SPARSEDIAG: STATE-DEPENDENT DIAGNOSIS AND REPAIR FOR LOW-BIT LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Strong post-training quantization (PTQ) checkpoints can often be improved further, but deciding where to spend the additional optimization budget is difficult. We examine this problem using reconstruction and language-modeling objectives across six LLMs and two calibration corpora. Their relative perplexity varies across models and test domains, and independent layer-wise probes show that the set of beneficial updates changes with the PTQ checkpoint. We propose SparseDiag, a post-PTQ framework that tests candidate updates one layer at a time, ranks them by held-out full-model negative log-likelihood, and repairs a selected layer using the PTQ backend's existing trainable parameters. The model is re-diagnosed after each accepted repair. On four Llama and Qwen models, three rounds of refinement improve both WikiText-2 and C4 perplexity from strong W4A16 checkpoints. The results also expose two limits: extended full-model CE training can degrade accuracy severely, and improvements on small calibration splits need not transfer to the full test sets. SparseDiag provides a practical way to study and exploit remaining repair opportunities without assuming a fixed layer ranking.

1 INTRODUCTION

Post-training quantization (PTQ) reduces the memory and inference costs of large language models (LLMs). Methods such as GPTQ, AWQ, SmoothQuant, and more recent reconstruction-based approaches have made low-bit checkpoints increasingly accurate (Frantar et al., 2023; Lin et al., 2024b; Xiao et al., 2023; Shao et al., 2024; Liu et al., 2025; Ding et al., 2025; Wang et al., 2026b). Their central problem is how to produce a good quantized model. We ask a related question that arises *after* that process: given a completed PTQ checkpoint, which layer is still worth updating?

The answer need not follow from the layer's original quantization error. Reconstruction alters both intermediate representations and the errors passed to later layers. A layer that was sensitive at initialization may offer little benefit from another update, while a modest local error can still have a consequential effect on the model's final predictions. Existing PTQ procedures typically specify the layers or blocks to optimize as part of their schedule (Li et al., 2021; Shao et al., 2024; Ding et al., 2025; Wang et al., 2026a;b). They do not directly measure the benefit of a further update at the resulting checkpoint.

The choice of training loss adds another complication. Reconstruction mean squared error (MSE) encourages the quantized model to reproduce a full-precision teacher's representations; next-token cross-entropy (CE) instead optimizes language modeling on the calibration data. In controlled comparisons from identical quantized initializations, their relative effect depends on the model and corpus. CE may improve perplexity on the calibration domain without improving every other domain. Choosing a task loss, therefore, does not answer the separate question of which layer to refine.

We investigate that question through *residual repairability*: the reduction in full-model loss obtained by a specified additional layer update from the current checkpoint. On Llama-3-8B, independent MSE-trained layer candidates pass the same held-out NLL check in 22 of 32 layers at quantized initialization, 11 after SliderQuant, and 6 after single-layer refinement. The eligible layers also change location. These counts characterize the tested checkpoints under a fixed probe, not an inherent importance ranking for each layer. They nevertheless show why an initial sensitivity order is a poor substitute for examining the state one actually intends to refine.

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

We introduce **SparseDiag** to make that examination part of post-PTQ refinement. From a common accepted checkpoint, the method trains a temporary candidate for every layer using only the PTQ backend's native learnable parameters. It ranks candidates by their effect on held-out *full-model* negative log-likelihood (NLL), checks the sign of the gain on a separate split, and then restores the checkpoint to repair the selected layer. An accepted repair becomes the starting point for the next diagnosis. The diagnosis loss and the final repair loss may differ: MSE-trained candidates, for example, are selected by their NLL gain, not by the magnitude of their reconstruction error. Pretrained dense weights remain frozen, and no diagnostic adapter is added.

Our contributions are threefold:

- We compare reconstruction and language-modeling objectives across six models and two calibration corpora, and use controlled layer-wise probes to characterize residual repair opportunities at different PTQ checkpoints.
- We formulate residual repairability as the measured benefit of a specific update at the current state and introduce SparseDiag, which separates candidate training, held-out selection, selective repair, and re-diagnosis.
- We evaluate three diagnosis–repair objective pairs across several starting states and a second PTQ backend. Selective refinement improves WikiText-2 and C4 perplexity after strong W4A16 PTQ on four models; we also report cases where longer CE training or calibration-based acceptance fails.

2 MOTIVATION

Additional optimization after PTQ involves two decisions: the loss used to train an update and the layer to which the update is applied. We examine these decisions separately. The first experiment compares MSE and CE with the trainable parameters and budget held fixed; the second measures how many independent layer updates remain beneficial at different checkpoints.

2.1 OPTIMIZATION OBJECTIVES EXHIBIT MODEL- AND DATA-DEPENDENT BEHAVIOR

Reconstruction MSE and CE offer different signals. The former matches a frozen teacher's representations; the latter adjusts the model's next-token predictions to the calibration corpus. Improving either training objective does not, by itself, establish a benefit on another text distribution.

