DELTA-MOE: STREAMED COUPLED CORRECTIONS FOR COMPRESSED MOE EXPERTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Mixture-of-Experts (MoE) models activate few experts per token, but their expert banks consume substantial GPU memory. We study *recoverability* as a third axis for compression, alongside resident memory and uncorrected accuracy: how much lost computation can be restored under a limited correction budget? A compressed model with poor standalone accuracy can still provide an effective base if its errors admit sparse, input-dependent correction. This perspective makes expert merging a viable compression strategy when combined with selective recovery. We introduce Delta-MoE, which keeps compressed experts on the GPU and streams selected residual channels from host memory. Each correction couples the gate, up, and down projections of a SwiGLU channel and recomputes its nonlinear contribution. Our analysis characterizes recovery through residual error spectra and explains why improving an uncorrected merge need not improve its corrected counterpart. On Qwen3-30B-A3B, the fused implementation achieves 0.810 GSM8K accuracy at $16\times$ resident expert compression, with MATH-500 and IFEval point estimates similar to an $8\times$ quantized base, and Mixtral-8x7B also exhibits recovery from merging.

1 INTRODUCTION

Mixture-of-Experts (MoE) models decouple capacity from per-token computation: a router activates a small subset of experts, while all expert parameters contribute to the model's memory footprint (Shazeer et al., 2017; Fedus et al., 2022; Jiang et al., 2024; Dai et al., 2024; Yang et al., 2025). Qwen3-30B-A3B activates 8 of 128 experts per layer, yet conventional low-latency serving keeps its $\sim$58 GB expert bank in GPU memory, limiting space for KV cache and larger batches. Host offloading reduces this residency but incurs CPU–GPU transfers (Xue et al., 2025; Kamahori et al., 2025). Compressing the resident bank offers another way to save memory, at the cost of model accuracy.

With a limited transfer budget, the choice of compressor also depends on the structure of its error. A compressed expert can remain useful if a small, input-dependent correction recovers much of the lost accuracy. Quantization, pruning, and merging leave different opportunities for this correction. Accurate quantization leaves little error to recover (Chee et al., 2023; Tseng et al., 2024a; Egiazarian et al., 2024; Tseng et al., 2024b). Pruning removes experts (Lu et al., 2024; Muzio et al., 2024; Xie et al., 2024; Lasby et al., 2026), so correcting the surviving weights cannot restore the deleted computations. Merging retains a shared representation of every original expert (Li et al., 2026; Chen et al., 2025); although its uncorrected accuracy can be poor, the difference from each original expert can be recovered channel by channel.

Delta-MoE exploits this structure (Figure 1). For each original expert, it stores the original-minus-resident weights as coupled channel triplets $(\Delta W^1[j,:], \Delta W^3[j,:], \Delta W^2[:,j])$ in a host-memory NF4 bank. Resident activations select triplets for each routed batch, and the correction recomputes the selected channels through the SwiGLU nonlinearity. Before residual quantization, each selected channel reproduces its original contribution exactly. The resident expert supplies both the selection signal and the computation for unselected channels.

We describe the available correction with the *recovery spectrum* $\mathcal{R}_S(K)$: the best remaining output error after at most K channel corrections to resident base S. Minimizing uncorrected error need not minimize the corrected error. This observation motivates evaluating a compressor together with

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.

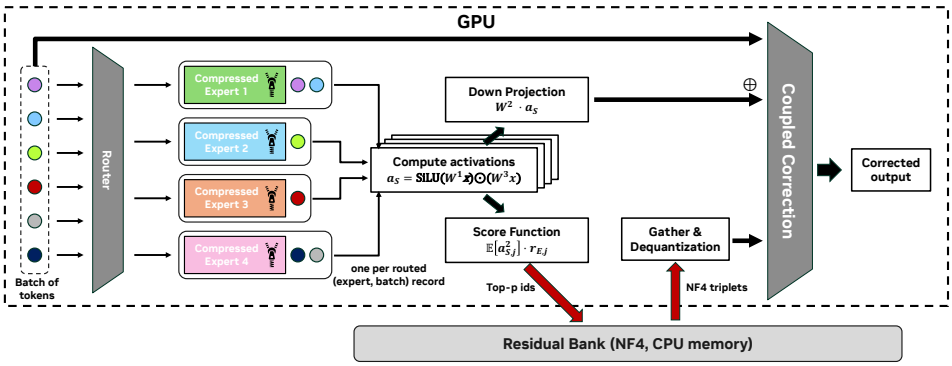

Figure 1: Delta-MoE streams coupled gate/up/down residuals from a CPU bank. Resident activations select channels for each routed (expert, batch) record. A consolidated per-layer fetch supplies the triplets, and nonlinear recomputation replaces the selected channel contributions. Unselected channels retain the resident computation.

its correction scheme; the spectrum is a diagnostic, rather than a general predictor of downstream accuracy.

The key insight is that a useful compressed base need not minimize every component of reconstruction error equally. In the merges we study, much of the residual energy concentrates in a small set of channels. Selective correction can recover expert-specific computation through these channels while the shared base supplies the remaining computation. Improving reconstruction on channels that correction already restores may therefore offer little additional benefit. The relevant target is the error left after the correction budget is spent. This makes expert merging a viable candidate for compression even when its standalone accuracy is poor, and motivates designing merged bases together with their recovery mechanism.

Our principal evidence is generated behavior. On Qwen3-30B-A3B, merging over QTIP reaches $16\times$ resident expert compression while retaining much of the uncompressed model's GSM8K accuracy. The fused implementation also preserves the MATH-500 and IFEval point estimates of an $8\times$ QTIP base at half its expert residency. Mixtral-8x7B confirms recovery from merging, although the deep-compression generation result is model-dependent.

Contributions. We develop a streamed correction algorithm that selects complete SwiGLU channels and recomputes their nonlinear contributions without loading complete experts. We demonstrate generation recovery for merged and quantized bases, with separate accounting for resident memory, host storage, and transferred bytes. We also study a layer-global allocator and recovery-spectrum diagnostics to characterize where selective correction helps. The evaluation distinguishes generated-answer accuracy from batch-conditioned likelihood scores (§3). Comparisons cover quantization (Frantar et al., 2023; Tseng et al., 2024b), compensation (Huang et al., 2025a), merging (Gu et al., 2025; Li et al., 2026; Chen et al., 2025), pruning (Lasby et al., 2026), and a projection-wise DecDEC-style control (Park et al., 2025).

2 Methodology

Problem statement. Given a resident expert-memory budget and a CPU–GPU transfer budget, we seek a compressed base and correction scheme that maximize accuracy. Let E be an original expert and S its resident approximation, obtained by merging, quantization, or both. All residuals are original minus resident weights.

2.1 Channelwise Residual Bank

A SwiGLU channel couples a gate row $W_E^1[j,:]$, an up row $W_E^3[j,:]$, and a down column $W_E^2[:,j]$. Its activation is $a_{E,j}(x) = \text{SiLU}(W_E^1[j,:]x)\,(W_E^3[j,:]x)$, and its output contribution is $W_E^2[:,j]a_{E,j}(x)$. We store the difference between E and S as a channel triplet:

$$\Delta W_E^1[j,:] = W_E^1[j,:] - W_S^1[j,:], \quad \Delta W_E^3[j,:] = W_E^3[j,:] - W_S^3[j,:], \quad \Delta W_E^2[:,j] = W_E^2[:,j] - W_S^2[:,j]. \tag{1}$$

The triplet contains all weights needed to recompute one channel through the gate and output projection. Summing these corrections gives

$$y_E(x) = y_S(x) + \sum_{j=1}^{d_{\text{ff}}} \left[W_S^2[:,j] \left(a_{E,j}(x) - a_{S,j}(x) \right) + \Delta W_E^2[:,j] a_{E,j}(x) \right]. \tag{2}$$

The two terms correct the activation and output direction, respectively. For a selected set $\mathcal{R}_E(x) \subset \{1, \ldots, d_{\text{ff}}\}$, we compute

$$\hat{y}_E(x) = y_S(x) + \Delta y_E(x), \tag{3}$$

$$\Delta y_E(x) = W_S^2[:,\mathcal{R}] \left(a_E^{\mathcal{R}}(x) - a_S^{\mathcal{R}}(x) \right) + \Delta W_E^2[:,\mathcal{R}] a_E^{\mathcal{R}}(x), \tag{4}$$

where

$$a_E^{\mathcal{R}}(x) = \text{SiLU}\left((W_S^1[\mathcal{R},:] + \Delta W_E^1[\mathcal{R},:])x \right) \odot \left((W_S^3[\mathcal{R},:] + \Delta W_E^3[\mathcal{R},:])x \right). \tag{5}$$

Only channels in $\mathcal{R}$ are recomputed; the others retain their resident contributions. For a quantized base, residuals are measured from the dequantized resident weights to the original bf16 weights.

2.2 Activation-Guided Channel Selection

Selection uses resident activations and a precomputed residual norm, without reading the CPU bank:

$$r_{E,j} = \left\| \Delta W_E^1[j,:] \right\|_2^2 + \left\| \Delta W_E^3[j,:] \right\|_2^2 + \left\| \Delta W_E^2[:,j] \right\|_2^2. \tag{6}$$

The norms remain on the GPU. A *record* contains all tokens routed to an expert in one forward pass over a batch of sequences. Using activations from the resident forward pass, we score channel j for expert E by

$$s_{E,j}(x) = \mathbb{E}_{t \in \text{batch}} \left[|a_{S,j}(x_t)|^2 \right] r_{E,j}, \tag{7}$$

and select p channels, $\mathcal{R}_E(x) = \text{TopK}_{j=1,\ldots,d_{\text{ff}}}(s_{E,j}(x), p)$ over all candidate channels. The score combines input-dependent activation energy with residual magnitude. Using original-expert activations instead would require fetching the candidate rows before selection.

Layer-global allocation. We can distribute the same transfer budget across a layer instead of assigning every record a fixed width. The allocator ranks (record, channel) items by Equation 7 multiplied by the record's token count and selects the layer-global top-K, subject to a per-record floor and cap ($[32, 512]$ at p=128 on Qwen3). This allocates more channels to records with greater aggregate correction scores without increasing transferred bytes.

2.3 Analysis: Recovery Objective and Selection

Exact substitution property. Assume that E and S share channel coordinates and the residuals are stored exactly. Adding each triplet recovers the original gate, up, and down weights on the selected channels. Equation 4 then gives

$$\hat{y}_E(x) = \sum_{j \in \mathcal{R}} W_E^2[:,j]\, a_{E,j}(x) + \sum_{j \notin \mathcal{R}} W_S^2[:,j]\, a_{S,j}(x). \tag{8}$$

The remaining error consists of unselected channel contributions:

$$y_E(x) - \hat{y}_E(x) = \sum_{j \notin \mathcal{R}} \left(W_E^2[:,j] a_{E,j}(x) - W_S^2[:,j] a_{S,j}(x) \right). \tag{9}$$

This identity holds for any resident base with the same channel coordinates. With NF4 residuals, selected channels also incur bank quantization error. Cross-channel cancellation means that selecting more channels need not decrease the total error monotonically.

Recovery spectrum. Optimal selection is combinatorial. Fix a layer input batch x and its routing. Let $c_{rj}(x)$ be the correction in Equation 2 for record r and channel j, multiplied by each token's routing weight and embedded in the batch output vector, with zeros outside the record's tokens. For a budget of at most K selected (record, channel) items, define the idealized recovery spectrum:

$$\mathcal{R}_S(K) = \mathbb{E}_x\Big[\min_{|A|\le K} \big\| \textstyle\sum_{(r,j)\notin A} c_{rj}(x)\big\|_2^2\Big]. \tag{10}$$

$\mathcal{R}_S(0)$ measures local uncorrected reconstruction error; $\mathcal{R}_S(K)$ is the best local error attainable with at most K exact corrections. These objectives can rank resident bases differently. They do not directly predict downstream accuracy or account for hidden-state and routing changes caused by earlier compressed layers.

