **Prunable (0-bit):** all signals below τ'=0.1 (48 neurons). **Compressible:** the remaining 238,250 (99.4%). Our main configuration uses only the skeleton tier; Section 4.4 shows the supporting tier adds nothing and zeroing the prunable tier is harmful.

Because raw signal magnitudes differ across model families, we also use a percentile rule: each signal is thresholded at its P-th percentile and the same consensus applied. On Gemma-2-2B, P=99 and P=99.5 select 137 and 51 neurons and both retain 99.0% (Section 4.3); on Llama-3-8B, P=99 selects 2,133 (0.46%), the skeleton used in Section 5.

3.3 Mixed-Precision Quantization

We use per-group round-to-nearest (RTN) quantization with group size 128 over each full weight matrix, then restore protected neurons from the original weights, so the grouping is identical regardless of which neurons are protected. Attention is quantized uniformly at 8-bit in the main configuration, since attention precision is the dominant factor on Gemma-2-2B (Section 4.4) and 208 heads are too few to tier. We write c3+a8 for skeleton at 16-bit, all other MLP weights at 3-bit, and attention at 8-bit.

4 Experiments

4.1 Setup

We fine-tune Gemma-2-2B (Gemma Team et al., 2024) on TinySQL (Harrasse et al., 2025) (Appendix A) and use the 105 evaluation samples it answers correctly, spanning five complexity levels, as the retention test set; signals are computed on a

Method	Bits	Ratio	Ret.%
Baseline 16-bit	16.00	1.00×	100.0
Uniform 4-bit	4.00	4.00×	75.2
Uniform 3-bit	3.00	5.33×	87.6
GPTQ uniform 4-bit	4.00	4.00×	96.2
GPTQ uniform 3-bit	3.00	5.33×	77.1
MLP@4 + attn@8, no skeleton	4.73	3.38×	100.0
CircuitTier c4+a8	4.78	3.35×	99.0
MLP@3 + attn@8, no skeleton	3.91	4.09×	98.1
CircuitTier c3+a8	**3.97**	**4.03×**	**100.0**
GPTQ MLP@3 + attn@8 (± skel.)	3.97	4.03×	81.9
(b) Skeleton selection at c3+a8 (1,295 neurons @16)			
None			98.1
Random (5 seeds)			98.7±0.5
TaCQ vulnerability top-1,295			98.1
Percentile P=99 (137 neurons)			99.0
Six-signal tier rule			100.0

Table 2: (a) Compression and (b) skeleton selection on Gemma-2-2B (n=105). cb+ab': skeleton at 16-bit, other MLP weights at b-bit, attention at b'-bit. GPTQ uses damping 0.01. Rows of (b) differ only in which neurons are restored.

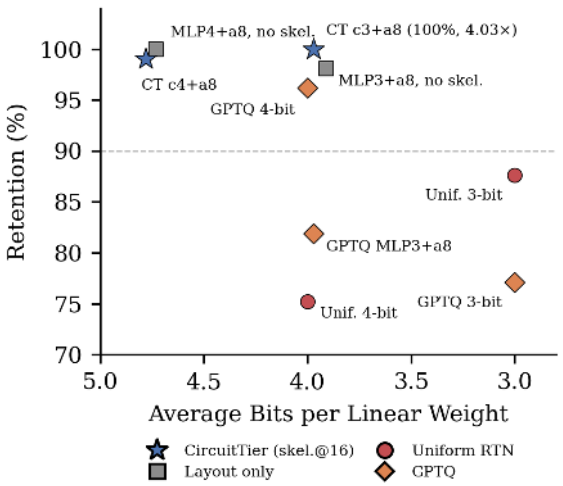


Figure 2: Compression and retention on Gemma-2-2B. Squares: no neuron protection; stars: with the 0.54% skeleton. Points correspond to rows of Table 2(a).

disjoint 500-sample analysis split. *Retention* is the percentage of these 105 preserved after compression, by exact match of the first generated line under greedy decoding (95% intervals in Appendix E). Compression ratio is $16/\bar{b}$ over linear weights, excluding embeddings and norms.

All Gemma and Llama results in this paper use the same quantizer and evaluation; the sole exception is the earlier protocol of Appendices D and L, which is marked wherever it appears.

4.2 Main Results

Uniform 4-bit loses a quarter of correct outputs (Table 2a). Moving attention to 8-bit and MLPs to 4-bit recovers all of them at 4.73 bits without protecting any neuron. At 3-bit MLPs with 8-bit attention, retention is 98.1% with no skeleton and 100% with the 0.54% skeleton at 3.97 bits (4.03×), against 75.2% for uniform 4-bit at 4.00 bits. GPTQ helps uniform 4-bit (96.2%) but reverses at 3-bit: 77.1% uniform and 81.9% at c3+a8, with or without the skeleton.

4.3 Skeleton Selection

Table 2(b) fixes the layout at c3+a8 and varies only which 1,295 neurons are restored. Every choice, including none, lands within two samples of the six-signal rule, which is noise at n=105. The tier-rule skeleton is never worse and costs 0.06 bits per weight, so we keep it as a safeguard, but on this model selection cannot drive retention: unprotected 3-bit MLPs already retain 98.1%, leaving nothing to recover. Section 5 asks the same question where uniform quantization does leave headroom.

4.4 Key Findings

Layout dominates. With 4-bit attention, retention is 75.2% (uniform) and 77.1% (with skeleton); with 8-bit attention, every MLP bit-width from 3 to 4 retains 98 to 100%. WikiText-2 perplexity follows suit (Appendix K): MLP@4 with 8-bit attention degrades it by 1.05× against 1.13× for uniform 4-bit, and c3+a8 by 1.47×. The interaction is not monotone in bit-width: with 4-bit attention, MLPs at 3-bit retain 95.2% while MLPs at 4-bit retain 75.2% (uniform 4-bit), and uniform 3-bit (87.6%) likewise exceeds uniform 4-bit. We do not have an explanation; the practical consequence is that uniform 4-bit is a weak baseline on this model, and layouts should also be compared against uniform 3-bit.

The prunable tier is harmful. Zeroing the 48 neurons all six signals rank lowest drops c4+a8 from 99.0% to 91.4%, six of the nine losses in the simplest level. Neurons every signal calls unimportant are not safely removable.