For a controlled comparison, we refine identical Initial State W4A16 checkpoints using either full-model final-hidden MSE or next-token CE. Within each calibration setting, both runs use the same trainable parameters, training budget, and evaluation procedure. We alternate the calibration corpus between WikiText-2 and C4, then evaluate on multiple domains. Figure 1 reports $R_{\mathrm{CE/MSE}} = \mathrm{PPL}_{\mathrm{CE}}/\mathrm{PPL}_{\mathrm{MSE}}$; values below one mean that CE yields lower perplexity.

The comparison is mixed. CE often performs better on the calibration domain, but on several Llama-model settings that advantage is reduced or reversed on other corpora. Changing the calibration corpus also changes the relative result. Qwen models in this comparison generally show broader CE gains. Thus, even with the same quantization setup, the value of the optimization signal depends on the model and data. This experiment begins from the same *initial* state in each pair; it does not imply that the preferred objective will remain unchanged at later PTQ checkpoints.

The ratio is an end-to-end evaluation, not a measure of the reconstruction error at any individual layer. It establishes that the two training signals can lead to different model behavior, but cannot tell us which local update would improve a checkpoint that has already been reconstructed. That requires a separate layer-wise test.

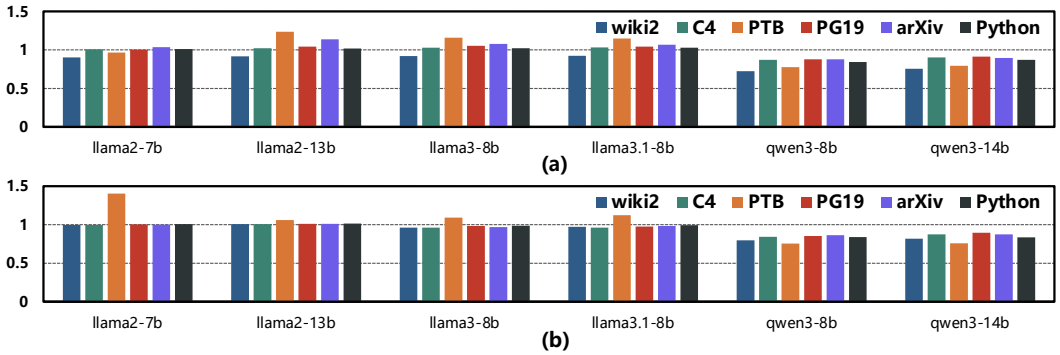

Figure 1: **Objective response across six W4A16 LLMs.** Test-set $\mathrm{PPL}_{\mathrm{CE}}/\mathrm{PPL}_{\mathrm{MSE}}$ after independent full-model refinement; values below one favor CE. (a) WikiText-2 and (b) C4 calibration. Ratios are compared within each model and test corpus.

2.2 Repair opportunities become sparse after stronger PTQ states

The second decision is where to spend an additional update. The original quantization sensitivity cannot answer this reliably: it describes the effect of quantizing a component, whereas the question after PTQ concerns the effect of *updating* it. To measure that effect, we restore the checkpoint before each independent layer probe, freeze the other layers, and train only the selected layer's existing quantization parameters.

For each of the three Llama-3-8B checkpoints in Figure 2, every temporary candidate receives 32 *MSE* updates. We then evaluate the candidate's *full-model NLL* on two held-out splits; a layer is eligible only when both gains are positive. The eligible counts fall from 22/32 at Initial State to 11/32 after SliderQuant and 6/32 after Single-layer refinement. The surviving layers are distributed across the network rather than confined to a fixed window.

The counts depend on the checkpoint, training operator, and calibration sample. In particular, "eligible" means that a *specific trained candidate* improves held-out NLL; it does not mean that every possible update to that layer would help. The 96/16/16 calibration split and the 32-update probe are kept the same across the three state scans. With that comparison fixed, the changing sets reflect the interaction between the starting checkpoint and the available local update. A ranking made at initialization therefore need not remain useful once PTQ has changed the model.

The use of MSE for this probe is deliberate: it constructs a candidate against a fixed teacher target, but the selection rule still asks whether the *whole quantized model* improves. This distinction permits MSE-based diagnosis to be paired with CE-based repair without equating local reconstruction error with final language-modeling quality.

2.3 Implication: diagnosis should precede repair

The first observation shows that a candidate's training loss and its effect on language modeling should not be conflated. The second shows that useful candidates change with the starting state. Together, they motivate testing prospective layer updates before committing them. SparseDiag follows this procedure and repeats it after each accepted repair. The cross-checkpoint probes motivate re-diagnosis, although they are not themselves a matched-budget comparison against a static ranking.