Independent energy ranking uses $v_i(x) = \|c_i(x)\|_2^2$ and drops off-diagonal terms of the correction Gram matrix $G_{ij}(x) = \langle c_i(x), c_j(x)\rangle$. Figure 3 plots pooled per-record tails under this diagonal approximation; its horizontal axis counts corrected channels per record, rather than the layer-global budget in Equation 10. Equation 7 is the deployed proxy for individual energies. The following standard bounded-coherence argument (Tropp, 2006) describes exact energy ranking.

Proposition 1 (Independent ranking under bounded coherence) *For a fixed input x, write $D_x(T) = \sum_{i\in T} v_i(x)$ for a tail T, and suppose the coherence bound*

$$\left|2\sum_{\substack{i<j\\ i,j\in T}} \langle c_i(x), c_j(x)\rangle\right| \le \gamma D_x(T), \qquad 0 \le \gamma < 1,$$

holds for the ranking-selected tail and for some error-minimizing tail. Selecting the layer-global K items with largest $v_i(x)$ yields residual error at most $(1+\gamma)/(1-\gamma)$ times the pointwise optimum inside Equation 10. If $\gamma = 0$ and items have equal transfer cost, layer-global top-K is exactly optimal.

This result concerns exact correction energies and an unconstrained cardinality budget. The deployed proxy, record floors/caps, and quantized residuals are evaluated empirically below.

Tail-error principle. Define the expected energy of channel j as $\varepsilon_j = \mathbb{E}_x[\|c_j(x)\|_2^2]$. Under the additive $\mathrm{diag}(G)$ approximation, Equation 9 gives, for a fixed selected set $\mathcal{R}$,

$$\mathcal{E}_{\mathrm{patched}}(\mathcal{R}) = \sum_{j\notin\mathcal{R}} \varepsilon_j. \tag{11}$$

For ideal top-p selection, this is the unselected energy tail. Improving only the selected channels has no effect on this additive objective after exact correction. A base with lower uncorrected error can therefore leave a larger tail after patching. We use this decomposition to interpret the observed ranking reversals.

Complete channel triplets give the substitution identity. Expanding a channel correction gives $c_j = (a_{E,j}-a_{S,j})W_S^2[:,j]+a_{S,j}\Delta W^2[:,j]+(a_{E,j}-a_{S,j})\Delta W^2[:,j]$. A down-only correction restores the middle term. Correcting a complete triplet also accounts for activation changes and their interaction with the output residual.

2.4 Deployment and Streaming Cost

The compressed experts, router, attention, and selection metadata remain in HBM. The Qwen3 residual bank occupies 15.4 GB in host memory: 14.5 GB of NF4 values plus scales. Each selected channel fetches three length-d vectors. The values-only transfer per record is therefore $p/(4d_{\mathrm{ff}})$ of a bf16 expert read: 4.2% at p=128 and 8.3% at p=256 (0.39 MB per record at p=128). Scales add 6.25% to the transferred values. Selection metadata occupies approximately 10 MB on the

GPU. Resident expert bytes exclude non-expert weights, KV cache, and temporary workspaces; the reported compression ratio is therefore not a ratio of total peak device memory.

The fixed-width budget is per record; global allocation preserves the total per-layer budget. Batching amortizes the transfer across tokens. Our measured traffic savings over expert caching occur in prefill; whole-expert caching transfers fewer bytes at the headline budget for the decode batches tested. Patches can also be applied to cached compressed experts. The latency measurements in Appendix F characterize a research executor; they do not establish a request-level serving-speed advantage.

3 Experimental Setup

We evaluate Qwen3-30B-A3B and Mixtral-8x7B. Qwen3 has 48 MoE layers, 128 experts per layer, top-k=8, d=2048, and d_{ff}=768. Merging uses a min-energy grouping objective and usage-weighted means. Groupings at ratios 2:1, 4:1, and 8:1 are solved independently. Unless stated otherwise, patches select p=128 of 768 channels, and evaluations use batch size 16. The ablation reference is a 2:1 bf16 merged base with NF4 residuals and p=128.

We report the unweighted mean over eight zero-shot tasks: ARC-c/e (Clark et al., 2018), BoolQ (Clark et al., 2019), HellaSwag (Zellers et al., 2019), MMLU (Hendrycks et al., 2021), OBQA (Mihaylov et al., 2018), RTE (Dagan et al., 2006), and WinoGrande (Sakaguchi et al., 2021), using lm-evaluation-harness (Gao et al., 2021) and `acc_norm` where standard. For Qwen3, we measure WikiText-2 word-perplexity (Merity et al., 2017) on a fixed quarter of the evaluation windows; Mixtral results use token-perplexity. Generation is evaluated on GSM8K (Cobbe et al., 2021) (reported 8-shot configuration, flexible extraction, n=500). Qwen3 generation uses a chat template with `/no_think`, greedy decoding, and up to 512 new tokens. MATH-500 uses a 4-shot Minerva prompt and symbolic-equivalence scoring; IFEval uses 541 prompts. Mixtral uses a base-model prompt without a chat template.

Evaluation scope. The selector pools all tokens supplied to an expert in a forward pass. During generation these are observed prompt or generated prefix tokens. During parallel teacher-forced likelihood scoring, later tokens in the supplied continuation or window can instead affect the correction set used at earlier positions. Causal attention does not remove this path through the shared selection set. We therefore report Avg8 and perplexity for patched models as *batch-conditioned diagnostics*, not as validated prefix-causal next-token likelihoods. The same qualification applies to likelihood-scored selection and allocation ablations. The unpatched baselines do not acquire this selection dependence; their comparison with patched diagnostics does not establish causal likelihood improvement. No prefix-causal rerun is reported. Our generation claims rely on generated-answer benchmarks, which do not provide the selector with future reference answers.

Scalar baselines include RTN, an AWQ-style activation-aware variant (Lin et al., 2025), and GPTQ (Frantar et al., 2023), evaluated with group-128 fake quantization. QTIP provides the codebook base. Compression ratios count resident expert-weight bytes, including quantizer scales. Table 1 retains the reported reference-path results; the fused executor implements allocation and host-bank streaming. On the 16× configuration, its GSM8K score is 0.810 versus 0.832 on the reference path; its batch-conditioned Avg8 is 0.6808 versus 0.6804. These paths have measurable run-to-run variation.

4 Results

4.1 Generation Under Deep Expert Compression

Table 1 compares generated-answer accuracy under deep resident expert compression, alongside batch-conditioned diagnostics. At 16×, the stacked model's GSM8K score is 0.832, compared with 0.868 uncompressed and 0.292 for fixed-width patched pruning at matched resident expert bytes and transfer budget. Without global allocation, the same merged-codebook base already reaches 0.792. This generation comparison is the main evidence that retaining correctable expert-specific channels can be preferable to deleting experts at this budget.

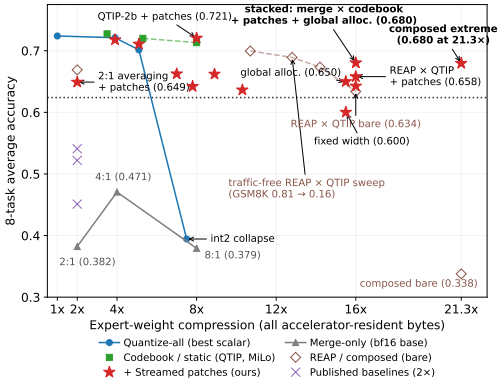

Figure 2: Resident expert compression versus batch-conditioned eight-task scores on Qwen3. The dotted line is a descriptive reference (within 10 points of unmerged), not a causal-accuracy or deployment criterion. Hollow diamonds show traffic-free REAP×QTIP. The 21.3× point is an exploratory endpoint selected using GSM8K. See §3 for the scope of the likelihood-scored comparisons.

The resident model alone need not be accurate. At 2×, 2:1 averaging has a batch-conditioned suite score of 0.3822; correction raises it to 0.6491. At 16×, the original expert is represented by a residual around a merged QTIP base. These diagnostics help describe recovery but do not replace the generated-answer comparison. The complete measured operating points, including traffic-free alternatives, appear in Appendix B.

Table 1: Qwen3 at deep compression. GSM8K measures generated answers; Avg8 and patched word-PPL are batch-conditioned diagnostics (§3). HBM counts resident expert weights. PCIe counts fetched values relative to a bf16 expert read; scales add 6.25%. Rows 8–11, including 9a, have identical 3.63 GB residents; rows 9–11 also match the transfer budget. Row 9a is the strongest reported pruning suite result; dashes denote unreported metrics. †: reference allocation path. ∗: exploratory endpoint selected using GSM8K.

#	Compr.	HBM (GB)	Method	PCIe	Avg8	GSM8K	Wiki word-PPL
1	1×	58.1	unmerged bf16	0	0.7240	0.868	11.09
2	8.0×	7.26	QTIP-2b (bare)	0	0.7132	0.862	12.31
3	8.0×	7.26	QTIP-2b + patches (ours)	4.2%	0.7205	0.878	11.46
4	10.7×	5.44	REAP×QTIP-2b (25%)	0	0.6996	0.814	12.70
5	12.8×	4.53	REAP×QTIP-2b (37.5%)	0	0.6892	0.348	13.38
6	14.2×	4.08	REAP×QTIP-2b (43.75%)	0	0.6734	0.264	13.94
7	15.5×	3.75	4:1 int4 + global alloc. (ours)†	8.3%	0.6499	0.750	14.83
8	16.0×	3.63	REAP×QTIP-2b (bare, 50%)	0	0.6339	0.156	14.79
9	16.0×	3.63	REAP×QTIP-2b + patches (ours)	8.3%	0.6580	0.292	12.66
9a	16.0×	3.63	+ global alloc. (ours)	8.3%	0.6646	—	—
10	16.0×	3.63	2:1 over QTIP-2b + patches (ours)	8.3%	0.6427	0.792	20.09
11	16.0×	3.63	+ global alloc. (stacked, ours)†	8.3%	0.6804	0.832	13.09
12*	21.3×	2.73	REAP-25%×QTIP×2:1 + patches (ours)†	8.3%	0.6796	0.800	13.98

Matched comparisons and their scope. Rows 9–11 separate the base and allocation choices at the same resident expert bytes and transfer budget. Patching pruning repairs surviving weights but cannot restore deleted expert computations. Its fixed-width GSM8K score is 0.292; GSM8K for pruning with allocation has not been reported. The strongest pruned suite diagnostic is 0.6646 (row 9a), so the stacked row's margin is 1.58 points, rather than 2.24 against fixed-width pruning. A direct paired interval for the best-arm comparison is not available. In the task-level stacked replicate, MMLU accounts for 2.075 of the 2.10 average-point difference from fixed-width patched pruning. This is not a uniform gain across tasks.

Generation beyond GSM8K. The fused stacked model reaches 0.484 on MATH-500 and 0.599 on IFEval prompt-strict, versus 0.486 and 0.599 for bare 8× QTIP-2b at twice its resident expert memory (Table 2). These similar point estimates support generation preservation; without paired intervals they are not an equivalence test. Bare pruning at the same 3.63 GB scores 0.024 and 0.224. Patched pruning and bf16 were not evaluated on these two tasks, so this extension does not establish superiority over the strongest patched baseline or parity with bf16.

