THE DEVIL IS IN THE RECONSTRUCTION LOSS SCALE: RETHINKING OPTIMIZATION IN LLM QUANTIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Post-training quantization (PTQ) methods typically quantize a pre-trained large language model (LLM) sequentially using a small set of calibration samples, achieving high data and computational efficiency. Specifically, the model is partitioned into a series of units, such as transformer blocks, with one unit quantized at each stage. State-of-the-art PTQ methods are predominantly learning-based, optimizing auxiliary quantization parameters (e.g., scaling factors, rotation matrices, clipping thresholds, and adapters) via gradient descent to minimize a reconstruction loss, thereby mitigating the impact of outliers and reducing quantization errors. In learning-based PTQ, a common practice is to use mean squared error (MSE) as the reconstruction loss function, yet its induced optimization behavior remains largely unexplored. In this work, we take a holistic view of sequential quantization and systematically investigate how optimization evolves from the first quantization stage to the last, aiming for a deep understanding of optimization in learning-based PTQ schemes. Through extensive empirical studies spanning representative learning-based PTQ methods, model families, model scales, architectures, and quantization settings, we consistently uncover *Optimization Imbalance*: reconstruction loss magnitudes vary dramatically across stages, accompanied by highly uneven gradient magnitudes and parameter updates under MSE. We term the cross-stage range of loss magnitudes the *reconstruction loss scale*. Our analysis reveals that MSE translates the unexpectedly large reconstruction loss scale into highly uneven gradient magnitudes, which in turn lead to uneven optimization strength across quantization stages. This finding suggests a general principle for improving learning-based PTQ: optimization strength across stages should be decoupled from the reconstruction loss scale. Theoretically, we show that root mean squared error (RMSE) naturally realizes this principle through implicit gradient normalization, thereby enabling balanced optimization across quantization stages. Moreover, RMSE can be naturally defined over different feature groupings, yielding four variants at the sample, channel, token, and element levels. As a drop-in replacement for MSE, RMSE consistently improves quantization performance across language modeling, commonsense reasoning, mathematical reasoning, and coding tasks, with all four variants outperforming MSE and the gains becoming increasingly pronounced under more aggressive quantization.

1 INTRODUCTION

Large language models (LLMs) (Vaswani et al., 2017; Brown et al., 2020; Achiam et al., 2023; Anil et al., 2023; Touvron et al., 2023; Grattafiori et al., 2024; Liu et al., 2024; Guo et al., 2025; Qwen Team, 2026) have achieved remarkable success across language understanding, generation, and reasoning tasks, yet their growing memory and computational demands pose a major obstacle to practical deployment. Post-training quantization (PTQ) offers an efficient solution by converting pre-trained high-precision models into low-bit replacements using a small calibration set, without the costly full-parameter retraining required by quantization-aware training (Zafrir et al., 2019; Ma et al., 2024; Xu et al., 2024). To efficiently quantize LLMs, PTQ methods generally partition the model into a sequence of units (e.g., transformer blocks) and quantize them in a stage-wise manner, processing one unit at each stage while keeping previously quantized units fixed. We refer to this popular stage-wise PTQ paradigm as *sequential quantization*.

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.