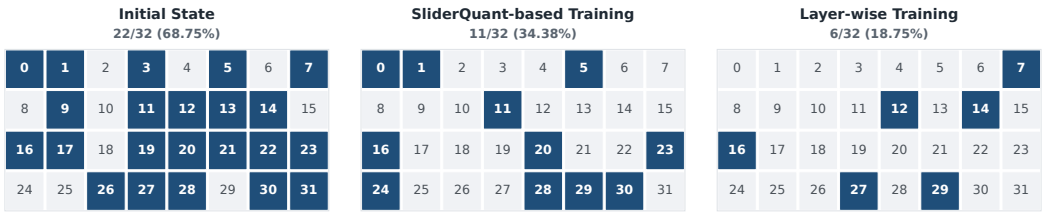

Figure 2: **MSE-diagnosed repairability across Llama-3-8B PTQ states.** Dark cells denote layers whose 32-update MSE candidates improve held-out full-model NLL on both splits: Initial State (22/32), SliderQuant (11/32), and Single-layer (6/32).

3 METHOD

SparseDiag refines an existing PTQ checkpoint through independent candidate training, held-out selection, and selected-layer repair. All temporary candidates start from the same checkpoint; the scan therefore does not accumulate updates in layer order. We evaluate each candidate by its effect on full-model NLL, then train the selected repair from the restored checkpoint. Once accepted, that repair becomes the starting point for the next scan.

3.1 RESIDUAL REPAIRABILITY

Let Θ denote the current quantized model obtained from an existing PTQ procedure. We assume that the PTQ method provides a set of native trainable quantization parameters, such as clipping factors or low-rank quantization parameters, while the pretrained dense weights remain frozen.

For layer l, we define a candidate refinement operator $\mathcal{A}_l$ that updates only the trainable parameters associated with that layer. For a fixed evaluation set $\mathcal{D}_{\text{eval}}$, the residual repairability of layer l is defined as:

$$\Delta_l(\Theta) = \mathcal{L}_{\text{NLL}}(\Theta; \mathcal{D}_{\text{eval}}) - \mathcal{L}_{\text{NLL}}(\mathcal{A}_l(\Theta); \mathcal{D}_{\text{eval}}). \tag{1}$$

A positive value indicates an improvement on the specified evaluation data, not necessarily on a different distribution.

Unlike initial layer sensitivity, which measures the damage from quantization, repairability measures the benefit of an additional update. It depends on the checkpoint, update operator, and evaluation data, and must be estimated from candidate updates rather than inferred from the initial quantization error.

3.2 SPARSEDIAG: DIAGNOSIS BEFORE REPAIR

At refinement round r, SparseDiag starts from the current accepted checkpoint Θ_r.

Layer-wise candidate generation. For each transformer layer l, SparseDiag creates an independent candidate $\widetilde{\Theta}_{r,l}$ by restoring Θ_r and updating only layer l.

Other layers and pretrained dense weights remain frozen. Candidates use either reconstruction MSE or language-modeling CE.

For MSE-based diagnosis, we optimize:

$$\mathcal{L}_{\text{MSE}}^{(l)} = \mathbb{E}_{x \sim \mathcal{D}_{\text{tr}}}[\|h_l^Q(x) - h_l^{FP}(x)\|_2^2], \tag{2}$$

where h_l^Q denotes the candidate representation with the current quantized prefix as input, and h_l^{FP} denotes the corresponding frozen full-precision teacher target.

Alternatively, CE diagnosis directly minimizes the language-modeling loss:

$$\mathcal{L}_{\mathrm{CE}} = -\mathbb{E}_{x\sim\mathcal{D}_{\mathrm{tr}}} \sum_t \log p_{\widetilde{\Theta}_{r,l}}(x_t \mid x_{<t}). \tag{3}$$

Candidate training and selection are deliberately separate: an MSE-trained candidate is judged by its full-model NLL gain, not by its local reconstruction loss. The candidate is evaluated in the complete quantized network, with its other layers left at the accepted checkpoint. A lower local reconstruction loss is therefore not, by itself, a sufficient selection criterion; the resulting change must reduce the model's prediction loss on held-out examples.

Held-out diagnosis and selection. SparseDiag evaluates each candidate by its full-model negative log-likelihood (NLL) gain.

The calibration data are divided into three disjoint subsets:

$$\mathcal{D}_{\mathrm{tr}}, \mathcal{D}_{\mathrm{rank}}, \mathcal{D}_{\mathrm{accept}}.$$

The first subset trains candidates, while the latter two are held out from training and used for candidate ranking and confirmation.