Resident memory and transfer cost. The generation results trade expert residency for host traffic. At 3.63 GB of resident expert weights, a 50% QTIP-2b expert cache transfers fewer bytes than headline-width patches at all tested decode batches. For 2k/8k-token prefill, patch payloads are 2.05/0.54 MB per token versus 3.07/0.81 MB for the cache. Residual scales add 6.25% to patch payloads. The cache serves the more accurate bare QTIP model (GSM8K 0.862 versus 0.832 for the

Table 2: Generated-answer results beyond GSM8K (fused path). IFEval is prompt-level strict accuracy.

Method	Expert HBM (GB)	MATH-500	IFEval
QTIP-2b, 8×	7.26	0.486	0.599
QTIP-2b + patches, 8×	7.26	0.516	0.595
Bare REAP×QTIP-2b, 16×	3.63	0.024	0.224
Stacked Delta-MoE, 16×	3.63	0.484	0.599

stacked reference path). These are routing-trace replay results, with width-256 patch bytes derived from width-128 measurements. These traces and uncached forward timings do not establish request-level serving speed.

Exploratory endpoint. Combining 25% pruning with merging and quantization gives a 21.3× configuration with GSM8K 0.800. GSM8K was used to select this endpoint, so its score is a selected result rather than an independent confirmation. We retain it as an exploratory operating point (Appendix B).

4.2 Cross-Model Validation on Mixtral-8x7B

Table 3 reports Mixtral results across compression families. Patches improve GSM8K from 0.222 to 0.454 for Sub-MoE and from 0.472 to 0.534 for GPTQ-int3.

Table 3: Mixtral-8x7B across compression families. Ratios count resident expert bytes. Patches use FP8 residuals at mean K=2048 of 14,336 channels (7.1% of a bf16 expert read). GSM8K measures generated answers (n=500). Avg8 and patched token-PPL are batch-conditioned diagnostics (§3). Rows 11–15 match residency and patched transfer budgets. ‡: legacy PPL protocol, so cross-block PPL is not directly comparable.

#	Compr.	Method	PCIe	Avg8	GSM8K	Wiki token-PPL
1	1×	unmerged bf16	0	0.7230	0.606	3.34
2	2.0×	Sub-MoE 2:1 merge	0	0.6117	0.222	7.35‡
3	2.0×	+ patches (ours)	7.1%	0.6808	0.454	5.13‡
4	2.0×	REAP-50% pruning	0	0.6360	0.156	5.70
5	3.8×	GPTQ-int4 experts	0	0.7195	0.570	3.50
6	3.8×	+ patches (ours)	7.1%	0.7202	0.570	3.47
7	5.1×	GPTQ-int3 experts	0	0.7001	0.472	4.01
8	5.1×	+ patches (ours)	7.1%	0.7052	0.534	3.91
9	8.0×	QTIP-2b (bare)	0	0.7026	0.502	4.05
10	8.0×	+ patches (ours)	7.1%	0.7029	0.500	3.95
11	16.0×	REAP-50%×QTIP-2b (bare)	0	0.5975	0.118	6.39
12	16.0×	+ patches (ours)	7.1%	0.6035	0.120	6.28
13	16.0×	2:1 usage-wt. over QTIP-2b (bare)	0	0.3533	0.000	9521
14	16.0×	+ patches (ours)	7.1%	0.6031	0.192	6.65
15	16.0×	+ global alloc. (ours)	7.1%	0.6040	0.222	6.62
16	21.3×	REAP-25%×2:1×QTIP-2b (bare)	0	0.3755	0.016	74.6
17	21.3×	+ patches + alloc. (ours)	7.1%	0.5217	0.100	7.12

At 16×, bare merging performs much worse than pruning, yet patches and allocation yield the highest reported Avg8 and GSM8K point estimates (row 15). At matched resident bytes and transfer budget, Avg8 is slightly higher than patched pruning (0.6040 versus 0.6035), while GSM8K reaches 0.222 versus 0.120. Patched pruning retains lower perplexity (6.28 versus 6.62).

At the extreme 21.3× point, correction and allocation raise Avg8 from 0.3755 to 0.5217 and GSM8K from 0.016 to 0.100 (rows 16–17). Only the bare and corrected compositions are tested at this ratio. This demonstrates recovery at the deepest compression, but generation remains well below uncompressed Mixtral (0.606) and bare 8× QTIP (0.502).

4.3 Correction Granularity and Budget Allocation

The correction budget has two choices: which computations to restore, and how to divide the budget across expert records. Table 4 separates these choices using existing controls. All scores in this table are likelihood-scored suite diagnostics subject to §3; they test behavior under that protocol rather than establish causal next-token accuracy.

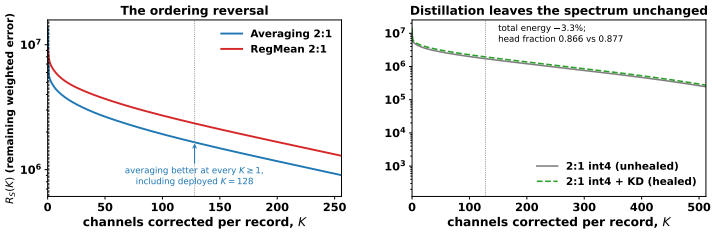

Figure 3: Pooled per-record diagonal energy tails on original-model hidden states, with squared routing weights. Left: averaging and RegMean 2:1 bases. Correcting one highest-energy channel per record removes 58.7% of averaging's pooled energy; RegMean has the smaller tail only at zero corrections. The horizontal axis is a per-record width, not the layer-global budget of Equation 10. Right: distillation changes total diagonal energy by only -3.3%.

Coupled channels versus projection-wise correction. The DecDEC-style control selects gate/up input coordinates and down-projection channels independently, correcting the first two projections before the SwiGLU nonlinearity. Its widths $(340, 340, 129)$ fetch the same 786,432 residual values per record as 128 coupled triplets, with matched nominal NF4 values and scales. Coupled correction improves the suite diagnostic by 4.12–5.95 points on merged or composed bases. In contrast, the projection-wise control differs by only $+0.25$ and -0.33 points on int4 and int3 quantization alone, with reported paired intervals including zero. This contrast supports complete channel recomputation for merge-scale residuals. The control is an accuracy emulation, not a comparison with DecDEC's optimized transport or kernels.

Table 4: Qwen3 correction and allocation controls at equal nominal residual budgets within each row. Values are batch-conditioned Avg8 diagnostics. Upper block: independent projection corrections versus coupled triplets. Lower block: calibration-profiled static allocation, fixed widths, and dynamic layer-global allocation. Ratios include resident quantizer scales. Full controls and reported intervals are in Appendix C.

Resident base	Projection-wise	Coupled	Gain (pp)
2:1 bf16 merge (2.0×)	0.5966	0.6491	5.25
2:1 int4 merge (7.8×)	0.6009	0.6421	4.12
2:1 int3 merge (10.3×)	0.5769	0.6364	5.95
Resident base	**Static alloc.**	**Fixed width**	**Dynamic alloc.**
4:1 int4 merge (15.5×)	0.5709	0.6002	0.6499
2:1 QTIP-2b merge (16.0×)	0.6133	0.6427	0.6804

Allocation should see the current input. At 16×, dynamic allocation improves the suite diagnostic over a calibration-profiled schedule by 6.71 points (reported 95% paired interval $[5.73, 7.71]$), and over fixed widths by 3.77 points from the printed scores. The corresponding GSM8K change is 0.792 to 0.832 (Table 1). The gain is not universal: allocation is a null at the 2:1 bf16 ablation anchor. NF4 storage provides a separate benefit by permitting wider selection at fixed transfer bytes.

The controls also identify limits of the selector. Removing the residual-norm factor changes the suite diagnostic by only $+0.02$ points ($[-0.72, +0.77]$), so that factor is not needed for the observed benefit. Increasing the headline evaluation batch from 16 to 64 and 128 lowers its diagnostic from 0.6804 to 0.6713 and 0.6666. Larger records improve transfer amortization but share one correction set over more tokens. The proxy captures approximately 70% of the per-record energy oracle's mass at 2:1 merging on original-model hidden states; this does not guarantee deployed accuracy. Additional controls examine resident low-rank substitutes and LoRA (Hu et al., 2022) recovery.

4.4 Better Bare Merges Need Not Improve Recovery

RegMean (Jin et al., 2023) raises the bare suite diagnostic from 0.3822 for averaging to 0.4774, a 9.52-point gain, yet their patched scores are 0.6491 and 0.6440, respectively. These are similar point estimates, not an equivalence test. RegMean's reconstruction gain concentrates in the high-energy channels that patching replaces. Although it starts with less total error, averaging has the

smaller remaining tail after correcting the highest-energy channel in each record under this surrogate (Figure 3). Better reconstruction of a fixed merge group therefore need not yield a better recoverable base.

Distillation illustrates a complementary effect. For the 2:1 int4 base, it raises the patched suite diagnostic from 0.6421 to 0.6625, while total diagonal residual energy changes by only -3.3% (Figure 3, right). The improvement therefore need not track a large reduction in this local energy measure. It has an additional cost: resident rank-16 bf16 LoRA factors add approximately 0.8 GB, reducing expert compression from $7.8\times$ to $7.0\times$ at the same correction width. This comparison shows that training-based recovery can complement channel correction, but does not establish a gain at identical resident memory. Both comparisons use the batch-conditioned protocol of §3.

Re-encoding merged weights also outperforms keeping one member's codes. These results motivate evaluating the base and correction together, without treating a single spectrum statistic as a universal selection rule. Across nine bases, bare accuracy ranks patched suite scores more strongly than the proxy tail (ρ=0.81 versus 0.63; Appendix D). The spectrum describes channel-recoverable error, but it does not account for every functional change to the base or replace downstream evaluation.

5 Related Work

MoE compression. Expert merging and clustering reduce the expert count (Li et al., 2026; Chen et al., 2025; Li et al., 2023), while pruning removes experts (Lu et al., 2024; Muzio et al., 2024; Xie et al., 2024; Lasby et al., 2026). Quantization and decomposition retain experts at lower precision or rank (Frantar & Alistarh, 2023; Huang et al., 2025b; Li et al., 2025; Chen et al., 2026; He et al., 2025). We use scalar (Frantar et al., 2023; Lin et al., 2025) and codebook (Chee et al., 2023; Tseng et al., 2024a; Egiazarian et al., 2024; Tseng et al., 2024b) quantization as resident bases. Our results show that choosing a base for streamed correction requires considering the error that remains after correction, alongside its bare accuracy.

Residuals and compensators. BitDelta and DeltaZip compress fine-tuning deltas (Liu et al., 2024; Yao et al., 2025); D2-MoE compresses expert-specific deltas over a shared base (Gu et al., 2025); and MiLo adds low-rank compensators (Huang et al., 2025a). ResMoE extracts a barycenter expert and compresses its residuals (Ai et al., 2025). Its channelwise sub-MLP representation already couples gate/up rows with down columns for SwiGLU experts. Our contribution uses complete channel residuals as the units of input-dependent retrieval from host memory, with NF4 storage on Qwen3 (Dettmers et al., 2023).

Dynamic correction and offloading. DecDEC streams activation-selected quantization residuals from CPU memory and corrects individual linear projections (Park et al., 2025). Delta-MoE instead selects and recomputes complete SwiGLU channels, restoring their nonlinear contributions up to residual quantization. Our nominal-byte-matched accuracy emulation compares these correction granularities on merged and quantized bases; it does not reproduce DecDEC's optimized executor. Whole-expert offloading and caching (Xue et al., 2025; Kamahori et al., 2025) provide a complementary way to reduce residency; cached compressed experts can also use channel corrections.