GPTQ at 3-bit. GPTQ improves uniform 4-bit from 75.2% to 96.2% but degrades every 3-bit layout: uniform 3-bit from 87.6% to 77.1%, c3+a8 from 100% to 81.9% with and without the skeleton. Restored skeleton columns receive a mean $|\Delta w|$ of 0.00029 from error redistribution against 0.00074 for quantized columns, so the cause is not contamination of protected weights. GPTQ is also sensitive

Restored to 16-bit (2,133 neurons)	Ret.%
None (uniform 4-bit)	77.1
Random, 3 seeds	77.0±0.1
Weight Δ	77.1
Activation Δ	77.5
Edge	76.8
EAP	90.6
Magnitude	91.4
Gradient	91.9
Six-signal consensus	**91.5**

Table 3: Llama-3-8B, n=894. Every row is uniform 4-bit per-group RTN with 2,133 MLP neurons restored to 16-bit; single signals restore their top-2,133. Intervals are no wider than 3 points (Appendix E).

to Hessian conditioning: uniform 4-bit gives 96.2% at damping 0.01, 88.6% at 0.1, and 79.0% with the same code on a different GPU (Appendix F). We treat the 3-bit degradation as robust and the 4-bit numbers as indicative.

5 Does Selection Matter? Llama-3-8B on Cypher

Because unprotected low-bit MLPs on Gemma-2-2B retain nearly everything, no skeleton can show an effect there. We repeat the selection experiment on a model where uniform quantization leaves headroom.

Setup. We fine-tune Meta-Llama-3-8B (Grattafiori et al., 2024) on Cypher generation with the Neo4j Text2Cypher 2024 dataset (Ozsoy et al., 2024) (39,554 train, 4,833 test; Appendix A), reaching 42.27% exact match and 2,043 baseline-correct samples. We evaluate 1,000 of these by index; batched decoding reproduces 894 at 16-bit, and retention is relative to those 894. The skeleton is the 99th-percentile consensus of Section 3.2: 2,133 neurons (0.46%). The layout is uniform 4-bit on all projections with only the skeleton restored (4.05 bits, 3.96×), so rows differ only in which neurons are restored.

Results. Here the identity of the protected neurons matters (Table 3). Uniform 4-bit retains 77.1%; a random 0.46% changes nothing (77.0±0.1); the six-signal skeleton recovers 14.4 points to 91.5%.The 95% intervals do not overlap: 74.2 to 79.7 for no skeleton against 89.5 to 93.2 for the consensus skeleton (Appendix E). The gain falls on the harder classes: retained counts rise from 124 to 165 of 177 on CY3, 66 to 93 of 113 on CY4, and 142 to 183 of 211 on CY5, while CY1 and CY2 are near ceiling throughout. The single signals split cleanly: EAP, magnitude, and gradient each select an effective skeleton (90.6 to 91.9%); weight delta, activation delta, and edge are indistinguishable from random. The consensus matches the best single signal without exceeding it. Its value is robustness: on Gemma-2-2B weight delta was among the strongest signals and magnitude among the weakest (Appendix D), so which signals work is not predictable, and consensus lands at the effective level on both models. Effective skeletons are also not unique: the gradient skeleton shares only 623 of 2,133 neurons with the consensus yet performs the same. Beyond this layout, no configuration between 4.6× and 9.3× retains more than 73% (Appendix L).

Recipe. The two models together suggest a procedure. First, quantize uniformly at the target bit-width and measure retention; this is one evaluation and tells you the headroom. If retention is already near the 16-bit baseline, as on Gemma-2-2B, spend the remaining budget on attention precision and stop. If it is not, as on Llama-3-8B, extract the six signals, select the skeleton by percentile consensus, restore it to 16-bit, and re-evaluate. On Llama-3-8B the second step costs 0.05 bits per weight and recovers 14.4 of the 23 points lost. Nothing in the procedure is specific to either model or task.

6 Conclusion

CircuitTier selects a sparse skeleton of task-critical neurons from six interpretability signals for mixed-precision compression. On Gemma-2-2B the 0.54% skeleton with 3-bit MLPs and 8-bit attention retains 100% at 4.03×, though on this model the layout does the work and the skeleton is a free safeguard. On Llama-3-8B, where uniform 4-bit loses 23 points, the skeleton recovers 14.4 and a random one recovers none; three of six signals are effective, the effective set differs between models, and consensus is reliable where single signals are not. GPTQ degrades 3-bit layouts by 16 to 18 points regardless of protection, and zeroing all-low neurons costs 8 points. Selection matters where uniform quantization leaves headroom, which one uniform 4-bit evaluation measures; we propose measuring headroom, selecting by consensus, and protecting as a recipe for larger and harder-to-quantize models.

Limitations

Scope. We evaluate two models from two families on two formal query languages. Larger scales, natural-language tasks, parameter-efficient fine-tuning recipes, and standard text-to-SQL benchmarks such as Spider remain untested.

Evaluation scale. The Gemma retention set has 105 samples, so one sample is 0.95 points and differences of two points or less are not interpretable; per-complexity subgroups are smaller still. The Llama set (894) is more reliable, and all cross-selection claims rest on it.

Selection. The selection effect is demonstrated on one larger model at one layout. The consensus rule matches but does not exceed the best single signal, and which single signals are effective differs by model in a way we cannot yet predict. Absolute Llama retention differs between an earlier run and the unified protocol reported here because the quantizer and evaluation changed; all reported numbers use the unified protocol.

Thresholds. The absolute threshold τ=0.3 is insensitive on Gemma (skeletons of 0.14% to 1.70% retain 95 to 96% at c4+a8, Appendix G), and the percentile rule transfers to Llama without tuning, but we have not tested either beyond these two models.

General capability. The c3+a8 layout raises WikiText-2 perplexity by 1.47× against 1.13× for uniform 4-bit; the method targets dedicated single-task deployment.

Inference hardware. We report average bits per weight and do not measure latency. Three-bit weights lack native kernel support in current engines; the 4-bit layouts map to existing engines with the skeleton stored separately.

GPTQ sensitivity. GPTQ results depend on Hessian damping and platform numerics, so its 4-bit rows are indicative rather than definitive. AWQ was not evaluated.

Computational cost. Signal extraction for Gemma-2-2B took roughly two hours on a single 40 GB GPU, dominated by activation patching.

Ethics Statement

Model compression enables deployment on resource-constrained devices, broadening access to capable language models. However, reducing deployment barriers may also facilitate use of models without adequate safety review. We encourage practitioners to evaluate compressed models for harmful outputs before deployment, as compression may alter model behavior in ways not captured by task-specific retention metrics.

Acknowledgments

AI assistants (Claude, Anthropic) were used in preparing this paper for editing and restructuring prose and LaTeX, for drafting the evaluation and plotting code used in the camera-ready reruns, and for cross-checking reported numbers against result files. All experiments were designed and run by the author; every reported number was verified by the author against the released result files, and every reference was verified by the author against the original source. No text was generated without the author's review, and no AI system is credited as an author.

References

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. 2023. Towards automated circuit discovery for mechanistic interpretability. *Preprint*, arXiv:2304.14997.

Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. 2023. SpQR: A sparse-quantized representation for near-lossless llm weight compression. *Preprint*, arXiv:2306.03078.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael Mahoney, and Kurt Keutzer. 2019. HAWQ: Hessian aware quantization of neural networks with mixed-precision. *Preprint*, arXiv:1905.03696.