For candidate l, we compute:

$$g_{r,l}^{rank} = \mathcal{L}_{\mathrm{NLL}}(\Theta_r; \mathcal{D}_{\mathrm{rank}}) - \mathcal{L}_{\mathrm{NLL}}(\widetilde{\Theta}_{r,l}; \mathcal{D}_{\mathrm{rank}}), \tag{4}$$

$$g_{r,l}^{accept} = \mathcal{L}_{\mathrm{NLL}}(\Theta_r; \mathcal{D}_{\mathrm{accept}}) - \mathcal{L}_{\mathrm{NLL}}(\widetilde{\Theta}_{r,l}; \mathcal{D}_{\mathrm{accept}}). \tag{5}$$

A candidate is eligible for selection when both held-out gains are positive:

$$\mathcal{E}_r = \{l : g_{r,l}^{rank} > 0 \wedge g_{r,l}^{accept} > 0\}. \tag{6}$$

Among eligible candidates, SparseDiag selects:

$$l_r^* = \arg\max_{l\in\mathcal{E}_r} g_{r,l}^{rank}. \tag{7}$$

The ranking split orders candidates, while the confirmation split checks the sign of their gain on different calibration examples. The selection is made *among candidates that pass both checks*; a candidate with the highest ranking gain is not chosen if its confirmation gain is non-positive. Both splits are reused across candidates and rounds, so this is an empirical screen rather than an independent generalization guarantee. The final repair is evaluated again because it is trained from the restored checkpoint and may use a different objective than the diagnostic candidate.

Selective repair and iterative refinement. After selecting layer l_r^*, SparseDiag restores Θ_r and trains only the selected layer using repair objective u, producing a tentative checkpoint:

$$\widehat{\Theta}_{r+1} = \mathcal{R}_{l_r^*}^{u}(\Theta_r), \tag{8}$$

where u denotes the repair objective. The tentative repair is checked on the held-out acceptance split against Θ_r; only a positive improvement is committed as Θ_{r+1}. Otherwise, the previous checkpoint is retained and the current run terminates.

We evaluate MSE→MSE, MSE→CE, and CE→CE, denoting diagnosis→repair objectives.

After each accepted repair, previous scores are discarded and layer diagnosis is repeated from the updated checkpoint.

The run stops when no candidate is eligible, a tentative repair fails acceptance, or the round budget is exhausted.

Algorithm 1 SparseDiag: State-dependent Diagnosis and Repair

Require: PTQ checkpoint Θ_0; disjoint $\mathcal{D}_{\mathrm{tr}}, \mathcal{D}_{\mathrm{rank}}, \mathcal{D}_{\mathrm{accept}}$; diagnosis objective d; repair objective u; maximum rounds R

for $r = 0, \ldots, R-1$ **do**
  for each transformer layer l **do**
    Restore Θ_r; train independent candidate $\widetilde{\Theta}_{r,l}$ using d
    Measure full-model rank and acceptance gains
  end for
  Form $\mathcal{E}_r$ from candidates positive on both splits
  if $\mathcal{E}_r = \varnothing$ **then**
    break
  end if
  Select l_r^* by the largest ranking gain in $\mathcal{E}_r$
  Restore Θ_r; train tentative repair $\widehat{\Theta}_{r+1}$ using u
  if repair improves held-out acceptance NLL **then**
    Commit $\Theta_{r+1} \leftarrow \widehat{\Theta}_{r+1}$
  else
    Retain Θ_r and **break**
  end if
end for
return last accepted checkpoint

4 EXPERIMENTS

We evaluate whether SparseDiag can recover additional accuracy from different PTQ checkpoints, how diagnosis and repair objectives affect the resulting updates, and when calibration-based selection fails to transfer. We report full-test WikiText-2 and C4 perplexity (PPL); the cross-domain objective comparison that motivates the method is shown in Section 2.

4.1 EXPERIMENTAL SETUP

Models and starting states. We study W4A16 quantization on Llama-2-7B/13B, Llama-3-8B, Llama-3.1-8B, and Qwen3-8B/14B. Following Figure 2, we use three checkpoint labels: *Initial State* (quantized initialization before PTQ refinement), *SliderQuant* (checkpoint after the full SliderQuant PTQ procedure), and *Single-layer* (layer-wise refined checkpoint). These labels distinguish the starting states, not three different model architectures. SparseDiag optimizes the existing LoRA/LWC parameters and freezes pretrained dense weights. For OmniQuant, it updates only native LWC parameters without adding LoRA; the two backends have different parameterizations.

Calibration and evaluation protocol. The full-model objective comparison uses 128 sequences of 2048 tokens and 100 updates per run, with 50 updates at each quantization-rate stage. The two objectives are full-precision-teacher final-hidden MSE and next-token CE. We calibrate on WikiText-2 or C4 and evaluate PPL on WikiText-2/103, C4, PTB, PG19, arXiv text, and Python code; bounded evaluations are used for the latter domains where applicable. The corresponding CE/MSE ratios are presented in Figure 1. Absolute PPL is not compared across different tokenizers.

For Figure 2, the 128-sequence pool is split into 96 training, 16 ranking, and 16 confirmation sequences. All three state scans train independent candidates with 32 *MSE* updates per layer from identical within-state checkpoints; eligibility requires positive full-model NLL gains on both held-out splits. The main refinements compare MSE→MSE, MSE→CE, and CE→CE (diagnosis→repair). Initial State experiments use eight repair rounds; SliderQuant and Single-layer experiments use up to three. Full-test sets are evaluated only after the trajectory is fixed; layer indices are zero-based.