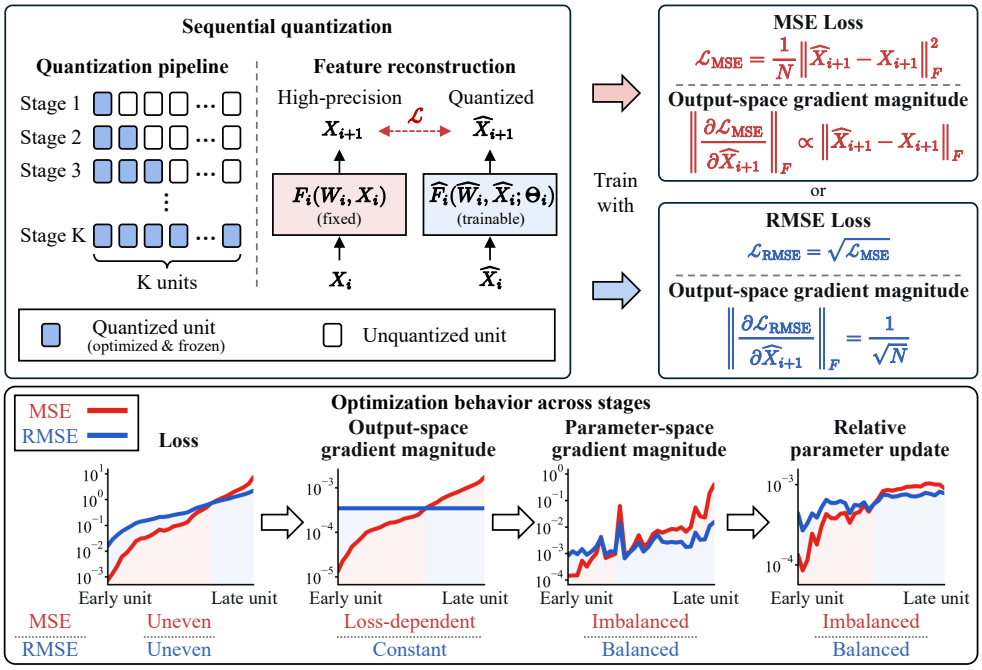

Figure 1: **Illustration of Optimization Imbalance in learning-based PTQ and its mitigation via RMSE.** Learning-based PTQ methods predominantly use MSE as the reconstruction loss function for each stage in sequential quantization. By tracking optimization dynamics from the first stage to the last, we observe that reconstruction loss magnitudes vary dramatically across quantization stages. Our analysis reveals that MSE translates these differences into highly uneven gradients and parameter updates, giving rise to Optimization Imbalance. RMSE breaks this loss-dependent coupling through implicit gradient normalization, enabling more balanced optimization across quantization stages. Statistics are collected using OmniQuant on Qwen3.5-9B under W4A4 quantization.

Under sequential quantization, existing PTQ methods differ primarily in how they perform feature (i.e., output) reconstruction to quantize each unit. Many methods employ carefully designed analytical or statistics-driven schemes for effective reconstruction, including second-order error compensation, activation-aware scaling, equivalent transformations, and fixed rotations. Representative methods include GPTQ (Frantar et al., 2023), AWQ (Lin et al., 2024b), SmoothQuant (Xiao et al., 2023), QuaRot (Ashkboos et al., 2024), and others (Li et al., 2025; Sanjeet et al., 2026; Chen et al., 2026; Li et al., 2026). By contrast, learning-based PTQ brings gradient descent into the feature reconstruction process without full-parameter retraining. At each quantization stage, the pre-trained weights remain frozen, while only a small set of auxiliary quantization parameters is optimized through feature reconstruction. This learning-based concept is adopted by most top-performing PTQ methods, which primarily differ in their learnable parameterizations and reconstruction schemes. OmniQuant (Shao et al., 2024) learns scaling factors and clipping thresholds, DuQuant (Lin et al., 2024a) optimizes orthogonal rotations and permutations, FlatQuant (Sun et al., 2025) learns affine transformations, and SliderQuant (Wang et al., 2026) combines scaling and low-rank adaptation with sliding-window reconstruction. Other recent methods further explore related learnable design choices (Liu et al., 2025; Ding et al., 2025; van Breugel et al., 2026; Liang et al., 2026). Despite differences in their formulations, we observe that nearly all learning-based PTQ methods use mean squared error (MSE) as the reconstruction loss function, with the exception of LRQuant (Zhao et al., 2024), which augments MSE with a cosine-similarity-based term in its proposed learnable smoothing strategy to address outliers. Although learning-based PTQ has advanced considerably, the optimization behavior induced by MSE throughout sequential quantization remains largely unexplored. Additionally, sequential quantization implicitly assumes that every unit receives sufficient optimization, yet this assumption has not been systematically examined.

Motivated by these two largely overlooked perspectives, we take a holistic view of the sequential learning-based PTQ process, tracking how reconstruction losses, gradients, and parameter updates evolve from the first quantization stage to the last. Our empirical studies cover representative learning-based PTQ methods (OmniQuant, DuQuant, FlatQuant, and SliderQuant), model families (LLaMA2, LLaMA3.1, and Qwen3.5), model scales (7B, 8B, 9B, 27B, and 35B-A3B), architec-

tures (dense and MoE), and quantization settings (W4A16, W3A16, W2A16, W2A16G128, and W4A4). Across this broad testing spectrum, we consistently uncover *Optimization Imbalance*, as illustrated in Fig. 1: reconstruction loss magnitudes span an unexpectedly large cross-stage range, which we term the *reconstruction loss scale*, with the largest often exceeding the smallest by more than $10{,}000\times$. Under MSE, this broad loss scale is accompanied by highly uneven gradients and parameter updates across quantization stages. This imbalance is consequential in sequential quantization because quantization errors propagate and accumulate across stages, meaning that insufficient optimization at one stage can negatively affect the quantization of all subsequent units.