6 Conclusion and Limitations

Delta-MoE uses host-resident residuals to recover selected SwiGLU channels of a compressed MoE. Its strongest evidence is generation: the Qwen3 $16\times$ configuration retains substantial GSM8K accuracy and has similar MATH-500/IFEval point estimates to an $8\times$ QTIP base. The exact substitution identity explains the correction unit, while recovery spectra help interpret its limits. The resulting tradeoff is workload- and model-dependent: Mixtral's deep-compression generation loss is larger, and a QTIP expert cache has lower traffic for the tested decode workloads.

Prefix-causal validation of the batch-conditioned suite and perplexity diagnostics remains open. The $21.3\times$ endpoint is exploratory, and several comparisons lack direct paired uncertainty. Compression ratios count expert weights rather than peak total HBM. The fused executor runs end to end, but uncached forward timings and routing-trace simulations do not establish request-level serving speed.

Reproducibility Statement

We will release the bank builder, fused Triton kernels, evaluation-harness patches, grouping JSONs, quantization drivers, and routing traces. Calibration data, comparator details, and implementation settings are documented in the appendix. The evaluation scope in §3 is part of the reported protocol. Evaluations were conducted on A100 and H200 nodes.

AI Disclosure

In this work, we used generative AI tools to help develop theoretical models and conceptual frameworks, to formulate mathematical claims, to assist in the writing of proofs, to assist in implementing methods and evaluation infrastructure, and to design and provide feedback on research methodology and experiments. We have not used generative AI tools to generate synthetic data sets, to provide critical ingredients for proving mathematical claims, to propose or refine hypotheses, or to clean and reformat data sets; translation and qualitative or thematic data analysis are not applicable to this work. Additionally, generative AI tools helped to develop and create software code, to create artifacts, to draft parts of this paper and edit it for readability, to identify relevant literature, to format references, and to propose the title and keywords. We have reviewed all AI-assisted work: mathematical claims and proofs were verified manually by the authors; AI-assisted code was validated against deterministic reference implementations under pre-registered accuracy gates (interval-based equivalence and bit-exactness tests, Appendices F and G); and all AI-drafted text was reviewed and edited by the authors. We take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

References

Mengting Ai, Tianxin Wei, Yifan Chen, Zhichen Zeng, Ritchie Zhao, Girish Varatkar, Bita Darvish Rouhani, Xianfeng Tang, Hanghang Tong, and Jingrui He. ResMoE: Space-efficient compression of mixture of experts LLMs via residual restoration. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.1*, pp. 1–12. Association for Computing Machinery, 2025. doi: 10.1145/3690624.3709196. URL https://doi.org/10.1145/3690624.3709196.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. QuIP: 2-bit quantization of large language models with guarantees. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

I-Chun Chen, Hsu-Shen Liu, Wei-Fang Sun, Chen-Hao Chao, Yen-Chang Hsu, and Chun-Yi Lee. Retraining-free merging of sparse MoE via hierarchical clustering. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=hslOzRxzXL.

Xiaodong Chen, Mingming Ha, Zhenzhong Lan, Jing Zhang, and Jianguo Li. MoBE: Mixture-of-basis-experts for compressing MoE-based LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=8RV6H50OSf.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions, 2019. URL https://arxiv.org/abs/1905.10044.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge, 2018. URL https://arxiv.org/abs/1803.05457.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

Ido Dagan, Oren Glickman, and Bernardo Magnini. The PASCAL recognising textual entailment challenge. In Joaquin Quiñonero-Candela, Ido Dagan, Bernardo Magnini, and Florence d'Alché Buc (eds.), *Machine Learning Challenges. Evaluating Predictive Uncertainty, Visual Object Classification, and Recognising Tectual Entailment*, pp. 177–190, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-33428-6.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. DeepSeekMoE: Towards ultimate expert specialization in mixture-of-experts language models, 2024. URL https://arxiv.org/abs/2401.06066.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: efficient finetuning of quantized LLMs. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA, 2023. Curran Associates Inc.

Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. Extreme compression of large language models via additive quantization. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: scaling to trillion parameter models with simple and efficient sparsity. *J. Mach. Learn. Res.*, 23(1), January 2022. ISSN 1532-4435.

Elias Frantar and Dan Alistarh. QMoE: Practical sub-1-bit compression of trillion-parameter models, 2023. URL https://arxiv.org/abs/2310.16795.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. OPTQ: Accurate quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=tcbBPnfwxS.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, Jason Phang, Laria Reynolds, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. A framework for few-shot language model evaluation, September 2021. URL https://doi.org/10.5281/zenodo.5371629.

Hao Gu, Wei Li, Lujun Li, Zhu Qiyuan, Mark G. Lee, Shengjie Sun, Wei Xue, and Yike Guo. Delta decompression for MoE-based LLMs compression. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=ziezViPoN1.

Shwai He, Daize Dong, Liang Ding, and Ang Li. Towards efficient mixture of experts: A holistic study of compression techniques. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=HTpMOl6xSI.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Beichen Huang, Yueming Yuan, ZELEI SHAO, and Minjia Zhang. MiLo: Efficient quantized MoE inference with mixture of low-rank compensators. In *Eighth Conference on Machine Learning and Systems*, 2025a. URL https://openreview.net/forum?id=NXVXiJmhe1.

Wei Huang, Yue Liao, Jianhui Liu, Ruifei He, Haoru Tan, Shiming Zhang, Hongsheng Li, Si Liu, and XIAOJUAN QI. Mixture compressor for mixture-of-experts LLMs gains more. In *The Thirteenth International Conference on Learning Representations*, 2025b. URL https://openreview.net/forum?id=hheFYjOsWO.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024. URL https://arxiv.org/abs/2401.04088.

Xisen Jin, Xiang Ren, Daniel Preotiuc-Pietro, and Pengxiang Cheng. Dataless knowledge fusion by merging weights of language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=FCnohuR6AnM.

Keisuke Kamahori, Tian Tang, Yile Gu, Kan Zhu, and Baris Kasikci. Fiddler: CPU-GPU orchestration for fast inference of mixture-of-experts models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=N5fVv6PZGz.

Mike Lasby, Ivan Lazarevich, Nish Sinnadurai, Sean Lie, Yani Ioannou, and Vithursan Thangarasa. REAP the experts: Why pruning prevails for one-shot MoE compression. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ukGxWd2aDG.

Lujun Li, Zhu Qiyuan, Jiacheng Wang, Wei Li, Hao Gu, Sirui Han, and Yike Guo. Sub-MoE: Efficient mixture-of-expert LLMs compression via subspace expert merging. In *AAAI 2026*, 2026.

Pingzhi Li, Zhenyu Zhang, Prateek Yadav, Yi-Lin Sung, Yu Cheng, Mohit Bansal, and Tianlong Chen. Merge, then compress: Demystify efficient SMoE with hints from its routing policy, 2023.

Wei Li, Lujun Li, Hao Gu, You-Liang Huang, Mark G. Lee, Shengjie Sun, Wei Xue, and Yike Guo. MoE-SVD: Structured mixture-of-experts LLMs compression via singular value decomposition. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=acJ3vdFljk.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Guangxuan Xiao, and Song Han. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. *GetMobile: Mobile Comp. and Comm.*, 28(4):12–17, January 2025. ISSN 2375-0529. doi: 10.1145/3714983.3714987. URL https://doi.org/10.1145/3714983.3714987.

James Liu, Guangxuan Xiao, Kai Li, Jason D. Lee, Song Han, Tri Dao, and Tianle Cai. BitDelta: Your fine-tune may only be worth one bit. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=XuWWq3gy7W.

Xudong Lu, Qi Liu, Yuhui Xu, Aojun Zhou, Siyuan Huang, Bo Zhang, Junchi Yan, and Hongsheng Li. Not all experts are equal: Efficient expert pruning and skipping for mixture-of-experts large language models, 2024. URL https://arxiv.org/abs/2402.14800.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Byj72udxe.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2381–2391, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1260. URL https://aclanthology.org/D18-1260/.

Alexandre Muzio, Alex Sun, and Churan He. SEER-MoE: Sparse expert efficiency through regularization for mixture-of-experts, 2024. URL https://arxiv.org/abs/2404.05089.

Yeonhong Park, Jake Hyun, Hojoon Kim, and Jae W. Lee. DecDEC: a systems approach to advancing low-bit LLM quantization. In *Proceedings of the 19th USENIX Conference on Operating Systems Design and Implementation*, OSDI '25, USA, 2025. USENIX Association. ISBN 978-1-939133-47-2.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. WinoGrande: an adversarial Winograd schema challenge at scale. *Commun. ACM*, 64(9):99–106, August 2021. ISSN 0001-0782. doi: 10.1145/3474381. URL https://doi.org/10.1145/3474381.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=B1ckMDqlg.

J. A. Tropp. Greed is good: algorithmic results for sparse approximation. *IEEE Trans. Inf. Theor.*, 50(10):2231–2242, September 2006. ISSN 0018-9448. doi: 10.1109/TIT.2004.834793. URL https://doi.org/10.1109/TIT.2004.834793.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP#: even better LLM quantization with Hadamard incoherence and lattice codebooks. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024a.

Albert Tseng, Qingyao Sun, David Hou, and Christopher De Sa. QTIP: Quantization with trellises and incoherence processing. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024b. URL https://openreview.net/forum?id=7sdkLVuYCU.

Yanyue Xie, Zhi Zhang, Ding Zhou, Cong Xie, Ziang Song, Xin Liu, Yanzhi Wang, Xue Lin, and An Xu. MoE-Pruner: Pruning mixture-of-experts large language model using the hints from its router, 2024. URL https://openreview.net/forum?id=hB6jYbvypa.

Leyang Xue, Yao Fu, Zhan Lu, Luo Mai, and Mahesh Marina. MoE-Infinity: Efficient MoE inference on personal machines with sparsity-aware expert cache, 2025. URL https://arxiv.org/abs/2401.14361.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Xiaozhe Yao, Qinghao Hu, and Ana Klimovic. DeltaZip: Efficient serving of multiple full-model-tuned LLMs. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, pp. 110–127, New York, NY, USA, 2025. Association for Computing Machinery. ISBN 9798400711961. doi: 10.1145/3689031.3717468. URL https://doi.org/10.1145/3689031.3717468.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In Anna Korhonen, David Traum, and Lluís Màrquez (eds.), *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Appendix

Contents

A Proofs 14
B The Full Compression Frontier 14
C Selection Ablations and Controls 15
D Recovery-Spectrum Diagnostics 17
E Mixtral Cross-Model Results 18
F Systems: Kernels, Latency, and Serving 19
G Reproduction Settings 22

A Proofs

Proposition 1. Fix x. For an uncorrected tail T, let $D_x(T) = \sum_{i\in T} v_i(x)$ and $E_x(T) = \|\sum_{i\in T} c_i(x)\|_2^2$. Expanding the square and applying the coherence assumption to a tail that satisfies it gives

$$(1-\gamma)D_x(T) \le E_x(T) \le (1+\gamma)D_x(T).$$

Selecting the K largest individual energies minimizes D_x over all feasible tails. If T_{rank} is its tail and $T^\star$ minimizes E_x, then

$$E_x(T_{\text{rank}}) \le (1+\gamma)D_x(T_{\text{rank}}) \le (1+\gamma)D_x(T^\star) \le \frac{1+\gamma}{1-\gamma}E_x(T^\star).$$

The chain uses the upper bound only on T_{rank} and the lower bound only on $T^\star$, which is why the proposition assumes coherence only for these two tails. The argument is pointwise in x. If a common $\gamma < 1$ satisfies the two-tail assumption almost surely over inputs, taking expectations preserves the same factor. When $\gamma = 0$, the bound shows that ranked selection attains the optimum. Equal per-item transfer costs make this a cardinality-constrained selection problem; record floors and caps impose additional constraints. □

Coherence diagnostics and proxy fidelity. We measure local coherence on sampled expert–token correction vectors, using original-model hidden states. On the deployed selector's uncorrected tail, the realized ratio $|2\sum_{i<j\in T}\langle c_i, c_j\rangle|/D_x(T)$ has mean 0.056–0.063 across the reported regimes (p95 $\le$0.20); 99.8% of sampled instances have a ratio below one. Boundary layers have larger mean ratios (0.35–0.44). For exact-energy ranked tails, the regime means are 0.0457 / 0.0432 / 0.0413, with p95 0.110 / 0.103 / 0.098 for merge / quantized / composed residuals. These measurements neither check optimal-tail coherence nor certify the full layer-global guarantee. Table 5 separately compares ranked selection with restricted exhaustive residual-error optima.