4.2 SELECTIVE REFINEMENT ACROSS PTQ STATES

Main results. Table 1 unifies the completed Initial State, SliderQuant, and Single-layer comparisons. Within every model–state group, all objective pairs start from the same checkpoint. Ini-

Table 1: **SparseDiag across W4A16 starting states.** Full-test PPL (lower is better); objective pair denotes diagnosis→repair. Within a model–state group all arms share the baseline. R: eight rounds (Initial State), up to three (SliderQuant/Single-layer). Bold: lowest PPL per model–state–dataset group.

Model	Eval.	Baseline	MSE→MSE	MSE→CE	CE→CE
Initial State ($R = 8$)					
Llama-2-7B	Wiki2	6.0990	6.0814	**5.8374**	5.8586
	C4	8.0626	8.0469	**7.9621**	8.1045
Llama-3-8B	Wiki2	8.2548	7.7383	7.0982	**6.9953**
	C4	11.8212	11.1954	11.0014	**10.8942**
Qwen3-8B	Wiki2	11.9506	10.8953	8.7189	**8.4610**
	C4	15.5908	14.6772	13.4544	**13.4484**
SliderQuant ($R \leq 3$)					
Llama-2-7B	Wiki2	5.7265	5.6673	5.6118	**5.5623**
	C4	7.5682	7.5021	7.5013	**7.4959**
Llama-3-8B	Wiki2	7.0295	6.9694	6.7776	**6.7461**
	C4	10.4507	**10.3240**	10.3310	10.3482
Llama-3.1-8B	Wiki2	7.0669	6.9662	6.7957	**6.7401**
	C4	10.4931	**10.3080**	10.3624	10.3759
Qwen3-8B	Wiki2	10.4845	10.2800	8.5487	**8.2824**
	C4	14.1249	14.0376	13.0127	**12.8742**
Single-layer ($R \leq 3$)					
Llama-2-7B	Wiki2	5.7205	5.7120	5.6330	**5.5586**
	C4	7.5959	7.5944	7.5918	**7.5893**
Qwen3-8B	Wiki2	10.6185	10.4344	8.4755	**8.4550**
	C4	16.0134	15.9330	**13.9362**	14.2081

tial State runs use eight selective repair rounds, whereas SliderQuant and Single-layer runs use up to three; their results should therefore be compared within each state rather than interpreted as matched-budget comparisons across states.

For the four models with complete SliderQuant comparisons, all three objective pairs improve both WikiText-2 and C4 over the starting checkpoint. On Llama-3-8B, CE→CE gives the lower WikiText-2 PPL (6.7461), while MSE→MSE gives the lower C4 PPL (10.3240). On Qwen3-8B, CE→CE reduces WikiText-2/C4 PPL from 10.4845/14.1249 to 8.2824/12.8742. The relative results depend on the model and evaluation corpus.

The Initial State and Single-layer groups show how results change with the starting checkpoint and repair budget. In particular, CE→CE from Initial State improves WikiText-2 on Llama-2-7B but increases C4 PPL relative to its baseline (8.1045 versus 8.0626). Both evaluation domains are therefore necessary to characterize the trajectory.

Additional completed configurations. The SliderQuant Llama-2-13B MSE→MSE run changes WikiText-2/C4 from 5.0304/6.6716 to 5.0256/6.6707; its CE repair arms and the Qwen3-14B SliderQuant arms are unavailable. From a Single-layer checkpoint, Llama-3.1-8B reaches 7.1298/10.5761 with MSE→MSE and 6.8491/10.5460 with CE→CE, starting from 7.2166/10.6565; the MSE→CE arm is incomplete. For Qwen3-14B, Single-layer MSE→MSE improves 9.2383/12.8109 to 9.1236/12.7569. Missing arms are not interpreted as accuracy failures.

Transfer to another PTQ backend. Starting from the official OmniQuant Llama-2-7B W4A16 checkpoint (WikiText-2/C4: 5.7439/7.6499), three rounds yield 5.7389/7.6464 with MSE→MSE, 5.6916/7.6376 with MSE→CE, and 5.6774/7.6228 with CE→CE. Only OmniQuant's existing LWC parameters are updated. This demonstrates an improvement with a second native parameterization, although we have not measured its full-layer repairability distribution.

Diagnosis and repair objectives. Candidates trained with MSE→CE are ranked by held-out *full-model NLL gain*, not local reconstruction error. The resulting layer trajectories differ across objectives: Llama-3-8B selects $1 \rightarrow 29 \rightarrow 20$ with MSE→CE and $1 \rightarrow 3 \rightarrow 31$ with CE→CE; Qwen3-8B selects $19 \rightarrow 35 \rightarrow 1$ and $16 \rightarrow 35 \rightarrow 16$, respectively. The selected paths are objective-dependent; we do not compare them with a matched-budget static ranking. The three arms also isolate two choices. Comparing MSE→MSE with MSE→CE keeps candidate training fixed while changing the repair objective; comparing MSE→CE with CE→CE holds the repair objective fixed while changing diagnosis. For Llama-3-8B, these choices lead to different WikiText-2/C4 trade-offs: CE repair yields a larger WikiText-2 gain than MSE repair, whereas MSE→MSE has the lowest C4 PPL among the three arms. The objective pair should therefore be read as a design choice, not as a universal ordering of training losses.