Nelson Elhage, Neel Nanda, Catherine Olsson, Tom Henighan, Nicholas Joseph, Ben Mann, Amanda Askell, Yuntao Bai, Anna Chen, Tom Conerly, Nova DasSarma, Dawn Drain, Deep Ganguli, Zac Hatfield-Dodds, Danny Hernandez, Andy Jones, Jackson Kernion, Liane Lovitt, Kamal Ndousse, and 6 others. 2021. A mathematical framework for transformer circuits. Transformer Circuits Thread.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. GPTQ: Accurate post-training quantization for generative pre-trained transformers. *Preprint*, arXiv:2210.17323.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak

Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, and 179 others. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The Llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Abir Harrasse, Philip Quirke, Clement Neo, Dhruv Nathawani, Luke Marks, and Amir Abdullah. 2025. TinySQL: A progressive text-to-sql dataset for mechanistic interpretability research. *Preprint*, arXiv:2503.12730.

Samyak Jain, Robert Kirk, Ekdeep Singh Lubana, Robert P. Dick, Hidenori Tanaka, Edward Grefenstette, Tim Rocktäschel, and David Scott Krueger. 2024. Mechanistically analyzing the effects of fine-tuning on procedurally defined tasks. *Preprint*, arXiv:2311.12786.

Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. 2024. SqueezeLLM: Dense-and-sparse quantization. *Preprint*, arXiv:2306.07629.

Changhun Lee, Jungyu Jin, Taesu Kim, Hyungjun Kim, and Eunhyeok Park. 2024. OWQ: Outlier-aware weight quantization for efficient fine-tuning and inference of large language models. *Preprint*, arXiv:2306.02272.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H. S. Torr. 2019. SNIP: Single-shot network pruning based on connection sensitivity. *Preprint*, arXiv:1810.02340.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. AWQ: Activation-aware weight quantization for llm compression and acceleration. *Preprint*, arXiv:2306.00978.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. LLM-pruner: On the structural pruning of large language models. *Preprint*, arXiv:2305.11627.

Xin Men, Mingyu Xu, Qingyu Zhang, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. 2024. ShortGPT: Layers in large language models are more redundant than you expect. *Preprint*, arXiv:2403.03853.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. 2020. Zoom in: An introduction to circuits. *Distill*, 5(3):e00024.001.

Makbule Gulcin Ozsoy, Leila Messallem, Jon Besga, and Gianandrea Minneci. 2024. Text2Cypher: Bridging natural language and graph databases. *Preprint*, arXiv:2412.10064.

Abhishek Panigrahi, Nikunj Saunshi, Haoyu Zhao, and Sanjeev Arora. 2023. Task-specific skill localization in fine-tuned language models. *Preprint*, arXiv:2302.06600.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. 2024. Fine-tuning enhances existing mechanisms: A case study on entity tracking. *Preprint*, arXiv:2402.14811.

Aaquib Syed, Can Rager, and Arthur Conmy. 2023. Attribution patching outperforms automated circuit discovery. *Preprint*, arXiv:2310.10348.

Hidenori Tanaka, Daniel Kunin, Daniel L. K. Yamins, and Surya Ganguli. 2020. Pruning neural networks without any data by iteratively conserving synaptic flow. *Preprint*, arXiv:2006.05467.

Mangadoddi Srikar Vardhan and Lekkala Sai Teja. 2026. Disentangling direction and magnitude in transformer representations: A double dissociation through l2-matched perturbation analysis. *Preprint*, arXiv:2602.11169.

Chaoqi Wang, Guodong Zhang, and Roger Grosse. 2020. Picking winning tickets before training by preserving gradient flow. *Preprint*, arXiv:2002.07376.

Hanqi Xiao, Yi-Lin Sung, Elias Stengel-Eskin, and Mohit Bansal. 2025. Task-circuit quantization: Leveraging knowledge localization and interpretability for compression. *Preprint*, arXiv:2504.07389.

A Experimental Details

Reproducibility. Code, notebooks, and all result files are available at `https://github.com/M-SRIKAR-VARDHAN/circuittier`. Every number in this paper corresponds to a JSON file in that repository, including per-sample correctness for the Llama-3-8B experiments. Signal arrays, tier maps, fine-tuned checkpoints, and calibration data are hosted at `https://huggingface.co/datasets/primal-sage/tinysql-compression-results` because of their size. The repository also contains an interactive 3D circuit explorer that renders the Gemma-2-2B skeleton with toggleable signal overlays.

Gemma-2-2B fine-tuning. We fine-tune all 2.6B parameters with 8-bit AdamW (learning rate 2×10^{-5}, effective batch size 16 via 2 sequences with 8 accumulation steps, cosine schedule with 10% warmup, weight decay 0.01, bf16). Training was stopped at step 5,500 (0.88 epoch of the 100,000-example training split) after the loss plateaued at 0.62; this took about 5.5 hours on a single A40. We chose full fine-tuning over parameter-efficient methods because two signals measure weight deltas between the base and fine-tuned models; adapter-based methods would concentrate all changes in a small set of injected parameters.

Gemma data splits. TinySQL provides SQL generation examples across five complexity levels (CS1 to CS5), from single-table SELECT queries to multi-table joins with aggregation. From each level we draw three disjoint splits: 20,000 training examples, 40 evaluation examples, and 100 analysis examples for signal extraction (500 in total). Of the 200 evaluation examples, the fine-tuned model answers 105 correctly (52.5%): CS1 = 28, CS2 = 25, CS3 = 23, CS4 = 15, CS5 = 14. These 105 form the retention test set.

Llama-3-8B fine-tuning. We fine-tune all 8.03B parameters of Meta-Llama-3-8B with 8-bit AdamW (learning rate 2×10^{-5}, linear schedule with 10% warmup, weight decay 0.01, bf16, maximum sequence length 512, schemas truncated to 800 characters) on the 39,554 training examples of the Neo4j Text2Cypher 2024 dataset, for 1,236 optimizer steps on a single A100-80GB. On the 4,833 test examples the model reaches 42.27% exact match, giving 2,043 baseline-correct samples. Examples are tagged CY1 to CY5 by Cypher feature complexity (number of relationship patterns, aggregation, ordering, and filtering clauses). We evaluate on the first 1,000 baseline-correct samples by test-set index.

Compression ratio. We report $16/\bar{b}$, where $\bar{b}$ is the average bits per linear-layer weight. Gemma-2-2B has 2,614,341,888 parameters, of which 2,024,275,968 (77.4%) are in linear projections: 1,656,225,792 in the MLP (gate, up, down) and 368,050,176 in attention (q, k, v, o). The remaining embeddings, norms, and language model head stay at 16-bit and are excluded, following GPTQ, AWQ, and TaCQ. For the c3+a8 layout, the MLP averages $3 + 13 \times 0.0054 = 3.07$ bits and attention 8 bits, giving $\bar{b} = 0.818 \times 3.07 + 0.182 \times 8 = 3.97$ and a ratio of 4.03×. Llama-3-8B has 6,979,321,856 linear parameters, 80.8% in the MLP; the 2,133-neuron skeleton at 16-bit over uniform 4-bit gives $\bar{b} = 4.05$ and 3.96×.