Our analysis identifies MSE as the fundamental source of Optimization Imbalance, revealing that it inherently couples optimization strength with reconstruction loss magnitude. Specifically, MSE translates the unexpectedly large reconstruction loss scale into large differences in output-space gradient magnitude, which are further reflected in parameter-space gradients and ultimately manifest as highly uneven parameter updates across quantization stages. This loss-dependent coupling directly explains the observed Optimization Imbalance and suggests a general principle for improving learning-based PTQ: optimization strength across quantization stages should be decoupled from the reconstruction loss scale. This principle naturally leads to root mean squared error (RMSE). As a square-root transformation of MSE, RMSE preserves the gradient direction and relative magnitudes among gradient elements while implicitly normalizing the overall output-space gradient magnitude to a constant. RMSE therefore breaks the loss-dependent coupling of MSE and enables more balanced optimization across quantization stages. RMSE can be naturally defined over different feature groupings, yielding a unified family of four variants operating at the sample, channel, token, and element levels. These variants preserve the same cross-stage balancing property while applying gradient normalization within different feature groups at each stage.

Extensive experiments demonstrate the generality and practical benefits of replacing MSE with RMSE in learning-based PTQ: (1) RMSE consistently improves quantization performance in language modeling, commonsense reasoning, mathematical reasoning, and coding tasks across representative learning-based PTQ methods, model families, model scales, architectures, and quantization settings, with all four RMSE variants outperforming MSE; (2) the gains generally become more pronounced under aggressive quantization, with absolute average accuracy improvements ranging from 5.44% to 10.83% under W2A16 and from 0.75% to 8.05% under W4A4; and (3) as a drop-in replacement for MSE, RMSE requires no additional trainable parameters or hyperparameters while introducing negligible runtime and memory overhead. More importantly, our analysis and experimental results establish Optimization Imbalance as a previously overlooked phenomenon arising from the loss-dependent gradient scaling of MSE in sequential quantization and highlight optimization behavior as a fundamental design consideration for learning-based PTQ.

2 Understanding Optimization Imbalance in Learning-based PTQ

2.1 Learning-based PTQ Preliminaries

We formulate learning-based PTQ under sequential quantization at the unit level, where a unit can be a layer, a block, or a group of consecutive blocks. Without loss of generality, we present the formulation for the more general weight-activation quantization setting, with weight-only quantization as the special case where activations remain in high precision. At the i-th stage, the i-th quantized unit is optimized to reconstruct the output of its high-precision counterpart. For the i-th unit, let F_i denote its high-precision mapping with pre-trained weights W_i, and let $\widehat{F}_i$ denote its quantized counterpart. The quantized unit operates with effective quantized weights $\widehat{W}_i$ and quantized activations, governed by trainable auxiliary quantization parameters Θ_i, such as scaling factors and rotation matrices. During reconstruction, W_i remains frozen and only Θ_i is optimized. Given the high-precision input X_i and quantized input $\widehat{X}_i$, the corresponding unit outputs are

$$X_{i+1} = F_i(W_i, X_i), \qquad \widehat{X}_{i+1} = \widehat{F}_i(\widehat{W}_i, \widehat{X}_i; \Theta_i). \tag{1}$$

Most existing learning-based PTQ methods use mean squared error (MSE) as the reconstruction loss function. Specifically, the auxiliary parameters Θ_i are optimized by minimizing

$$\mathcal{L}_{\mathrm{MSE}} = \frac{1}{N} \left\| \widehat{X}_{i+1} - X_{i+1} \right\|_F^2, \tag{2}$$

where N denotes the total number of elements in X_{i+1}.