4.3 Global Refinement and Selection Limitations

Full-model versus selective CE refinement. We compare full-model CE continuation (updating all native trainable parameters) with selective CE→CE refinement from identical SliderQuant starting checkpoints. For Llama-3-8B, the baseline WikiText-2/C4 PPL is 7.0295/10.4507. Full-model CE obtains 6.6580/10.8974, whereas SparseDiag obtains 6.7461/10.3482: the former has lower WikiText-2 PPL, while only the latter improves C4. For Qwen3-8B, the baseline is 10.4845/14.1249; full-model CE reaches 7.5657/12.6237, compared with 8.2824/12.8742 under selective refinement. Selective refinement is therefore not uniformly superior. The budgets and diagnostic overhead differ, so this comparison concerns accuracy rather than matched-compute speed.

Longer full-model CE training. Extending full-model CE refinement of Initial State W4A16 models from 100 to 860 updates causes severe full-test degradation under the tested protocol: Llama-3-8B changes from 6.7821/11.0152 to 85.4529/114.9063 and Qwen3-8B from 7.7516/12.8049 to 70.5106/97.5912 (WikiText-2/C4). The longer run was designed to match a sequential training budget and does not isolate the number of updates from learning-rate schedule effects. It nevertheless shows that degradation under this protocol does not require sequential layer updates.

Failure of small-sample acceptance. We further refine Initial State checkpoints that have already undergone 100 updates of full-model CE optimization. Although the ranking and confirmation splits accept three additional CE→CE repairs, full-test performance deteriorates: Llama-3-8B changes from 6.7821/11.0152 to 6.8973/11.1867 along layers $11 \rightarrow 15 \rightarrow 28$, and Qwen3-8B changes from 7.7516/12.8049 to 7.8076/12.8840 along $19 \rightarrow 23 \rightarrow 29$. This experiment has a different starting state from the successful SliderQuant continuations in Table 1. Positive gains on small held-out calibration splits thus do not guarantee full-test transfer.

Scope and computational cost. Each diagnosis round trains an independent candidate for every layer; sparse committed updates do not imply an inexpensive scan. We do not claim a matched-compute speedup, a validated router between full-model and selective CE, or a cross-domain improvement guarantee.

4.4 Additional Quantization and Calibration Analyses

More aggressive quantization. Under W2A16, Llama-2-13B MSE→CE reduces WikiText-2/C4 PPL from 7.7124/11.3899 to 7.0794/11.1636 after two rounds, after which no candidate satisfies the configured eligibility criterion. This is one W2A16 experiment rather than a bitwidth sweep.

Learning rate and calibration corpus. For Llama-3-8B Initial State W4A16, full-model CE training calibrated on WikiText-2 gives 6.7821/11.0152 (WikiText-2/C4) at the default learning rate and 6.6798/10.8649 at half that rate. With C4 calibration, the corresponding results are 7.4562/10.6786 and 7.3941/10.5476. Together with the objective-response plot in Figure 1, this indicates sensitivity to calibration data and optimization hyperparameters rather than a calibration-independent model-family preference.

5 Related Work

Post-training quantization has progressed from adaptive rounding and block reconstruction (Nagel et al., 2020; Li et al., 2021; Wei et al., 2022) to methods tailored to LLM weights and activations. GPTQ and AWQ develop efficient weight calibration (Frantar et al., 2023; Lin et al., 2024b), while ZeroQuant, SmoothQuant, RPTQ, and Outlier Suppression+ address activation ranges and low-bit deployment (Yao et al., 2022; Xiao et al., 2023; Yuan et al., 2023; Yao et al., 2023; Wei et al., 2023; Liu et al., 2024). Outlier-aware, mixed-precision, and codebook-based methods tackle particularly sensitive weights or extreme compression (Dettmers et al., 2022; 2024; Kim et al., 2024; Lee et al., 2024; Chee et al., 2023; Tseng et al., 2024; Egiazarian et al., 2024; van Baalen et al., 2024; Badri & Shaji, 2023). These methods primarily seek a better quantized representation.

Optimization-based PTQ adds learnable clipping, transformations, and error compensation (Shao et al., 2024; Ma et al., 2024; Ashkboos et al., 2024; Liu et al., 2025; Sun et al., 2025; Lin et al., 2024a; Chen et al., 2026; Park et al., 2026). Other work addresses activation structure and low-bit formats (Federici et al., 2026; Lee et al., 2026; Blumenberg et al., 2026; Egiazarian et al., 2026). Most closely related to our setting, CBQ and ACBQ account for cross-block dependencies, and SliderQuant uses sensitivity-aware reconstruction schedules (Ding et al., 2025; Wang et al., 2026a;b); quantization-aware training extends optimization further (Liu et al., 2023; Chen et al., 2025). We take the checkpoint produced by such a procedure as the starting point and ask whether another update to a particular layer remains useful. Diagnosis is based on the resulting full-model loss, not only on a prescribed reconstruction schedule or a local error score.