Evaluation metric. Retention is exact string match of the first generated line against the reference query, with greedy decoding. Semantically equivalent queries with different surface forms count as failures, so retention is conservative. Gemma prompts are truncated to 400 tokens with up to 100 new tokens; Llama prompts to 480 tokens with up to 150 new tokens, decoded in batches of 8 with left padding. Under batched decoding the 16-bit Llama model reproduces 894 of the 1,000 baseline-correct samples; Llama retention is reported relative to these 894, and every configuration is evaluated on the same set.

Quantization implementation. All quantization is asymmetric min-max round-to-nearest over groups of 128 consecutive weights in row-major order of the full matrix. For a group g with minimum $w_{\min}$ and maximum $w_{\max}$ and target bit-width b,

$$\hat{w}_i = \mathrm{round}\left(\frac{w_i - w_{\min}}{s}\right) s + w_{\min},$$

with $s = (w_{\max} - w_{\min})/(2^b - 1)$.

To protect a set of neurons, the full matrix is quantized at the compressible bit-width and the protected rows of the gate and up projections and columns of the down projection are then overwritten with the original 16-bit weights. Quantizing a gathered subset of rows or columns instead, as in our earlier implementation, changes the group boundaries in the down projection with every protection set and introduces variance unrelated to which neurons are protected (Appendix D). Attention projections are quantized as full matrices at their assigned bit-width.

Computational cost. For Gemma-2-2B, signal extraction was dominated by activation patching over 50 analysis samples; gradient saliency (20 samples), activation deltas (500 samples), and the weight-based signals each took under 30 minutes on an A40, and the whole extraction roughly two hours. Quantizing and evaluating one layout takes about 3 minutes on an RTX PRO 4500; GPTQ adds about 2.5 minutes per layout after an 18-minute Hessian collection over 128 calibration samples.

B Signal Definitions

We provide full definitions of all six signals, expanding on the summary in Table 1. All signals are computed on the analysis split (Appendix A); the number of samples used differs by signal and is stated for each.

(1) Activation patching (EAP). Following Syed et al. (2023), we measure each neuron's causal importance by zero-ablating it and recording the change in the logit of the correct next token. For neuron n and input x,

$$\mathrm{EAP}(n) = \frac{1}{|X|} \sum_{x \in X} \big| y(x) - y(x \mid a_n \leftarrow 0) \big|,$$

where $y(x)$ is the target-token logit and $a_n \leftarrow 0$ denotes zeroing the neuron's activation at all positions. Because this requires one forward pass per neuron, it is computed on a balanced subset of 50 analysis samples (10 per complexity level) and is the most expensive signal.

(2) Gradient saliency. We backpropagate the language-modeling loss on the target query and accumulate absolute gradients over each neuron's weights. For neuron n with weight rows $\mathbf{w}_n^{\text{gate}}$, $\mathbf{w}_n^{\text{up}}$ and weight column $\mathbf{w}_n^{\text{down}}$,

$$\mathrm{Grad}(n) = \sum_{p \in \{\text{gate, up, down}\}} \mathrm{mean}\big(\big| \nabla_{\mathbf{w}_n^p} \mathcal{L} \big| \big),$$

accumulated over 20 analysis samples. Neurons with large gradients are those whose weight perturbation most affects the loss.

(3) Weight magnitude. The mean absolute value of each neuron's weights, summed across the three projections:

$$\mathrm{Mag}(n) = \sum_{p \in \{\text{gate, up, down}\}} \mathrm{mean}(|\mathbf{w}_n^p|).$$

This signal requires no data or forward passes.

(4) Weight delta. The mean absolute change in each neuron's weights between the base and fine-tuned checkpoints, summed across projections:

$$\mathrm{W}\Delta(n) = \sum_{p \in \{\text{gate, up, down}\}} \mathrm{mean}\big(\big| \mathbf{w}_{n,p}^{\text{ft}} - \mathbf{w}_{n,p}^{\text{base}} \big| \big).$$

Neurons with near-zero weight delta were not updated during fine-tuning. This signal requires the base model checkpoint.

(5) Activation delta. For each complexity level c, we record the maximum activation of neuron n over all tokens of that level's 100 analysis samples in the fine-tuned and base models, and average the absolute difference over the five levels:

$$\mathrm{Act}\Delta(n) = \frac{1}{5} \sum_{c=1}^{5} \left| \max_{x \in X_c} a_n^{\text{ft}}(x) - \max_{x \in X_c} a_n^{\text{base}}(x) \right|.$$

This captures neurons whose behavior changed during fine-tuning even if their weights changed only slightly, for example through upstream changes propagating forward. It requires forward passes through both models on all 500 analysis samples.

(6) Edge importance. For each MLP neuron, we aggregate the absolute patching scores of all incoming attention-head-to-neuron edges obtained from the same patching runs as signal (1):

$$\mathrm{Edge}(n) = \sum_{\ell'=1}^{26} \sum_{h=1}^{8} \big| \mathrm{EAP}(\ell', h \to n) \big|,$$

where $\mathrm{EAP}(\ell', h \to n)$ is the score of the edge from attention head h in layer ℓ' to neuron n. The full tensor has shape $[26 \times 8 \times 26 \times 9216]$ (source layer, head, target layer, neuron). This signal measures how strongly each neuron receives attention-mediated information, capturing inter-layer connectivity that the other five per-neuron signals do not.

Normalization. Each signal is min-max normalized to $[0, 1]$ independently across all 239,616 MLP neurons:

$$\hat{s}_i(n) = \frac{s_i(n) - \min_m s_i(m)}{\max_m s_i(m) - \min_m s_i(m)}.$$

This ensures that signals with different raw scales contribute equally to the tier classification thresholds.

Signal distribution. Signal values are highly skewed across the neuron population. The median normalized value is 0.04, the 95th percentile is 0.18, and only a small tail exceeds the skeleton threshold τ=0.3. The choice of τ therefore falls in a region of low density, which is why the tier assignment is insensitive to small changes in τ (Appendix G).