The deployed proxy has no informative pointwise score-fidelity bound in these measurements. We instead report its captured diagonal energy relative to a per-record exact-energy selector: the median across layers is 70% at 2:1 merging and 54% at 4:1, with the worst-layer values reported in Table 16. Global allocation increases median capture by 5–7 points. These diagnostics use the routing-weighted energy convention and original-model states; they are not measurements of hidden-state drift in the deployed model.

B The Full Compression Frontier

Additional frontier details. At fixed int3 precision, the recovery lift is largest on RTN, smaller on GPTQ, and negligible on MiLo. Bare merge quality is not monotone: the 2:1 and 4:1 bases score 0.3822 and 0.4706. Their partitions are optimized independently, so this comparison combines grouping and depth. The 2:1 base is selected for its 3.4× smaller deployed-K tail. MiLo-int3 and AWQ-int4 have patch lifts of $+0.09$ and -0.15 points (patched averages 0.7166 and 0.7197); their residuals leave little recoverable mass at the deployed channel budget. From int3 downward, the

Table 5: Restricted exhaustive check on sampled (layer, expert, token) instances. We enumerate all $\binom{20}{K}$ corrections drawn from the 20 channels with largest $v_i(x)$; channels outside this candidate set remain in every residual. Each cell contains 768 instances (fixed seed). The ratio is ranked-selection error divided by the minimum error over these restricted correction sets. The last column gives illustrative values of $(1+\gamma)/(1-\gamma)$ using the regime's mean / p95 ranked-tail coherence. These are not error bounds: optimal-tail coherence is unmeasured, and aggregate coherence statistics do not certify the pointwise premise.

Regime	K	median	p95	max	illustrative factor
merge (2:1 bf16)	4	1.007	1.045	1.25	1.10 / 1.25
	8	1.016	1.060	1.64	
quantized (2:1 int4)	4	1.007	1.043	1.26	1.09 / 1.23
	8	1.017	1.051	1.63	
composed (4:1 int4)	4	1.006	1.046	2.75	1.09 / 1.22
	8	1.017	1.063	1.85	

patched rows have suite diagnostic scores close to the best base evaluated here; this is not a causal-likelihood equivalence claim. Allocation on the pruned base is also statistically unresolved: 0.6646 vs. 0.6580, paired-replicate interval $[-0.17, +0.95]$ points.

Memory and transport conventions. Table 1 uses fetched-scales transport: bank values and scales stream from the host, and persistent GPU memory contains the residents. The bf16 expert bank is 58.1 GB; HBM entries divide this by the ledger-exact compression ratio. Patched merged and quantized bases add a 15.4 GB host bank. For pruned bases this scales by the surviving-expert fraction: approximately 7.7 GB at 50% pruning and 11.6 GB at 25%. Per-token traffic is reported in Table 13; evaluation follows §3.

Selection of the $21.3\times$ endpoint. The composed configuration was estimator-guided. The deployed tail estimator assigned both candidate bases equal scores; GSM8K then selected the 25% pruning base. Thus GSM8K was used for endpoint selection as well as for the reported generation comparison. This endpoint is exploratory; its reported GSM8K score is not an independent test of the selected configuration.

The preferred configuration therefore changes with depth and metric. Traffic-free pruning remains strong through $\sim 10.7\times$ (row 4). Beyond that point, its suite average declines gradually while generation drops sharply. The $12.8\times$ pruning row still has a higher suite average than the stacked model (0.6892 vs. 0.6804), but much lower GSM8K (0.348 vs. 0.832). Streaming recoverable residuals preserves stronger generation at greater compression.

Table 6 prints the complete frontier: every measured operating point, including rows dominated by traffic-free alternatives, with the published $2\times$ baseline block reproduced under our harness. Ratios are printed under the deployed fetched-scales convention (§2.4): a patched row's resident bytes equal its base's, so its ratio equals the base ratio; for merged bases without a printed bare row, the ratio is the row's byte-ledger total minus the script-measured scale stack. Earlier lanes and appendix tables refer to configurations by their pre-rewrite labels; the mapping is $7.1\to 8.0$, $12.5\to 15.5$, $12.8\to 16.0$, $14.2\to 16.0$.

C Selection Ablations and Controls

Bank precision. In the equal-byte comparison, NF4 storage doubles the selected width and improves recovery.

Activation and residual-norm factors. Removing $r_{E,j}$ from Equation 7 changes accuracy by only $+0.02$ points ($[-0.72, +0.77]$). The measured benefit of the selector comes mainly from its input-dependent activation factor; the residual norm supplies an error-based interpretation but is not required for the observed gain.

Additional grouping and bank-scale controls: uniform pairs beat balanced-kmeans++ grouping by 4.5 points at 4.2% transfer, and group-64 bank scaling cuts bank error 32% vs. outlier-sensitive per-row scaling. Allocation detail: the layer-global lift transfers to the merged-codebook base at

Table 6: Resident expert compression and batch-conditioned suite diagnostics on Qwen3 (§3). Compression = bf16 expert bytes over *resident expert bytes* (weights and group-128 fp16 quantizer scales); the host-side bank, whose values and scales both stream (§2.4), is charged to the traffic axis. Patched rows fetch 4.2% of the expert read in values (p=256: 8.3%; in-row scales add 6.25% of fetched bytes). The 2× block reproduces published merging and REAP pruning; KD rows deploy rank-16 bf16 LoRA factors (∼0.8 GB, resident). † = reference-path allocation rows (§3). Sweep-row ratios are ledger-computed (10.66/12.79/14.21×; printed values round). Same-platform REAP re-encodes are deterministic (exact reproduction at every measured cell); cross-platform encode spread grows with depth (−0.84 at 12.8×, −1.84 at 16×; A100 vs. H200).

Compr.	Method	PCIe	Avg
1×	unmerged bf16	0	0.7240
2.0×	D2-MoE (cr=0.5)	0	0.5414
2.0×	Sub-MoE (best)	0	0.5226
2.0×	HC-SMoE	0	0.4511
2.0×	RegMean 2:1 (bare)	0	0.4774
2.0×	RegMean 2:1 + patches (ours)	4.2%	0.6440
2.0×	2:1 averaging (bare)	0	0.3822
2.0×	**2:1 averaging + patches (ours)**	4.2%	**0.6491**
2.0×	REAP pruning	0	0.6689
3.6×	**MiLo-int4 + patches (ours)**	4.2%	**0.7246**
3.5×	MiLo-u16-int4 (uniform r16)	0	0.7275
3.6×	MiLo-int4	0	0.7198
3.9×	**int4-all + patches (ours)**	4.2%	**0.7178**
3.9×	int4-all (RTN)	0	0.7103
3.9×	int4-all (AWQ-style)	0	0.7212
3.9×	int4-all (AWQ-style) + patches	4.2%	0.7197
3.9×	int4-all (GPTQ)	0	0.7163
4.6×	MiLo-int3	0	0.7157
4.6×	MiLo-int3 + patches	4.2%	0.7166
5.1×	**int3-all + patches (ours)**	4.2%	**0.7109**
5.1×	int3-all (RTN)	0	0.6913
5.1×	int3-all (AWQ-style)	0	0.6895
5.1×	int3-all (GPTQ)	0	0.7014
5.3×	QTIP-3bit	0	0.7201

Compr.	Method	PCIe	Avg
7.0×	**2:1 int4 + KD + patches (ours)**	4.2%	**0.6625**
7.5×	**int2-all + patches (ours)**	8.3%	**0.6380**
7.5×	int2-all (best scalar: RTN)	0	0.3944
7.8×	**2:1 int4 + patches (ours)**	4.2%	**0.6421**
8.0×	QTIP-2bit	0	0.7132
8.0×	**QTIP-2bit + patches (ours)**	4.2%	**0.7205**
8.9×	**2:1 int3 + KD + patches (ours)**	4.2%	**0.6621**
10.3×	**2:1 int3 + patches (ours)**	4.2%	**0.6364**
10.7×	REAP × QTIP-2b (25% removal)	0	0.6996
12.8×	REAP × QTIP-2b (37.5% removal)	0	0.6892
14.2×	REAP × QTIP-2b (43.75% removal)	0	0.6734
15.5×	**4:1 int4 + patches (ours)**	8.3%	**0.6002**
15.5×	**4:1 int4 + global alloc. (ours)**†	8.3%	**0.6499**
16.0×	REAP × QTIP-2b (bare, 50%)	0	0.6339
16.0×	**REAP × QTIP-2b + patches (ours)**	8.3%	**0.6580**
16.0×	2:1 over QTIP-2b (bare)	0	0.3375
16.0×	**2:1 over QTIP-2b + patches (ours)**	8.3%	**0.6427**
16.0×	REAP×QTIP + patches + alloc. (ours)	8.3%	0.6646
16.0×	**+ global alloc. (stacked, ours)**†	8.3%	**0.6804**
21.3×	REAP-25% × QTIP-2b × 2:1 (bare)	0	0.3378
21.3×	**+ patches + tiered global alloc. (ours)**†	8.3%	**0.6796**

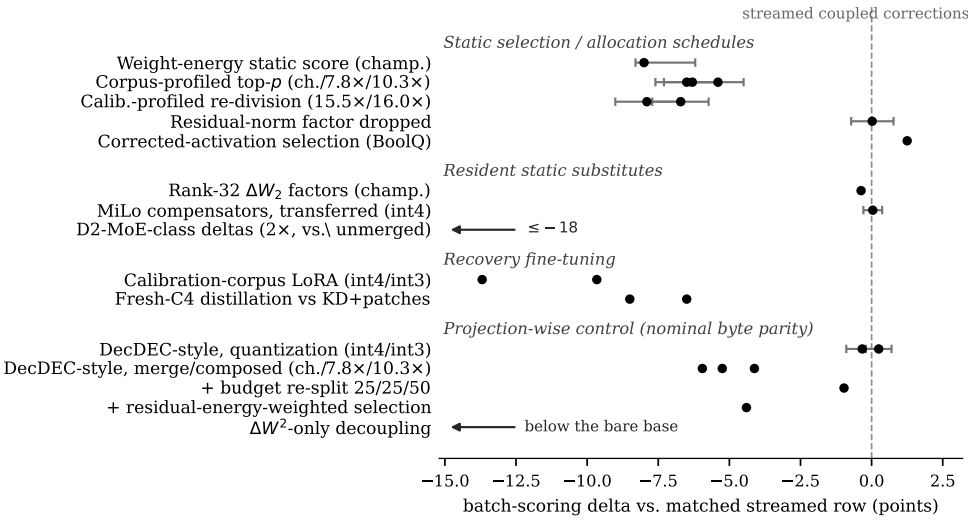

Figure 4: Batch-conditioned suite diagnostics for substitutes and controls at matched nominal residual budgets. Each dot is a printed-row difference against its matched streamed comparand (the ablation anchor 0.6491, the 7.8×/10.3×/15.5×/16.0× rows, or the KD+patches rows, as labeled); whiskers are the 95% paired-bootstrap CIs where measured; arrows mark off-scale rows. Only the two nulls and the byte-parity quantization ties reach the zero line. Absolute values in Table 7 below.

three-quarters strength (+3.78 [+2.85, +4.72]) and is null at the ablation anchor. These are batch-conditioned suite diagnostics; Proposition 1 does not establish the gain of the deployed proxy or allocator.