2.2 OBSERVATION: OPTIMIZATION IMBALANCE IN SEQUENTIAL QUANTIZATION

While sequential quantization processes one unit at a time, it implicitly assumes that each unit receives sufficient optimization. But is this really the case? To answer this question, we systematically investigate optimization dynamics across units throughout the sequential quantization process, covering representative learning-based PTQ methods, model families, model scales, architectures, and quantization settings. A consistent pattern emerges across this broad spectrum: reconstruction loss magnitudes span a broad cross-stage range, which we term the *reconstruction loss scale*, even though all units are optimized using the same MSE loss function. In many cases, the largest loss magnitude exceeds the smallest by more than $10{,}000\times$. Fig. 1 presents a representative example, while comprehensive statistics are provided in Sec. 4.2 and Appendix D. This broad loss scale arises naturally during sequential quantization. As quantization proceeds through the network, errors introduced by previously quantized units accumulate, making subsequent units increasingly difficult to reconstruct. Meanwhile, reconstruction losses are computed directly over unit outputs, whose magnitudes can vary substantially across units. Together, these factors give rise to the broad reconstruction loss scale observed across stages. This broad loss scale raises another question: *do units with substantially different loss magnitudes receive comparable optimization strength?* We quantify this using the relative parameter update

$$\frac{\|\Delta\Theta_i\|_F}{\|\Theta_i\|_F}, \tag{3}$$

computed over all trainable parameters at each optimization step and averaged across all steps in the quantization stage of the i-th unit. The relative parameter update measures how strongly a unit is modified during reconstruction and therefore serves as a practical proxy for the optimization strength received by that unit. We observe that the relative parameter update is consistently much larger for units with larger loss magnitudes, whereas units with smaller loss magnitudes receive only minimal updates. The close correspondence between loss magnitudes and relative parameter updates reveals a systematic imbalance in optimization across units, which we term *Optimization Imbalance*. This imbalance is especially critical in sequential quantization, where each quantized unit remains fixed, so the errors left by an under-optimized unit propagate and accumulate through all subsequent units. Moreover, since loss magnitudes generally increase from early to late stages, early units tend to receive the weakest optimization, despite affecting the most subsequent units.

2.3 WHY MSE LEADS TO OPTIMIZATION IMBALANCE

What connects the reconstruction loss scale to Optimization Imbalance? Since gradients serve as the carriers of optimization signals, understanding this relationship requires analyzing how the reconstruction loss function influences the gradients received by each unit. A natural starting point is the parameter-space gradient, which directly determines parameter updates during reconstruction. Using the chain rule, we get

$$\frac{\partial \mathcal{L}_{\mathrm{MSE}}}{\partial \Theta_i} = \frac{\partial \mathcal{L}_{\mathrm{MSE}}}{\partial \widehat{X}_{i+1}} \cdot \frac{\partial \widehat{X}_{i+1}}{\partial \Theta_i}. \tag{4}$$

The parameter-space gradient is jointly determined by the loss function and the network Jacobian. Since our goal is to understand the optimization behavior induced by the loss function itself, independent of any particular network parameterization, we focus on $\partial \mathcal{L}_{\mathrm{MSE}}/\partial \widehat{X}_{i+1}$, which characterizes the output-space optimization signal generated directly by the loss function. This output-space perspective enables a cleaner theoretical analysis, while Sec. 4.2 examines how the resulting gradient imbalance is reflected in parameter-space gradients and parameter updates. For the MSE loss defined in Eq. 2, the output-space gradient magnitude is

$$\left\| \frac{\partial \mathcal{L}_{\mathrm{MSE}}}{\partial \widehat{X}_{i+1}} \right\|_F = \frac{2}{N} \left\| \widehat{X}_{i+1} - X_{i+1} \right\|_F = 2\sqrt{\frac{\mathcal{L}_{\mathrm{MSE}}}{N}}. \tag{5}$$

This derivation reveals a fundamental property of MSE: the output-space gradient magnitude is proportional to the square root of the loss magnitude. Since the reconstruction loss scale can span more than $10{,}000\times$, this square-root scaling produces a range of output-space gradient magnitudes exceeding $100\times$. Thus, MSE inherently couples optimization strength to loss magnitude, directly accounting for Optimization Imbalance. Although the ultimate optimization dynamics are further influenced by network parameterization, this broad range of output-space gradient magnitudes provides a natural explanation for the highly uneven optimization strength observed in practice.