C Signal Analysis

Figures 3 and 4 and Table 4 show why we describe the six signals as *complementary* rather than decorrelated. Magnitude and Weight Δ share 72% of

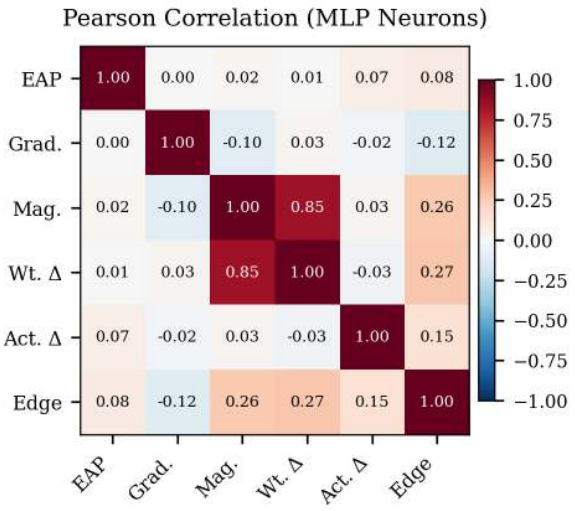


Figure 3: Pearson correlation across the six signals for all 239,616 MLP neurons of Gemma-2-2B. Only Magnitude and Weight Δ are strongly correlated (r=0.85); all other pairs are near-orthogonal ($|r| < 0.27$). Mean off-diagonal $|r| = 0.098$.

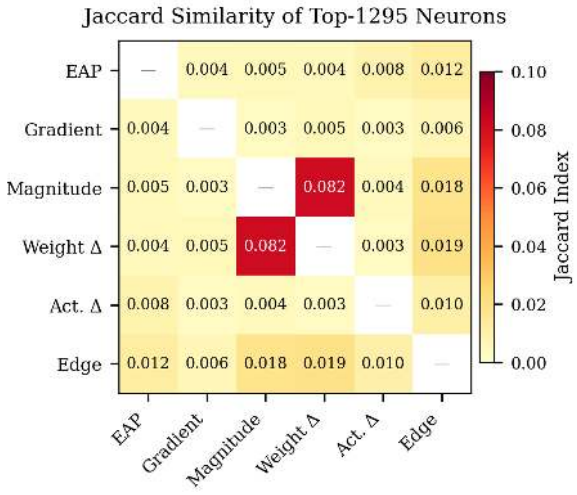


Figure 4: Jaccard similarity of the top-1,295 neurons selected by each signal. Despite the Pearson correlation between Magnitude and Weight Δ (r=0.85), even this pair has low top-k overlap (Jaccard = 0.082); all other pairs are below 0.02. Mean pairwise Jaccard = 0.012.

	EAP	**Grad**	**Mag**	**WΔ**	**ActΔ**	**Edge**
EAP		.000	.023	.007	.066	.080
Grad			−.104	.027	−.024	−.125
Mag				**.850**	.031	.260
WΔ					−.033	.269
ActΔ						.147
Edge						

Table 4: Pearson correlation between signals on Gemma-2-2B MLP neurons. Mean off-diagonal $|r| = 0.098$.

their variance globally (r=0.85), yet their top-k neuron sets overlap by only 8.2%; every other pair overlaps by under 2%. Selection by rank therefore draws different neurons from each signal even when their raw values are correlated.

The Magnitude and Weight Δ correlation is expected: neurons with large weights tend to receive large absolute updates because optimizer steps scale with gradients, and gradients are larger for high-magnitude weights. The rankings differ because Magnitude captures importance inherited from pretraining while Weight Δ captures task-specific adaptation: a neuron can have large magnitude and small delta (important for general language, untouched by fine-tuning) or small magnitude and large delta (recruited by fine-tuning).

Why six signals? Low overlap establishes that the signals identify different neurons; it does not by itself establish that combining them helps compression, and Sections 4.3 and 5 test that directly. On Gemma-2-2B every selection rule, including random, retains 98 to 100% at the main layout, so the question cannot be resolved there. On Llama-3-8B three signals (EAP, Gradient, Magnitude) select an effective skeleton and three (Weight Δ, Activation Δ, Edge) do no better than random, while on Gemma-2-2B at 4-bit Weight Δ was among the best single signals and Activation Δ the worst (Appendix D). Which signals are effective therefore depends on the model, and we have no way to predict it in advance. The consensus rule is our answer to that uncertainty: it matched the best single signal on both models without knowing which one that was. We make no claim that six is the right number; each signal we use is cheap relative to one forward pass of the base model, and the rule tolerates inert signals because a neuron must exceed the threshold on at least three.

Why not tier attention heads? We quantize all attention heads uniformly rather than tiering them like MLP neurons, for two reasons. First, with 208 heads against 239,616 neurons, cross-signal consensus has little resolution: a threshold applied to 208 components yields very coarse tiers. Second, attention precision is not sparse in its effect on Gemma-2-2B: with 8-bit attention, every MLP layout from 3-bit to 4-bit retains 98 to 100%, while dropping attention to 4-bit with the same skeleton reduces retention to 77.1% (Section 4.4), which indicates that attention heads are broadly rather than sparsely important for SQL generation. All 8 heads of Layer 0 are classified as skeleton by our signals,

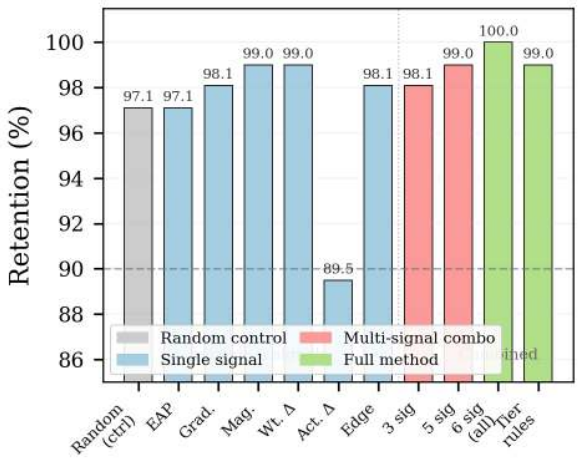


Figure 5: Single-signal and combined selection on Gemma-2-2B at the c4+a8 layout under the gathered-subset quantizer (Table 5). Each bar selects the top-1,295 neurons by one signal (blue), by a combination (pink), or by the full tier rule (green); gray is one random seed. All bars lie between 89.5 and 100%.

Selection method	Ret.%	Kept
Random (5 seeds)	98.7±1.1	102 to 105
EAP	97.1	102/105
Gradient	98.1	103/105
Magnitude	99.0	104/105
Weight Δ	99.0	104/105
Activation Δ	89.5	94/105
Edge importance	98.1	103/105
EAP + Gradient	98.1	103/105
EAP + Weight Δ	96.2	101/105
EAP + Activation Δ	93.3	98/105
Gradient + Magnitude	97.1	102/105
Magnitude + Weight Δ	98.1	103/105
EAP + Gradient + Magnitude	98.1	103/105
Four signals (+ Weight Δ)	95.2	100/105
Five signals (+ Activation Δ)	99.0	104/105
Six-signal top-k union	100.0	105/105
Full tier rule	99.0	104/105

Table 5: Signal ablation on Gemma-2-2B. Every row protects exactly 1,295 neurons at 16-bit with all other MLP weights at 4-bit and attention at 8-bit, using the gathered-subset quantizer of the original submission (Appendix E). Combinations select the top-1,295 by summed normalized score.

consistent with the first layer's attention handling token routing that the whole task depends on.