Projection-wise correction control. Our DecDEC-style comparison is an accuracy-only emulation, with the same nominal residual-value and scale budget as coupled retrieval. For each expert record, exact top-k selection ranks input coordinates by mean absolute activation for the gate and up projections. Their selected residual columns are added before the SwiGLU nonlinearity. The cor-

Table 7: Absolute values for Figure 4. Comparands: the 2:1 bf16 ablation anchor (0.6491) and the exact-path Table 6 rows 7.8× (0.6421) and 10.3× (0.6364); intervals are the reported 95% paired-bootstrap CIs. Scores are batch-conditioned diagnostics (§3). Iso-traffic denotes the nominal residual budget; the DecDEC-style control emulates that budget for accuracy and does not measure packed-NF4 transport.

Alternative	Setting	Result vs. streamed coupled corrections
Static selection schedules (is dynamic selection necessary?)		
Weight-energy static score	iso-traffic	0.5694 (anchor base): -8.0 $[-8.3, -6.2]$
Corpus-profiled top-p	iso-traffic	-6.3 $[-7.3, -5.4]$ / -5.4 $[-6.4, -4.5]$ / -6.5 $[-7.6, -5.4]$ at champ. / 7.8× / 10.3×
Record size b=1→128	fixed bytes	0.654/0.651/0.647/0.649/0.633/0.634/0.623: graceful (-2.6 vs. b=16), 4.3 above calibrated-static (0.5796)
Record size, headline row	fixed bytes	0.6804/0.6713/0.6666 at b=16/64/128: -1.4, gentler than the anchor's -2.6
Calib.-profiled unequal re-division	static allocation	0.5709 / 0.6133 at 15.5×/16.0×; dynamic $+7.90$ $[+6.78, +9.01]$ / $+6.71$ $[+5.73, +7.71]$; both below uniform twins
Full-width corrected-activation selector	diagnostic	$+1.25$ BoolQ over the deployed proxy; no accuracy upper bound
Residual-norm factor dropped	ablation	$+0.02$ $[-0.72, +0.77]$: null
Resident static substitutes		
Rank-32 ΔW_2 factors	atop anchor	0.6454: no gain over 0.6491
D2-MoE-class full-width deltas	2×, resident-only	loses 18+ vs. unmerged
Full-width 2-bit deltas	resident-only	caps at ∼3.7× (≈7.3 GB); the stream reaches 16.0×
MiLo-u16 compensators	int4, quant. scale	0.7275: statics can excel off the merge regime
MiLo compensators, transferred	int4 (App. G)	$+0.04$ $[-0.29, +0.37]$: null; kurtosis policy transfers degenerately
Recovery fine-tuning		
Calibration-corpus LoRA	1.6 GPU-h	0.5525 / 0.4995 (int4/int3); more budget *worse*: CE 5.53→7.20
Fresh-C4 distillation (r=64)	21 GPU-h	0.5972 / 0.5776; never closes: 6.5 / 8.5 below KD+patches (Table 6)
Projection-wise corrections at nominal byte parity (340/340/129 channels/record)		
DecDEC-style, quantization	int4 / int3	$+0.25$ $[-0.20, +0.70]$ / -0.33 $[-0.89, +0.25]$: tie on quantization residue
DecDEC-style, merge/composed	nominal byte parity	0.5966 / 0.6009 / 0.5769 vs. 0.6491 / 0.6421 / 0.6364 (champ. / 7.8× / 10.3×); budget re-split 25/25/50: -0.97; residual-energy-weighted selection: $+0.84$, still -4.4
ΔW^2-only decoupling		0.3675, below bare (0.3822)

rected intermediate activation then supplies both the down-projection input and its mean-absolute-activation selection scores. Writing I_1, I_3, I_2 for the selected coordinates, the control computes

$$u = W_1^S x + \Delta W_1[:, I_1] x_{I_1}, \qquad v = W_3^S x + \Delta W_3[:, I_3] x_{I_3},$$
$$a = \mathrm{SiLU}(u) \odot v, \qquad y = W_2^S a + \Delta W_2[:, I_2] a_{I_2}.$$

The default widths $(340, 340, 129)$ select $(340 + 340)768 + 129(2048) = 786{,}432$ scalar residual values, equal to 128 coupled triplets. Residual precision is emulated with group-64 absmax NF4 and fp16 scales along each fetched column. Base products and intermediate rounding use bf16; residual products use fp32. The emulation forms residuals from a CPU-resident original model and does not measure DecDEC kernel latency or packed-NF4 transport. The additional residual-energy-weighted selector multiplies mean-square activation by squared residual-column norm; it is a diagnostic heuristic, not an oracle for nonlinear output error.

D Recovery-Spectrum Diagnostics

Spectrum comparisons. The averaging–RegMean spectrum crossing is observed in the pooled per-record energy diagnostic for all 48 layers. This is not an exact layer-global error curve.

Scope and provenance of the predictor. The tail mass in Equation 11 uses channel energies and the deployed selector's top-K, without bank construction, kernels, or patched evaluation. These statistics were computed on the task-text set with *original-model hidden states*; see the reproduction settings for the data protocol. Across nine bases, the deployed-selector tail ranks patched accuracy at Spearman ρ=0.63 (exact permutation p=0.08; leave-one-out 0.48–0.81), compared with only 0.26 within the six merge bases and 0.37 for the oracle tail across all nine. Excluding REAP and KD gives ρ=0.82, p=0.03. Within the K=128 family, the correlation is 0.71 including KD and 0.90 excluding it (n=5, p=0.08).

REAP has the smallest channel tail but only a moderate patched result because deleted-expert routing error is absent from the channel energies. KD has a larger tail than plain merging but the strongest merge-family result because its functional gain barely changes the spectrum. Bare accuracy itself ranks patched accuracy at ρ=0.81 across the nine bases, better than the tail; the tail wins only on the

Table 8: Recovery diagnostic: proxy-tail mass at each base's deployed K vs. measured patched accuracy. Here K is the per-record width; both tails use the diagonal energy approximation, and the oracle ranks exact channel energies within each record. Flagged rows are the two non-channel error classes: REAP's damage is routing-shaped (invisible to channel energies); KD's gain is functional (spectrum unchanged, Figure 3). Spearman ρ=0.63 over all nine (n=9, exact permutation p=0.08); 0.82 excluding the flagged rows (n=7, p=0.03); 0.90 within the K=128 family excluding the KD base (n=5, p=0.08; 0.71 with it). The oracle tail ranks at only ρ=0.37; the 4:1 base's proxy tail exceeds its oracle tail by approximately 8.4×, indicating energy missed by the deployed proxy (Table 16).

Base	K	Oracle tail	Proxy tail	Patched acc.
REAP50×QTIP *(routing)*	256	2.80×10^5	3.38×10^5	0.6580
QTIP-2b	128	4.49×10^5	5.13×10^5	0.7205
int3-all	128	6.37×10^5	7.21×10^5	0.7109
2:1 over QTIP (re-enc)	256	1.31×10^6	2.69×10^6	0.6427
2:1 bf16 (anchor)	128	1.66×10^6	3.20×10^6	0.6491
2:1 int4	128	1.71×10^6	3.27×10^6	0.6421
2:1 int4 + KD *(orthogonal)*	128	1.94×10^6	3.58×10^6	0.6625
RegMean 2:1	128	2.35×10^6	3.81×10^6	0.6440
4:1 int4	256	1.28×10^6	1.08×10^7	0.6002

channel-structured subsets in Table 9. The 4:1 base's large proxy-to-oracle gap records correctable energy missed by the deployed selector. These controls limit the tail's role to diagnosing channel correction, rather than replacing evaluation of candidate bases.

Table 9: Control: does bare accuracy rank patched accuracy as well as the proxy tail? Spearman ρ over the bases of Table 8. Two bare values are not printed and are proxied by their bf16 merge-only analogues (2:1 int4 ←0.3822, 4:1 int4 ←0.4706); any plausible value moves ρ(bare) only within 0.78–0.85. The lift (patched − bare) is anti-ranked by bare accuracy at $\rho\approx-0.98$, a ceiling effect, and only weakly by the tail (-0.4 to -0.6).

Subset	n	$\rho(-\text{proxy tail, patched})$	$\rho(\text{bare, patched})$
all nine	9	0.63	0.81
minus the two proxied bares	7	0.46	0.93
minus the two flagged bases (REAP, KD)	7	0.82	0.63
K=128 family	6	0.71	0.90
K=128 family minus KD	5	0.90	0.82

E MIXTRAL CROSS-MODEL RESULTS

Table 3 in the main text gives the complete operating-point comparison. The tables below provide channel-budget comparisons, per-task diagnostics, and evaluation conventions.

Table 10 shows that residual channel retrieval recovers a large fraction of the loss introduced by merging. The simple all-channel score reduces WikiText-2 perplexity from 7.341 to 5.714 and improves Avg3 from 0.625 to 0.671. The FP8 version closely matches the full-precision residual bank, which supports the memory hierarchy motivation: the cold bank can be stored in a low-precision format without eliminating the recovery.

The K=1024 vs. K=512 comparison confirms that the active channel budget matters; the first-order score does not clearly dominate the simpler one.

The full parity grid (Table 11) extends the second-model evidence to the suite diagnostics and generation. Conventions: the same eight tasks and flags as §3 (avg8); GSM8K n=500, 8-shot, greedy, no chat template (base model); token-PPL under one snapshot per block — the Sub-MoE rows follow the legacy convention of Table 10, whose original K=1024 run reproduced the 5.714 cell exactly (replicate Δ 3×10^{-5} relative), while the remaining rows share a fresh snapshot whose PTB value validates the pipeline (-0.1% vs. the printed cell; Wiki/C4 drift of -7.1%/$+5.6\%$ is dataset-snapshot drift, so cross-block PPL comparisons carry that caveat). Hook exactness (rel $\leq3.8\times10^{-3}$) held in every patched lane. At suite level the flip pair (a K=1024 control) separates only through patches: bare suite averages are chance-level (0.344/0.347) and cannot rank orderings; patched,

Table 10: Mixtral Sub-MoE batch-conditioned diagnostics. Lower is better for the perplexity diagnostic; higher is better for downstream accuracy. Avg3 is the mean of ARC-c, BoolQ, and HellaSwag.