3 MITIGATING OPTIMIZATION IMBALANCE VIA RMSE

3.1 RMSE FOR BALANCED OPTIMIZATION

Our analysis suggests a general principle for improving learning-based PTQ: an ideal loss function should decouple optimization strength across stages from the reconstruction loss scale. Root mean squared error (RMSE), a square-root transformation of MSE, naturally realizes this principle:

$$\mathcal{L}_{\mathrm{RMSE}} = \sqrt{\frac{1}{N}\left\|\widehat{X}_{i+1} - X_{i+1}\right\|_F^2}. \tag{6}$$

Specifically, its output-space gradient magnitude is given by

$$\left\|\frac{\partial \mathcal{L}_{\mathrm{RMSE}}}{\partial \widehat{X}_{i+1}}\right\|_F = \left\|\frac{1}{\sqrt{N}}\frac{\widehat{X}_{i+1} - X_{i+1}}{\left\|\widehat{X}_{i+1} - X_{i+1}\right\|_F}\right\|_F = \frac{1}{\sqrt{N}}, \tag{7}$$

which is independent of the reconstruction loss magnitude. Compared with MSE, RMSE preserves the same gradient direction and relative magnitudes among gradient elements, while normalizing the overall gradient magnitude to a constant. In other words, although the RMSE loss itself may still span a broad range of magnitudes, this variation does not translate into imbalanced gradient magnitudes, thereby enabling balanced optimization despite the broad reconstruction loss scale. Importantly, this implicit gradient normalization emerges naturally from the loss function itself, requiring neither additional hyperparameters nor modifications to the learning-based PTQ pipeline.

3.2 A UNIFIED FAMILY OF RMSE LOSS FUNCTIONS

The analysis above establishes the gradient normalization property of RMSE, enabling balanced optimization across stages. The specific form of RMSE depends on which reconstruction errors are aggregated under each square root. Formally, let $\{G_1, G_2, \ldots, G_M\}$ denote a partition of all N feature elements into M equal-sized groups. The generalized RMSE loss function is defined as

$$\mathcal{L}_{\mathrm{RMSE}}^{(G)} = \frac{1}{M}\sum_{m=1}^{M}\sqrt{\frac{1}{|G_m|}\sum_{j\in G_m}\left(\widehat{X}_{i+1,j} - X_{i+1,j}\right)^2}. \tag{8}$$

For any such equal-sized grouping, the overall output-space gradient magnitude remains $1/\sqrt{N}$, independent of both the reconstruction loss magnitude and the grouping strategy. Different groupings therefore preserve the same cross-stage balancing property while applying gradient normalization within different feature groups at each stage. For a reconstruction error tensor with sample, token, and channel dimensions, its elements can be naturally grouped at the sample, channel, token, or element level. These grouping strategies yield four RMSE variants. Specifically, **RMSE-Sample** treats each calibration sample as an individual group. **RMSE-Channel** treats each channel within each sample as an individual group. **RMSE-Token** treats each token within each sample as an individual group. **RMSE-Element** treats each feature element as an individual group and therefore reduces to the L1 loss. Their concrete formulations are provided in Appendix A.2. Rather than independent heuristics, these variants represent different instantiations of the same gradient normalization principle. Consistent with their shared principle, all four variants outperform MSE across the evaluations in Sec. 4.3 and Appendix C, while their relative performance differs across model families. We further observe that reconstruction loss magnitudes exhibit different degrees of within-stage variation under different feature groupings (Fig. 9), indicating that feature grouping constitutes a meaningful design dimension.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models. For evaluation, we select five representative LLMs spanning different model families, model scales, and architectures (dense and MoE): LLaMA2-7B and LLaMA3.1-8B from the LLaMA family (Touvron et al., 2023; Grattafiori et al., 2024), and Qwen3.5-9B, Qwen3.5-27B,

Table 1: Generalization of RMSE under OmniQuant across diverse LLMs and quantization settings. Acc denotes the average accuracy over PIQA, ARC-Easy, ARC-Challenge, Winogrande, HellaSwag, and BoolQ.