D Signal Ablation on Gemma-2-2B

Figure 5 and Table 5 give the per-signal picture on Gemma-2-2B. Each row selects the top-1,295 neurons (the size of the tier-rule skeleton) by the stated signal or combination and protects them at 16-bit with every other MLP neuron at 4-bit and attention at 8-bit. These runs use the gathered-subset quantizer of the original submission, which adds grouping variance in the down projection (Appendix E); we retain them because they are the only per-signal measurements on this model, and the pattern they show is corroborated on Llama-3-8B under the corrected quantizer (Section 5).

Three observations survive the noise. First, at this layout selection barely matters: random protection over five seeds averages 98.7%, and twelve of the sixteen rows are within two samples of it. Second, the one clear outlier is Activation Δ alone at 89.5%, eight to ten points below the other single signals; activation deltas are noisy because upstream weight changes propagate to downstream neurons that are not themselves task-critical. Third, Weight Δ alone is among the best signals here (99.0%), whereas on Llama-3-8B it is indistinguishable from random (Table 3). The effective single signal is therefore model-dependent, which is the empirical basis for using a consensus rule rather than any one signal. The combinations do not improve monotonically with the number of signals (four signals score below three), and the six-signal union at 100.0% is one sample above the tier rule at 99.0%; neither difference is interpretable at n=105.

Under the corrected quantizer at the main c3+a8 layout, Table 2(b) repeats the comparison for none, random, a TaCQ-style vulnerability selection, the percentile rule, and the tier rule; all lie between 98.1 and 100.0%.

E Quantizer Sensitivity and Confidence Intervals

The gathered-subset artifact. The original submission quantized protected and unprotected neurons by gathering the corresponding rows of the gate and up projections and columns of the down projection into a sub-matrix and quantizing that sub-matrix per-group. For the down projection the number of unprotected columns is not a multiple of the group size, so the group boundaries shift with every different protection set. At 3-bit this shifts the min-max grid of most groups and produces retention differences that depend on which neurons were selected but not on their importance. Table 6 shows the effect: under the gathered quantizer, five random skeletons at c3+a8 span 69.5 to 99.0%, and one seed at c3+a4 falls to 60.0%. Under the corrected quantizer of Section 3.3, which quantizes

Layout	Selection	Gathered	Aligned
c3+a8	None	n/a	98.1
c3+a8	Random (5 seeds)	91.6±12.4	98.7±0.5
c3+a8	Tier-rule skeleton	97.1	100.0
c3+a4	None	n/a	95.2
c3+a4	Random (5 / 3 seeds)	84.4±14.3	94.9±0.5
c3+a4	Tier-rule skeleton	94.3	95.2

Table 6: Retention on Gemma-2-2B (n=105) under the gathered-subset quantizer of the original submission and the aligned protect-by-restore quantizer used throughout this version. Random rows report mean and standard deviation over seeds (five gathered; five and three aligned for c3+a8 and c3+a4). Selection-dependent variance disappears once the grouping is held fixed.

the full matrix once and restores protected neurons afterwards, the same five seeds span 98.1 to 99.0% and are indistinguishable from protecting nothing. The reliability advantage reported in the original submission was this artifact, and all results in the present version use the corrected quantizer.

Determinism. Given the same checkpoints, analysis samples, and hardware, signal values, tier assignments, quantized weights, and greedy decoding are deterministic: re-running the c4+a8 configuration and the GPTQ uniform 4-bit configuration on the same machine reproduced 101 of 105 in both cases. Determinism does not extend across platforms for GPTQ, whose Hessian inverse is numerically fragile (Appendix F); round-to-nearest results reproduced exactly between the February and September runs on different GPUs.

Confidence intervals. Table 7 reports 95% Wilson score intervals for the configurations in Tables 2 and 3. On Gemma-2-2B, the c3+a8 layout is separated from uniform 4-bit and from every GPTQ 3-bit configuration at the interval level, but its interval overlaps that of the same layout with no skeleton, which is why Section 4.3 makes no selection claim on this model. On Llama-3-8B, the skeleton and no-skeleton intervals do not overlap.

F GPTQ Analysis

Implementation. We use the standard GPTQ procedure (Frantar et al., 2023): Hessians $H = X^\top X$ are accumulated over 128 analysis samples for every linear projection, damped by 0.01 times the mean diagonal, and inverted by Cholesky decomposition; columns are quantized sequentially in blocks of 128 with error propagated to later

Configuration	Ret.%	95% CI
Gemma-2-2B (n=105)		
Uniform 4-bit	75.2	66.2 to 82.5
Uniform 3-bit	87.6	80.0 to 92.6
GPTQ uniform 4-bit	96.2	90.6 to 98.5
GPTQ uniform 3-bit	77.1	68.2 to 84.1
MLP@4 + attn@8, no skeleton	100.0	96.5 to 100.0
CircuitTier c4+a8	99.0	94.8 to 99.8
MLP@3 + attn@8, no skeleton	98.1	93.3 to 99.5
CircuitTier c3+a8	100.0	96.5 to 100.0
GPTQ MLP@3 + attn@8	81.9	73.5 to 88.1
Four-tier c4+a8 (with pruning)	91.4	84.5 to 95.4
Llama-3-8B (n=894)		
Uniform 4-bit	77.1	74.2 to 79.7
Gradient skeleton	91.9	89.9 to 93.5
Six-signal skeleton	91.5	89.5 to 93.2

Table 7: 95% Wilson score intervals for retention.

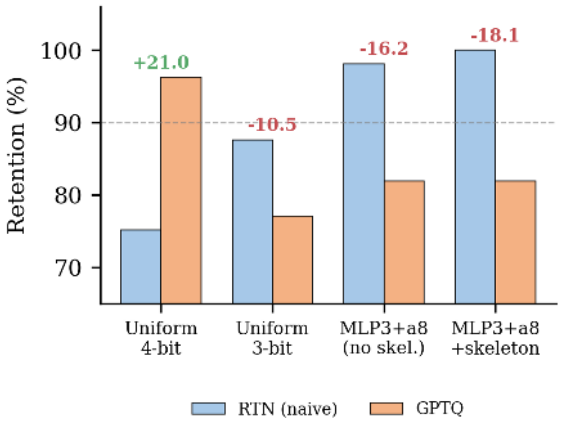


Figure 6: Round-to-nearest versus GPTQ on Gemma-2-2B (n=105), same per-group grid in both. GPTQ helps uniform 4-bit by 21 points and hurts every 3-bit layout by 10 to 18 points, with and without the skeleton.

columns through the inverse Hessian. Each column is quantized with the same asymmetric min-max grid over groups of 128 weights as our RTN baseline, so the two methods differ only in error compensation. For protected neurons we quantize the full matrix and restore their weights afterwards, exactly as for RTN.