Variant	Wiki PPL	PTB PPL	C4 PPL	ARC-c	BoolQ	HellaSwag	Avg3
Unmerged	3.598	15.418	6.409	0.572	0.840	0.720	0.711
Sub-MoE no residual	7.341	40.425	12.271	0.450	0.782	0.642	0.625
Delta-MoE, all→1024 simple	5.714	24.169	9.943	0.516	0.802	0.694	0.671
Delta-MoE, all→512 simple	6.255	28.573	11.079	0.482	0.800	0.686	0.656
Delta-MoE, all→1024 simple FP8	5.717	24.194	9.950	0.502	0.806	0.704	0.671
Delta-MoE, all→1024 first-order	5.990	25.411	10.788	0.508	0.798	0.708	0.671
Delta-MoE, seq-balanced first-order	6.003	25.139	10.812	0.512	0.790	0.704	0.669

usage-weighted leads on token-PPL (6.165 vs. 6.203) and GSM ($+2.0$, stderr ~ 1.9) while avg8 ties, so the reversal claim is scoped to output error, where the spectra made their prediction. In the $K{=}1024$ runs the FP8 bank has small observed suite and generation differences from its bf16 counterpart (avg8 $+0.06$, GSM -2.0 vs. stderr ~ 2.0) while halving fetch bytes, to 3.6% of an expert read at that deployment. In the refreshed grid, patches on the clean int4 base change Avg8 by $+0.07$ points (0.7195 to 0.7202) and leave GSM8K unchanged. On int3, the observed GSM8K gain is $+6.2$ points and the perplexity diagnostic changes from 4.01 to 3.91. These point estimates show a larger recovery at int3; a paired generation-difference interval is not reported.

The ladder rungs (Table 3) share these conventions with three further disclosures. QTIP Hessians are fp32 at batch size 8, uniform across the MQ/MC/MRQ/MX encodes. The Mixtral allocator uses a continuous floor/cap ($[256, 4096]$ at $p{=}1024$, the $p/4$–$4p$ mirror of Qwen's kernel tiers) and degenerates to fixed-p bit-for-bit when cap=floor=p (validation gate: wiki perplexity matched to 13 digits). The composed and extreme configurations were selected by the deployed-K tail estimator (usage-weighted, 4/4 convention×sort-key cells), the same estimator-guided procedure as the Qwen extreme row. The alloc-on-pruned control is a null (0.6024 vs. 0.6034 in the $K{=}1024$ runs), consistent with the limited benefit observed when reallocating corrections among surviving pruned experts. Iso-byte at $16.0\times$ holds by construction (both arms carry four QTIP-2b expert-equivalents per layer; ledger $15.997\times$ each).

F Systems: Kernels, Latency, and Serving

Scope of the results. Recovery is largest for merge error and smaller for accurate quantized bases. Patching surviving experts does not restore deleted experts' computations. Mixtral reproduces the recovery mechanism but reaches a shallower usable compression range. The measured transfer advantage over expert caching is in prefill; at the headline width, caching moves fewer bytes in the decode settings tested. Latencies below are per-forward proxies, and fused accuracy has measurable run-to-run variation. Queueing and link contention under serving load are unmeasured.

An early CPU-bank prototype paid $5.73\times$ the bare baseline's forward latency under a naive ΔW^2 layout; channel-major layout ($3.62\times$), packed $(\Delta W^1, \Delta W^3, \Delta W^2)$ triplets ($2.59\times$), and layer-grouped fetch ($1.83\times$) recovered it. The overhead is the granularity and layout of fetch/dequant/compute, not the active payload size.

Exact vs. fused paths. Table 6 prints reference-path (exact-math) numbers by convention. The rebuilt fused kernels (single-stage tiling to $p{\le}512$) pass their acceptance tests within measured run-noise envelopes: fixed $p{=}256$ on the stacked base is -0.41 $[-1.26, +0.43]$ against the exact twin; the power-of-2 tier constraint is null on the reference path (0.6845 vs. 0.6804, $+0.40$ $[-0.34, +1.14]$); and fused *global allocation* reproduces the stacked headline (0.6808 vs. 0.6804, $+0.04$; the paired-replicate CI $[-1.16, +0.41]$ spans zero; GSM8K 0.81 vs. 0.832). The allocator is iso-bytes exact (mean width 256.0 over 4,097 calls) and matches brute force within 0.006% at production scale. Fused accuracy lies within the reported intervals of the reference path but is not identical to it: fp32 atomic accumulation gives run-to-run nondeterminism (measured run-noise $+1.35$ $[+0.28, +2.48]$), which is the disclosed fused-path spread. An earlier kernel generation paid a width-dependent tax at wide p ($+1.7$ at $p{=}128$, $+3.0$ at $p{=}256$ on the 4:1 base, 0.5699 vs. 0.6002), traced to the wide-p launch configuration and resolved in the rebuild; we retain the exact-path printing convention for all Table-1 rows.

Table 11: Mixtral-8x7B full grid; likelihood-based metrics follow §3. Refreshed +patches rows use the deployed K=2048 (14.3% of channels); the FP8-bank, plain-mean/usage-weighted 2:1, and alloc-on-pruned control rows retain the earlier K=1024 deployment (7.1%; FP8 bank halves fetch bytes); GSM8K n=500; ‡ = legacy token-PPL convention (that of Table 10); remaining PPLs share the fresh snapshot (see text).

Configuration	avg8	GSM8K	Wiki-PPL
Unmerged bf16	0.7230	0.606	3.342
Sub-MoE 2:1 (bare)	0.6117	0.222	7.349‡
+ patches	0.6808	0.454	5.126‡
+ patches, FP8 bank	0.6692	0.398	5.717‡
2:1 plain-mean (bare)	0.3442	0.002	1617
+ patches	0.6241	0.212	6.203
2:1 usage-weighted (bare)	0.3470	0.000	6365
+ patches	0.6218	0.232	6.165
GPTQ int4 experts (bare)	0.7195	0.570	3.503
+ patches	0.7202	0.570	3.471
GPTQ int3 experts (bare)	0.7001	0.472	4.013
+ patches	0.7052	0.534	3.908
REAP-50% pruning (2.0×, bare)	0.6360	0.156	5.703
REAP-25% pruning (1.33×, bare)	0.6809	0.426	4.399
QTIP-2b experts (8.0×, bare)	0.7026	0.502	4.049
+ patches	0.7029	0.500	3.953
REAP-50%×QTIP-2b (16.0×, bare)	0.5975	0.118	6.390
+ patches	0.6035	0.120	6.277
+ patches + global alloc.	0.6024	0.130	6.336
2:1 usage-wt. over QTIP-2b (16.0×, bare)	0.3533	0.000	9521
+ patches	0.6031	0.192	6.649
+ patches + global alloc.	0.6040	0.222	6.616
REAP-25%×2:1×QTIP-2b (21.3×, bare)	0.3755	0.016	74.60
+ patches + tiered alloc.	0.5217	0.100	7.124

Table 12: Per-forward latency (ms; 30 iterations after 5 warmup, randomized inputs, `use_cache=False`) — per-forward proxies, not serving tokens/s; all columns run in the same eager research harness. Ratios compare these implementations, not the isolated cost of correction. Bare = the same compressed model with patches disabled (`--no-channel-patch`) through the stock HF per-expert loop, a different implementation from the fused path and therefore not a patches-off control (the patches-off cost is read from the fused path's counters; see *Latency control*); Ref. = exact-math reference path; Fused = rebuilt layer-grouped kernels, which also batch the resident forward per cluster. H200 runs the stacked 16.0× configuration; A100 runs the 15.5× (4:1+int4) configuration (stacked residents are cluster-local); both exercise identical kernel machinery.

Platform	Shape	Bare	Ref.+alloc	**Fused+alloc**	Fused fixed
H200	prefill b8/s128	607	3885	**1109**	859
H200	decode b1/s1	368	319	**181**	151
H200	decode b32/s1	467	612	**300**	202
A100	prefill b8/s128	1928	9709	**2410**	2219
A100	decode b1/s1	1341	788	**584**	444
A100	decode b32/s1	1672	2260	**925**	794

The fetched-scales full8 runs are a *jointly measured* operating point: the fused engine executes the stacked row end-to-end with streaming in-path at 0.6808 avg8 (+0.04 against the printed 0.6804; paired-replicate CI $[-1.16, +0.41]$) while the latencies of Table 12 and the +1.8% streamed-scales overhead are measured on the same engine; accuracy and transport are not extrapolated from separate systems. The fused results reported here use these rebuilt kernels. Per-forward latency across shapes and platforms is given in Table 12; the earlier dispatch-granularity prototype study above (5.7× → 1.83×) predates it. At accuracy within the interval above, the full 8-task suite evaluates in 1h33m fused+alloc vs. 5h45m on the reference path (3.7×, one H200).

Serving traffic and offload comparison. Measured on real routing traces from the 7.8× configuration (48 layers × 4,644 forwards; 571k per-token selections; the trace-collection run doubles as an independent replicate of the 7.8× row: avg8 0.6372, within the disclosed spread of 0.6421). Ours streams 0.393 MB per (expert, record) at p=128 NF4. The LRU simulation caches 56% of int4 experts (matched ∼8.4 GB resident HBM; 2.36 MB per expert); offload accuracy is the exact int4 model (0.7103 RTN / 0.7212 AWQ) versus our 0.6421.

Table 13: Streamed payload bytes per token: Delta-MoE patches vs. LRU whole-expert offloading at matched resident HBM. Patch values exclude residual scales, which add 6.25%. Measured at p=128; the p=256 column is exact arithmetic (fixed width doubles the bytes; global allocation is iso-bytes at mean width 256.0). Upper block: 56% int4 cache (∼8.4 GB) on the full8 trace. Lower block: the 16× block's own 3.63 GB residents, as a 25% int4 cache (2.36 MB/expert) or a 50% QTIP-2b cache (1.18 MB/expert), replayed on a BoolQ routing trace whose 56% anchor reproduces the upper block within ∼15% (cache slightly less favored); the QTIP-2b cache serves the row-2 model (0.7132). Misses are counted per distinct (layer, expert) per decode step against a per-layer LRU of capacity cache-fraction×128; decode cells list b=1/8/64, prefill cells 2k/8k tokens. A 2k-token prefill chunk touches 111 of 128 experts per layer, so both caches miss every access; even an optimal (any-policy) cache would need 1.30/0.37 MB per token at 3.63 GB with QTIP-2b experts.

Granularity	rec/layer	ours MB/tok (p=128)	(p=256)	LRU miss	offload MB/tok
decode b=1	8.0	151.0	302.0	2.3%	21.2
decode b=8	26.8	63.1	126.2	5.2%	21.4
decode b=64	59.1	17.4	34.8	25.7%	29.9
prefill 2k	107	0.99	1.98	—	—
prefill 8k	115	0.26	0.52	—	—
int4 cache 25%, 3.63 GB	8.0/27.1/61.3	151/63.9/18.1	302/128/36	19.4/48.8/99.3%	176/196/108
QTIP-2b cache 50%, 3.63 GB	8.0/27.1/61.3	151/63.9/18.1	302/128/36	4.1/9.1/54.2%	18.6/18.9/31.2
prefill 2k/8k, int4 cache 25%, 3.63 GB	—	1.02/0.27	2.05/0.54	100/100%	6.15/1.62
prefill 2k/8k, QTIP-2b cache 50%, 3.63 GB	—	1.02/0.27	2.05/0.54	100/100%	3.07/0.81

At the 16× block's own 3.63 GB (lower block of Table 13), an int4 expert cache holds only a quarter of the bank and thrashes (99% misses at b=64), moving 3–6× more bytes than our patches at p=128 and losing at $b\geq$8 even at p=256; a QTIP-2b expert cache holds half the bank, moves 18.6/18.9/31.2 MB per token, and serves the more accurate row-2 model, so at this residency it is a favorable accuracy/traffic point for small-batch decode. Our traffic is closer at $b\approx$64, but remains higher at the headline p=256; our measured traffic advantage is prefill: a prefill chunk touches nearly every expert, so both caches miss every access and the QTIP-2b cache streams 3× our prefill bytes at p=128 (1.5× at p=256); even an optimal cache could not go below 1.30/0.37 MB per token at 3.63 GB, above our p=128 stream. Sweeping the cache budget downward, the QTIP-2b cache beats the p=128 stream at b=1 down to ∼1.2 GB and at b=8 down to ∼2.3 GB, while at b=64 the stream wins at every budget ≤3.63 GB (the cache's b=64 traffic saturates near 54 MB per token, a full reload of the ∼61 distinct experts a 64-token step touches).