Low-rank adaptation provides economical trainable parameters (Hu et al., 2022; Dettmers et al., 2023), but SparseDiag does not attach a new adaptation module. It reuses the native LoRA/LWC parameters in our SliderQuant implementation or the LWC parameters in OmniQuant. Its layer selection is conditional on the current checkpoint and candidate-update rule; an accepted repair triggers a new diagnosis rather than reusing an initial sensitivity ranking.

6 Conclusion

We studied what remains worth optimizing after PTQ. Under a fixed MSE probe, the layers with positive held-out full-model gains change across quantized checkpoints and become concentrated in a non-contiguous subset. SparseDiag uses this observation to diagnose independent candidates, repair a selected layer, and reassess the new state. It improves strong W4A16 checkpoints on several Llama and Qwen models without changing pretrained dense weights. The result is not unconditional: calibration-based acceptance can fail on full-test data, and a compute advantage over static selection remains unestablished. This separation between diagnosing a promising update and committing it provides a concrete way to study the remaining optimization scope after PTQ.

Reproducibility statement

Sections 3–4 specify native-parameter candidate updates, MSE/CE losses, a 96/16/16 split of 128 calibration sequences (2,048 tokens each), 32 MSE updates per Figure 2 candidate, held-out NLL selection, and repair acceptance/stopping. Full-model objective comparisons use 100 updates (50 per quantization-rate stage); Initial State refinements use eight rounds, and SliderQuant/Single-layer refinements use up to three. Test sets are never used for layer selection or stopping.

AI use statement

Generative AI tools assisted with research brainstorming, hypothesis and methodology discussions, experimental planning, code drafting and debugging, literature discovery and summarization, and manuscript drafting, editing, and reference formatting. They were also used to discuss the interpretation and presentation of experimental findings. Experimental measurements were obtained from the authors' runs rather than generated by AI. The authors are responsible for checking AI-assisted material, verifying the reported results and references, and ensuring the accuracy and originality of the final submission.

REFERENCES

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L. Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. QuaRot: Outlier-free 4-bit inference in rotated LLMs. In *Advances in Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=dfqsW38v1X.

Hicham Badri and Appu Shaji. Half-quadratic quantization of large machine learning models. Technical report, 2023. URL https://dropbox.github.io/hqq_blog/.

Patrick Blumenberg, Thomas Graave, and Tim Fingscheidt. Improving block-wise LLM quantization by 4-bit block-wise optimal float (BOF4): Analysis and variations. In *International Conference on Learning Representations*, 2026.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. QuIP: 2-bit quantization of large language models with guarantees. In *Advances in Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=xrk9g5vcXR.

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. EfficientQAT: Efficient quantization-aware training for large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10081–10100, 2025. doi: 10.18653/v1/2025.acl-long.498. URL https://aclanthology.org/2025.acl-long.498/.

Mengzhao Chen, Yi Liu, Jiahao Wang, Yi Bin, Wenqi Shao, and Ping Luo. PrefixQuant: Eliminating outliers by prefixed tokens for large language models quantization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2026. doi: 10.1109/TPAMI.2026.3711802.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. LLM.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/abs/2208.07339.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/abs/2305.14314.

Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. SpQR: A sparse-quantized representation for near-lossless LLM weight compression. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2306.03078.

Xin Ding, Xiaoyu Liu, Zhijun Tu, Yun Zhang, Wei Li, Jie Hu, Hanting Chen, Yehui Tang, Zhiwei Xiong, Baoqun Yin, and Yunhe Wang. CBQ: Cross-block quantization for large language models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/15212bd2265c4a3ab0dbc1b1982c1b69-Abstract-Conference.html.

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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 12284–12303, 2024. URL https://proceedings.mlr.press/v235/egiazarian24a.html.

Vage Egiazarian, Roberto Castro, Denis Kuznedelev, Andrei Panferov, Eldar Kurtic, Shubhra Pandit, Alexandre Marques, Mark Kurtz, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Bridging the gap between promise and performance for microscaling FP4 quantization. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/b87bb4f6346d727b265088235e5bc389-Abstract-Conference.html.

Marco Federici, Riccardo Del Chiaro, Boris van Breugel, Paul Whatmough, and Markus Nagel. STaMP: Sequence transformation and mixed precision for low-precision activation quantization. In *International Conference on Learning Representations*, 2026.

URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/17ca4d387834b86126cff1a018bea82c-Abstract-Conference.html.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.17323.

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://arxiv.org/abs/2106.09685.

Sehoon Kim, Coleman Richard Charles Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, and Kurt Keutzer. SqueezeLLM: Dense-and-sparse quantization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 23901–23923, 2024. URL https://proceedings.mlr.press/v235/kim24f.html.