Results. Figure 6 shows that GPTQ raises uniform 4-bit retention from 75.2% to 96.2% but lowers uniform 3-bit from 87.6% to 77.1%, MLP@3 with 8-bit attention from 98.1% to 81.9%, and the same layout with the skeleton restored from 100% to 81.9%. The last two are identical to the sample, so at 3-bit the degradation is independent of neuron protection.

Contamination is not the mechanism. The original submission attributed GPTQ's degradation of

Neuron group	All projections	Down only
Skeleton (1,295)	0.00010	0.00029
Non-skeleton (238,321)	0.00078	0.00074

Table 8: Mean $|\Delta w|$ per weight after GPTQ at the c4+a8 layout under the sub-matrix implementation, by neuron group. Skeleton gate and up rows are not quantized, which lowers the all-projection mean; the down-only column is the relevant comparison.

protected layouts to error being redistributed into the 16-bit skeleton columns of the down projection. We measured this directly (Table 8) under the submission's sub-matrix implementation, in which skeleton columns are excluded from quantization but do receive redistributed error. Skeleton columns of the down projection are perturbed by a mean $|\Delta w|$ of 0.00029, against 0.00074 for quantized columns; skeleton rows of the gate and up projections are untouched. Redistribution into protected weights is real but less than half the size of ordinary quantization error, and the degradation persists when protected weights are fully restored, so contamination cannot explain it.

Sensitivity. GPTQ's inverse-Hessian step is numerically fragile in this setting: Hessian condition numbers reach 10^5 for MLP projections and 10^{13} for attention projections. Table 9 shows the consequences. Uniform 4-bit gives 96.2% at damping 0.01 and 88.6% at damping 0.1 on the same machine, and 79.0% with identical code, calibration set, and damping on a different GPU and framework build in our original submission; the sub-matrix c4+a8 layout moved from 89.5% to 98.1% across the same change. Round-to-nearest results reproduced exactly across both machines. We therefore treat the sign of GPTQ's effect at 3-bit as robust (it degraded 3-bit layouts in every run on both machines) and its 4-bit magnitude as indicative.

Interpretation. We do not have a verified mechanism for the 3-bit degradation. One hypothesis consistent with the data is that at 3-bit the per-column residual is large relative to the group's min-max range, so the sequential corrections pushed into later columns frequently move them across grid boundaries, compounding rather than cancelling error, whereas at 4-bit the corrections are small enough to act as intended. Testing this requires per-column error traces and is left to future work. The practical conclusion is narrower than the original submission's: in our setting, GPTQ should not be combined with 3-bit min-max grouping, where plain round-to-nearest is 16 to 18 points better, and GPTQ's benefit at 4-bit should be verified on the deployment hardware.

GPTQ configuration	Feb.	Sept.
Uniform 4-bit, damp 0.01	79.0	96.2
Uniform 4-bit, damp 0.1	n/a	88.6
Uniform 3-bit, damp 0.01	81.9	77.1
Sub-matrix c4+a8, damp 0.01	89.5	98.1
RTN uniform 4-bit	75.2	75.2
RTN c4+a8 (sub-matrix)	96.2	96.2

Table 9: GPTQ retention on Gemma-2-2B (n=105) across damping and hardware, with the same code and calibration set. Feb. runs used an A40, Sept. runs an RTX PRO 4500. RTN rows show the reproducibility of the non-GPTQ pipeline. Repeated runs on one machine are identical.

Rule	Skeleton	%	Ret.%
Absolute τ, four-tier c4+a8, gathered quantizer			
τ=0.25	4,074	1.70	95.2
τ=0.30	1,295	0.54	96.2
τ=0.35	339	0.14	96.2
Percentile P, c3+a8, aligned quantizer			
P=99	137	0.06	99.0
P=99.5	51	0.02	99.0
τ=0.30 (reference)	1,295	0.54	100.0
None (reference)	0	0.00	98.1

Table 10: Threshold sensitivity on Gemma-2-2B (n=105). The absolute sweep uses the four-tier layout and quantizer of the original submission; the percentile rows use the main layout of this version.

G Threshold Sensitivity

Table 10 varies the skeleton threshold over a twelvefold range of skeleton size, from 339 to 4,074 neurons, and retention moves by one sample. Under the percentile rule at the main layout, skeletons of 51 and 137 neurons retain 99.0%, one sample below the 1,295-neuron skeleton and one above no skeleton. On this model the threshold is not a sensitive parameter because, as Section 4.3 shows, selection itself is not: there is nothing in the interval for the threshold to get wrong. The prunable threshold τ'=0.1 identifies 48 neurons on which all six signals are low; zeroing them costs eight samples at c4+a8 (Appendix H), so we do not use it.

Layout	c4+a8	c3+a8
No skeleton	100.0	98.1
Two-tier (skeleton@16)	99.0	100.0
+ supporting@8	99.0	n/a
+ prunable@0 (four-tier)	91.4	99.0

Table 11: Tier ablation on Gemma-2-2B (n=105), aligned quantizer. Rows add tiers cumulatively: the two-tier layout restores the 1,295 skeleton neurons; the third row additionally holds the 23 supporting neurons at 8-bit; the fourth additionally zeroes the 48 prunable neurons.

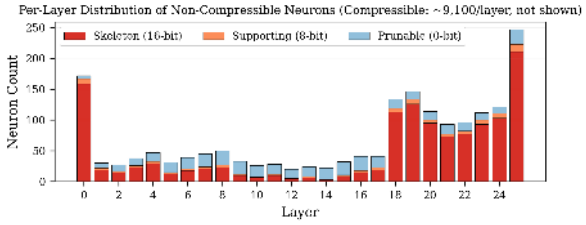


Figure 7: Per-layer distribution of skeleton, supporting, and prunable neurons on Gemma-2-2B. Skeleton neurons (red) concentrate in Layer 0 and Layers 18 to 25. The compressible majority (about 9,100 per layer) is omitted.

H Tier Ablation

Table 11 adds the tiers of Section 3.2 one at a time. The supporting tier changes nothing: with and without it the c4+a8 layout retains 99.0%, and merging supporting neurons into the compressible tier under the gathered quantizer gave 97.1% against 96.2% for the full four-tier layout. The prunable tier is harmful: zeroing 48 neurons drops c4+a8 from 99.0% to 91.4%, with six of the nine lost samples in CS1, while at c3+a8 the same zeroing costs one sample. The recommended layout is therefore two tiers, skeleton and everything else.

VulnSplit. The original submission subdivided the compressible tier by a TaCQ-style per-neuron vulnerability score, assigning the least vulnerable 50% or 75% of compressible neurons to 3-bit and the rest to 4-bit. Under the aligned quantizer both splits retain 95.2% (at 4.37 and 4.17 bits), below the plain c3+a8 layout at 3.97 bits (100%) and below MLP@4 with attention at 8-bit (100% at 4.73 bits). The subdivision is dominated on both axes and is not part of the method.