Two robustness checks. *Locality.* Randomly permuting the order of decode steps (an interleaved-request proxy) leaves our stream unchanged by construction and raises the QTIP-2b cache's traffic at 3.63 GB from 18.6/18.9/31.2 to 80.1/42.6/33.8 MB per token (b=1/8/64); permuting tokens as well (independent requests, the worst case for both sides) gives ours 151/98.4/26.3 at p=128 against the cache's 80.1/83.3/79.0. So most of the cache's small-batch advantage is within-document temporal locality, yet it keeps $b\leq$8 under full interleaving and we keep b=64 by 3×. *Second trace.* Replaying the routing trace of the 4:1 rows (571k tokens) reproduces every LRU cell within ∼15% (e.g., QTIP-2b at 3.63 GB, b=1: 15.7 vs. 18.6 MB); at the deployed p=256 the QTIP-2b cache there moves fewer bytes than our stream at every decode batch (24.5 vs. 34.8 at b=64), while ours wins prefill (1.37 vs. 2.05 MB) and wins b=64 once the budget drops to 1.81 GB. Remaining caveats: the cache uses a fixed per-layer allocation, and multi-tenant contention for the cache and the link is unmeasured. Offload composes with our patches: a cached expert can itself be a patched compressed expert.

Latency control. Table 12's Bare column is the stock HF per-expert loop, a different implementation from the fused path, so it is not a patches-off control; disabling patches under the grouped-forward flags falls back to the same stock loop (1878 vs. 1880 ms at b=8, s=128 on A100), so no same-implementation patches-off run exists. An estimate of the correction cost comes from the fused path's own counters: on A100 (PCIe Gen4, 23.5 GB/s pinned host-to-device) the p=256 patch work adds ≈2.3× the bare grouped forward at prefill (b=8, s=128: 1.56 s over 0.63–0.68 s) and

$\approx$2.4$\times$ at single-token decode (0.32 s over 0.13 s); at p=128 the additions are $\approx$1.1$\times$. End-to-end, the fused patched forward is $+19\%$ against the stock HF path at prefill for p=256 (-27% for p=128) and 2.9–5$\times$ faster at b=1 decode because the grouped kernel is a faster MoE implementation than HF's loop. On an H100 (Gen5, 54.4 GB/s pinned), the fused fixed-p=256 path adds $+1.4\%$ to the b=8, s=128 forward (836 vs. 824 ms) and tiered allocation 1126 ms; the stock decode path is 2–3$\times$ slower than the fused patched path on both platforms.

Generation beyond GSM8K. Table 14 adds MATH-500 (4-shot Minerva prompt, symbolic-equivalence scoring) and IFEval (541 prompts) under the GSM8K protocol (chat template with `/no_think`, greedy, 512 new tokens; fused path). The stacked 16$\times$ row has similar point estimates to 8$\times$ bare QTIP-2b on both (MATH-500 0.484 vs. 0.486; IFEval prompt-strict 0.599 vs. 0.599), while the iso-byte bare pruning row at the same 3.63 GB collapses (0.024; 0.224), the same multiple-choice-hides-the-cliff pattern as GSM8K. Patches lift QTIP-2b on MATH-500 by $+3.0$ points and produce a small observed change on IFEval relative to the bare QTIP-2b baseline; the unmerged model was not evaluated on this task. Per-row unpaired s.e. is $\approx$2.2 points (MATH-500) and $\approx$2.1 (IFEval prompt-level; 0.7/1.8 for the pruning row); no paired CIs exist for these rows, so the $+3.0$ point difference should be read as a point estimate, not a statistically established gain. The unmerged bf16 reference and a patched pruning row were not completed.

Table 14: Generation benchmarks beyond GSM8K (Qwen3-30B-A3B; MATH/IFEval on the fused path). GSM8K repeats Table 1; for the stacked row it is the reference-path value (the separately reported fused value is 0.810). IFEval: prompt-level / instruction-level, strict and loose.

Row (Table 1)	GSM8K	MATH-500	IFEval p-strict	i-strict	p-loose	i-loose
QTIP-2b (bare), 8.0$\times$	0.862	0.486	0.599	0.713	0.627	0.735
QTIP-2b + patches, 8.0$\times$	0.878	0.516	0.595	0.715	0.625	0.735
REAP$\times$QTIP-2b (bare), 16$\times$	0.156	0.024	0.224	0.359	0.237	0.372
Stacked 16$\times$ (row 11)	0.832	0.484	0.599	0.711	0.617	0.725

Per-task accuracies. Table 15 prints the eight task scores behind printed rows for which the evaluation JSONs were at hand. Three rows are independent replicates of printed cells and disclose build-to-build spread: unmerged 0.7237 (printed 0.7240), QTIP-2b + patches 0.7228 (printed 0.7205), and the stacked row's first run 0.6790 (printed 0.6804, its replicate); the fused p=256 member of the row-10 family reads 0.6373 against the reference path's 0.6427. In the printed stacked replicate, the average gain over fixed-width patched pruning is 2.10 points: MMLU contributes 2.075 points to that average, while the other seven tasks average a net $+0.029$ points. This decomposition concerns the printed 0.6790 replicate, not the separate 0.6804 run. All patched task scores here follow the batch-conditioned evaluation of §3.

Table 15: Per-task batch-conditioned scoring accuracies (`acc_norm` for ARC-c/e, HellaSwag, OBQA; `acc` otherwise).

Row	ARC-c	ARC-e	BoolQ	HSwag	MMLU	OBQA	RTE	WG	avg8
unmerged bf16 (replicate)	.564	.797	.888	.776	.769	.444	.838	.714	0.7237
QTIP-2b (bare), row 2	.541	.792	.879	.747	.752	.418	.866	.710	0.7132
QTIP-2b + patches (replicate)	.559	.793	.884	.768	.777	.454	.830	.719	0.7228
REAP pruning, 2$\times$	.483	.714	.881	.762	.587	.424	.805	.696	0.6689
REAP$\times$QTIP-2b (bare), row 8	.460	.684	.869	.730	.511	.406	.715	.697	0.6339
REAP$\times$QTIP-2b + patches, row 9	.468	.694	.873	.759	.578	.422	.769	.701	0.6580
2:1 int3 + patches, 10.3$\times$	.432	.700	.859	.662	.592	.408	.755	.684	0.6364
Stacked 16$\times$, row 11 (first run)	.486	.710	.851	.737	.744	.426	.783	.695	0.6790
2:1 over QTIP-2b + patches, fused p=256	.464	.684	.826	.647	.677	.380	.762	.658	0.6373
Static keep-A allocation control	.443	.681	.853	.627	.589	.372	.751	.644	0.6200
4:1 int4, fixed p=256, 15.5$\times$	.399	.598	.762	.597	.552	.372	.625	.651	0.5694

G Reproduction Settings

Statistical reporting. Where independent replicates exist, tables retain the first-logged run. Reported paired-replicate differences and their intervals can therefore differ from subtraction of the printed point estimates. We distinguish those intervals from printed-row arithmetic and from unpaired sampling uncertainty.

Table 16: Proxy energy capture relative to per-record exact-energy selection, using original-model hidden states and squared routing weights. The reported median is across layers.

Regime	Proxy capture (median)	Worst layer
2:1 merge	70%	0.38–0.61
4:1 merge	54%	0.006

Global allocation raises median capture by 5–7 points. Appendix A reports separate local coherence and restricted exhaustive-error diagnostics; these measurements do not certify the full layer-global guarantee.

Calibration data. The AWQ-style activation-aware baseline uses per-channel scales $s_j = (\mathbb{E}[x_j^2])^{\alpha/2}$ from calibration Gram diagonals, with α grid-searched over $[0.3, 0.7]$ per bit-width.

Unless noted, calibration-derived artifacts use one file: `c4-train.00000-of-01024.json` (C4 English shard; 2.26 MB; 971 texts >200 characters; ∼479k tokens under the Qwen3 tokenizer), loaded with the HC-SMoE `get_calib_dataloder("c4")` loader, shuffle seed 42. Consumers: (1) usage weights for merging (routing statistics); (2) AWQ-style scales (Gram diagonals); (3) RegMean Grams; (4) calibration-profiled static channel sets; (5) GPTQ calibration (256 texts); (6) healing corpus (Table 7; multi-epoch: 2.5M tokens at the 1.6-GPU-h budget); (7) REAP saliency (routing-weighted expert-output norms on the same C4 calibration shards). Exceptions using fresh, disjoint C4 shards: QTIP Hessians (4M tokens) and the KD-healing corpus (Appendix G). Runtime channel selection is conditioned on the evaluation batch only, never on calibration data. Spectrum and capture diagnostics use saved inputs to the original model's MoE layers, collected from task text independently of C4 calibration. The capture script uses ordered task-text prefixes and retains all positions in its padded input batches; the offline analysis reconstructs routing with the original router. These are local reconstruction diagnostics on a fixed captured input set, not measurements on deployed-model hidden states. Routing-weighted spectrum energies use squared routing weights.

Environment: torch 2.9.1+cu128, transformers 4.51.3, tokenizers 0.21.4, accelerate 1.13.0, lm-eval; batch 16; `acc_norm` for ARC-c/e, HellaSwag, OBQA. Standard channel flags for all patched rows: `--channel-p 128 --channel-candidate-p 768 --channel-delta-format native --channel-selection batch_resnorm_activation --reuse-base-activation --patch-residual-form delta --fuse-gate-up-delta --channel-bank-device cpu --channel-bank-pin-memory --channel-pack-cpu-bank --channel-bank-int4 --channel-int4-free-native --channel-int4-codebook nf4 --channel-unsorted-topk --channel-cache-gpu-metadata --layer-grouped-channel-forward --layer-share-cluster-base --fused-layer-channel-kernels`. Champion: pairs grouping JSON plus the flags above. Quantized bases add `--quantize-merged-bits {3,4} --quantize-merged-group 128`; the 15.5× row uses the 4:1 grouping JSON with `--channel-p 256`; global-allocation rows add `--channel-global-alloc` (per-layer iso-bytes top-$K$, floor/cap as stated in §2.2, token-count-weighted ranking, the `--channel-global-weight tokens` default, reference path). GPTQ rows: experts-only, g=128, activation ordering, evaluated through the same harness via override directories. MiLo rows: faithful port of the public MLSys'25 code (HQQ g=64 base; iterative quantize-then-compensate; kurtosis rank policy computed on Qwen3 weights, 88% of expert matrices at rank 0, average rank 1.6); rank-0 ablation gives the HQQ-only rows, and uniform-rank ablations (u16/u32) give the policy-independent compensator rows. Note the kurtosis policy transfers degenerately to Qwen3 (88% rank-0), so the null result for MiLo's compensators (Table 7) partly reflects the policy's fit to this model; the HQQ-only rows isolate the base quantizer. QTIP rows: vendored bitshift-trellis codebook (L=16, K=2, V=2, symmetric quantlut; 2/3 bits per weight), per-expert Hessians from 4M fresh C4 tokens (fp64), expert-batched LDLQ/Viterbi, *no fine-tuning*; the port is bit-exact against upstream in equivalence tests, and composition rows rebuild the bank against the dequantized QTIP resident (weights folded to the original basis).