Changhun Lee, Jungyu Jin, Taesu Kim, Hyungjun Kim, and Eunhyeok Park. OWQ: Outlier-aware weight quantization for efficient fine-tuning and inference of large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 13355–13364, 2024. doi: 10.1609/aaai.v38i12.29237.

Jaeseong Lee, Seung-won Hwang, Aurick Qiao, Zhewei Yao, and Yuxiong He. TAGQuant: Token-aware clustering for group-wise quantization. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 5: Industry Track)*, pp. 253–262, 2026. doi: 10.18653/v1/2026.eacl-industry.18. URL https://aclanthology.org/2026.eacl-industry.18/.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: Pushing the limit of post-training quantization by block reconstruction. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=POWv6hDd9XH.

Haokun Lin, Haobo Xu, Yichen Wu, Jingzhi Cui, Yingtao Zhang, Linzhan Mou, Linqi Song, Zhenan Sun, and Ying Wei. DuQuant: Distributing outliers via dual transformation makes stronger quantized LLMs. In *Advances in Neural Information Processing Systems*, 2024a. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/9febda1c8344cc5f2d51713964864e93-Abstract-Conference.html.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. AWQ: Activation-aware weight quantization for LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, 2024b. URL https://arxiv.org/abs/2306.00978.

Jing Liu, Ruihao Gong, Xiuying Wei, Zhiwei Dong, Jianfei Cai, and Bohan Zhuang. QLLM: Accurate and efficient low-bitwidth quantization for large language models. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/437bc4ccafd3fc6d4289bd10940be42b-Abstract-Conference.html.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. LLM-QAT: Data-free quantization aware training for large language models, 2023. URL https://arxiv.org/abs/2305.17888.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. SpinQuant: LLM quantization with learned rotations. In *International Conference on Learning Representations*, 2025. URL https://arxiv.org/abs/2405.16406.

Yuexiao Ma, Huixia Li, Xiawu Zheng, Feng Ling, Xuefeng Xiao, Rui Wang, Shilei Wen, Fei Chao, and Rongrong Ji. AffineQuant: Affine transformation quantization for large language models, 2024. URL https://arxiv.org/abs/2403.12544.

Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 7197–7206, 2020. URL https://proceedings.mlr.press/v119/nagel20a.html.

Yeonsik Park, Hyeonseong Kim, and Seungkyu Choi. SERQ: Saliency-aware low-rank error reconstruction for LLM quantization. In *International Conference on Learning Representations*, 2026.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. OmniQuant: Omnidirectionally calibrated quantization for large language models. In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=8Wuvhh0LYW.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, Xin Jiang, Wulong Liu, and Jun Yao. FlatQuant: Flatness matters for LLM quantization. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57587–57613, 2025. URL https://proceedings.mlr.press/v267/sun25l.html.

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*, volume 235 of *Proceedings of Machine Learning Research*, pp. 48630–48656, 2024. URL https://proceedings.mlr.press/v235/tseng24a.html.

Mart van Baalen, Andrey Kuzmin, Markus Nagel, Peter Couperus, Cedric Bastoul, Eric Mahurin, Tijmen Blankevoort, and Paul Whatmough. GPTVQ: The blessing of dimensionality for LLM quantization, 2024. URL https://arxiv.org/abs/2402.15319.

Hailing Wang, Jianglin Lu, Yitian Zhang, Huimin Zeng, and Yun Fu. ACBQ: Adaptive cross-block quantization of large language models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 42578–42592, 2026a. doi: 10.18653/v1/2026.acl-long.1971. URL https://aclanthology.org/2026.acl-long.1971/.

Shigeng Wang, Chao Li, Yangyuxuan Kang, Jiawei Fan, Zhonghong Qu, and Anbang Yao. SliderQuant: Accurate post-training quantization for LLMs. In *International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=YNqZqw4fLT.

Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. QDrop: Randomly dropping quantization for extremely low-bit post-training quantization. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2203.05740.

Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. Outlier suppression+: Accurate quantization of large language models by equivalent and effective shifting and scaling. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1648–1665, 2023. doi: 10.18653/v1/2023.emnlp-main.102. URL https://aclanthology.org/2023.emnlp-main.102/.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. SmoothQuant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 38087–38099, 2023. URL https://arxiv.org/abs/2211.10438.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. ZeroQuant: Efficient and affordable post-training quantization for large-scale transformers. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27168–27183, 2022. URL https://arxiv.org/abs/2206.01861.

Zhewei Yao, Xiaoxia Wu, Cheng Li, Stephen Youn, and Yuxiong He. ZeroQuant-V2: Exploring post-training quantization in LLMs from comprehensive study to low rank compensation, 2023. URL https://arxiv.org/abs/2303.08302.

Zhihang Yuan, Lin Niu, Jiawei Liu, Wenyu Liu, Xinggang Wang, Yuzhang Shang, Guangyu Sun, Qiang Wu, Jiaxiang Wu, and Bingzhe Wu. RPTQ: Reorder-based post-training quantization for large language models, 2023. URL https://arxiv.org/abs/2304.01089.