I Layer Distribution

Skeleton neurons are distributed bimodally across layers (Figure 7): Layer 0 holds about 160 and Layers 18 to 25 between roughly 90 and 245 each, while the middle layers (10 to 16) hold few. All 8 attention heads of Layer 0 are classified as skeleton.

This pattern is interpretable for SQL generation. The early layers, especially Layer 0, perform token processing and schema recognition, identifying table names, column names, and SQL keywords in the prompt; the model must parse the input before it can generate. The final layers convert the internal representation into syntactically correct SQL tokens. The middle layers, which handle general language processing, were largely unchanged by fine-tuning. On Llama-3-8B the skeleton concentrates in the last five layers (27 to 31); the late-layer component replicates, the Layer 0 component does not.

Configuration	CS1	CS2	CS3	CS4	CS5	All
Baseline	100	100	100	100	100	100.0
Uniform 4-bit	57	84	87	67	86	75.2
Uniform 3-bit	100	84	83	87	79	87.6
GPTQ uniform 3-bit	100	64	74	73	64	77.1
MLP@3 + attn@8, no skeleton	100	92	100	100	100	98.1
CircuitTier c3+a8	100	100	100	100	100	100.0
GPTQ MLP@3 + attn@8	100	80	83	67	64	81.9
Four-tier c4+a8 (with pruning)	79	100	96	93	93	91.4

Table 12: Retention (%) by SQL complexity level on Gemma-2-2B (CS1 simplest, CS5 most complex). Baseline-correct samples: CS1 = 28, CS2 = 25, CS3 = 23, CS4 = 15, CS5 = 14; CS4 and CS5 are small and per-level numbers should be read with caution.

J Per-Complexity Results

Two patterns stand out in Table 12. Uniform 4-bit degrades CS1, the simplest queries, most (57%), and zeroing the prunable tier at c4+a8 also loses its samples mainly in CS1 (79%): the simplest queries appear to depend on a small, precise set of weights whose disruption is not compensated elsewhere. GPTQ at 3-bit, by contrast, keeps CS1 intact and loses samples across CS2 to CS5. The c3+a8 layout retains every sample at every level.

K General Capability

Table 13 reports general-domain perplexity on WikiText-2 (Merity et al., 2016). The fine-tuned model starts at 32.3, well above the base model, because SQL fine-tuning has already traded general capability for the task. The 4-bit layouts with 8-bit attention degrade perplexity less than uniform 4-bit (1.05× against 1.13×), so at 4-bit the layout costs nothing in general capability. The 3-bit MLP layouts degrade it by 1.47×; this is the price of the

Configuration	PPL	Ratio
Fine-tuned 16-bit	32.3	1.00×
Uniform 4-bit	36.5	1.13×
MLP@4 + attn@8, no skeleton	33.8	1.05×
CircuitTier c4+a8	33.8	1.05×
MLP@3 + attn@8, no skeleton	47.4	1.47×
CircuitTier c3+a8	47.6	1.47×

Table 13: WikiText-2 test perplexity of the fine-tuned Gemma-2-2B under each layout, computed over 2,048-token windows with a BOS token prepended to each window.

skel / supp / MLP / attn, g	Ratio	Ret.%
16 / 8 / 4 / 4, 128	3.84×	98
16 / 4 / 3 / 3, 128	5.20×	59
16 / 4 / 3 / 3, 64	5.20×	68
8 / 4 / 3 / 3, 64	5.25×	73
16 / 8 / 2 / 4, 32	6.13×	27
16 / 4 / 2 / 3, 64	6.95×	10
16 / 8 / 1 / 4, 32	8.71×	0

Table 14: Representative rows of a 22-configuration sweep on Llama-3-8B (n=100, earlier protocol, retention relative to the 16-bit model). Columns give the bit-widths of skeleton, supporting, and compressible MLP neurons and of attention, and the quantization group size. No configuration above 4.6× retains more than 73%.

4.03× headline configuration, and it is the same with or without the skeleton. For multi-purpose deployment, MLP@4 with 8-bit attention at 3.38× is the better operating point.

L Compression Wall on Llama-3-8B

An earlier sweep of 22 layouts between 4.6× and 9.3× on Llama-3-8B (n=100, under the original quantizer and evaluation) found a hard wall near 5×: every layout with 2-bit MLPs collapses regardless of skeleton size, supporting precision, or group size, and the best 3-bit-MLP layout retains 73% at 5.25× (Table 14). Smaller quantization groups help at the margin (59% to 68% from g=128 to g=64) but do not move the wall. Quantization-aware fine-tuning for 100 steps did not help even at 4-bit (uniform 56% to 54%, skeleton layout 96% to 93%, n=100).

M Future Directions

Scale and generalization. Validating the headroom hypothesis on a third model family and across a 7B to 70B range on one task is the most important next step: measure uniform 4-bit headroom first, then skeleton recovery. Diverse tasks (summarization, code generation, mathematical reasoning) would test whether the effective signals are a property of the task, the model, or the fine-tune. We no longer expect larger models to have sparser skeletons as a rule; Llama-3-8B's skeleton (0.46%) is close to Gemma-2-2B's (0.54%) under different rules, and skeleton size did not predict retention on either.

Why the effective signals differ. Weight delta selects an effective skeleton on Gemma-2-2B and not on Llama-3-8B, while EAP, gradient, and magnitude work on both. Understanding what makes a signal informative for a given fine-tune would let the consensus rule be replaced by a principled choice, or confirm that consensus is the right default.

Multi-task circuits. When a model is fine-tuned on multiple tasks, whether task skeletons overlap or are disjoint decides whether one protected set can serve several capabilities at once or whether each costs its own share of the precision budget. Mapping these interactions is both a compression and an interpretability question.

Continual fine-tuning. If a model is fine-tuned first on SQL and then on another task, does the second fine-tune overwrite the first skeleton? Re-extracting signals after each stage would track how the skeleton moves and whether earlier capabilities can be protected during later training.

Learned thresholds. On Gemma-2-2B the threshold was insensitive because selection was; on a model with headroom a validation-set search for the smallest skeleton that reaches a target retention would be informative, and the percentile formulation gives it a single scalar to search over.

GPTQ at low precision. The 3-bit degradation is robust across platforms and unexplained. Per-column error traces would test the grid-crossing hypothesis of Appendix F, and the result would say whether error compensation can be made to work below 4-bit with min-max grouping or should be avoided there.

Inference integration. Realizing the bit savings as latency requires mixed-precision kernels. The two-tier layout maps directly to existing 4-bit engines with the skeleton stored separately, which is the natural first deployment; 3-bit MLPs need kernel support that current engines lack.