Model	Loss	Original		W4A16		W3A16		W2A16		W2A16G128		W4A4	
		PPL↓	Acc↑	PPL↓	Acc↑	PPL↓	Acc↑	PPL↓	Acc↑	PPL↓	Acc↑	PPL↓	Acc↑
LLaMA2-7B	MSE	5.47	70.45	5.75	68.33	6.62	64.05	36.92	43.98	11.05	51.66	17.42	49.09
	RMSE			5.70	69.07	6.38	66.78	16.06	49.97	9.10	57.29	11.48	56.23
LLaMA3.1-8B	MSE	6.24	75.04	7.19	72.59	12.38	60.55	1021.33	37.00	124.49	38.07	110.88	40.09
	RMSE			6.92	72.89	9.46	67.17	140.63	42.44	20.21	48.60	44.29	43.51
Qwen3.5-9B	MSE	8.65	75.10	9.28	74.23	11.56	71.10	99.05	45.44	23.61	53.54	69.74	45.43
	RMSE			9.14	74.35	11.21	72.79	51.56	54.78	17.98	60.03	51.88	49.37
Qwen3.5-27B	MSE	6.90	77.10	7.17	77.71	8.42	77.12	37.51	60.45	11.73	70.88	29.45	59.46
	RMSE			7.09	78.37	8.26	78.13	22.20	70.96	11.29	73.09	16.13	67.51
Qwen3.5-35B-A3B	MSE	6.59	78.34	7.22	77.61	9.87	73.05	73.79	46.76	24.66	56.47	12.42	65.63
	RMSE			7.06	77.98	8.83	74.50	21.85	57.59	13.90	66.76	10.82	67.48

and Qwen3.5-35B-A3B from the Qwen3.5 family (Qwen Team, 2026). Among them, LLaMA2-7B and Qwen3.5-9B serve as our primary evaluation models, representing a widely adopted PTQ benchmark and a modern hybrid linear-attention architecture, respectively.

Quantization Methods. We evaluate four representative learning-based PTQ methods spanning diverse learnable quantization designs: scaling factors and clipping thresholds in OmniQuant (Shao et al., 2024), orthogonal rotations and permutations in DuQuant (Lin et al., 2024a), affine transformations in FlatQuant (Sun et al., 2025), and scaling, low-rank adaptation, and sliding-window reconstruction in SliderQuant (Wang et al., 2026). Despite their substantially different designs, all four methods rely on MSE-based feature reconstruction. This diversity allows us to examine whether Optimization Imbalance arises from the shared loss function rather than any particular quantization design, and whether RMSE generalizes across different learning-based PTQ methods. Following their official implementations, each unit corresponds to a transformer block, except for SliderQuant, which reconstructs a sliding window of consecutive blocks.

Quantization Settings and Implementation Details. We evaluate five quantization settings spanning weight-only and weight-activation quantization: W4A16, W3A16, W2A16, W2A16G128 (group size 128), and W4A4. Unless otherwise specified, we follow the official implementations and hyperparameter settings of each method and add support for model families not covered by the official code. For fair comparison, all results are obtained from our own reproductions, enabling a unified analysis of optimization behavior across different methods. We evaluate all methods using a unified LM-Evaluation-Harness (Gao et al., 2024) pipeline, reporting WikiText2 (Merity et al., 2017) perplexity (PPL) and zero-shot accuracy on PIQA (Bisk et al., 2020), ARC-Easy and ARC-Challenge (Clark et al., 2018), Winogrande (Sakaguchi et al., 2021), HellaSwag (Zellers et al., 2019), and BoolQ (Clark et al., 2019). Unless otherwise specified, we use RMSE-Token for LLaMA models and RMSE-Channel for Qwen models based on their overall empirical performance. *More implementation details are provided in Appendix A.*

4.2 Main Results

Generalization Across Models and Quantization Settings. If Optimization Imbalance is indeed detrimental to learning-based PTQ, mitigating it should consistently improve quantization performance across diverse settings. We first test this hypothesis across model families, model scales, architectures, and quantization settings. We adopt OmniQuant as a representative framework because it covers both weight-only and weight-activation quantization, enabling controlled evaluation across diverse quantization regimes. As shown in Table 1, replacing MSE with RMSE reduces perplexity and improves downstream accuracy in every evaluated configuration. While RMSE yields modest improvements under relatively mild settings such as W4A16, substantially larger gains emerge under more aggressive settings such as W2A16, W2A16G128, and W4A4. For example, on Qwen3.5-27B under W2A16 quantization, RMSE reduces perplexity from 37.51 to 22.20 while improving downstream accuracy from 60.45% to 70.96%. This trend suggests that Optimization Imbalance becomes increasingly detrimental as reconstruction becomes more difficult, making balanced optimization particularly important under aggressive quantization.

To relate these performance gains to the underlying optimization behavior, we examine the corresponding statistics in Fig. 2. Under MSE, the broad loss scale is translated into highly uneven

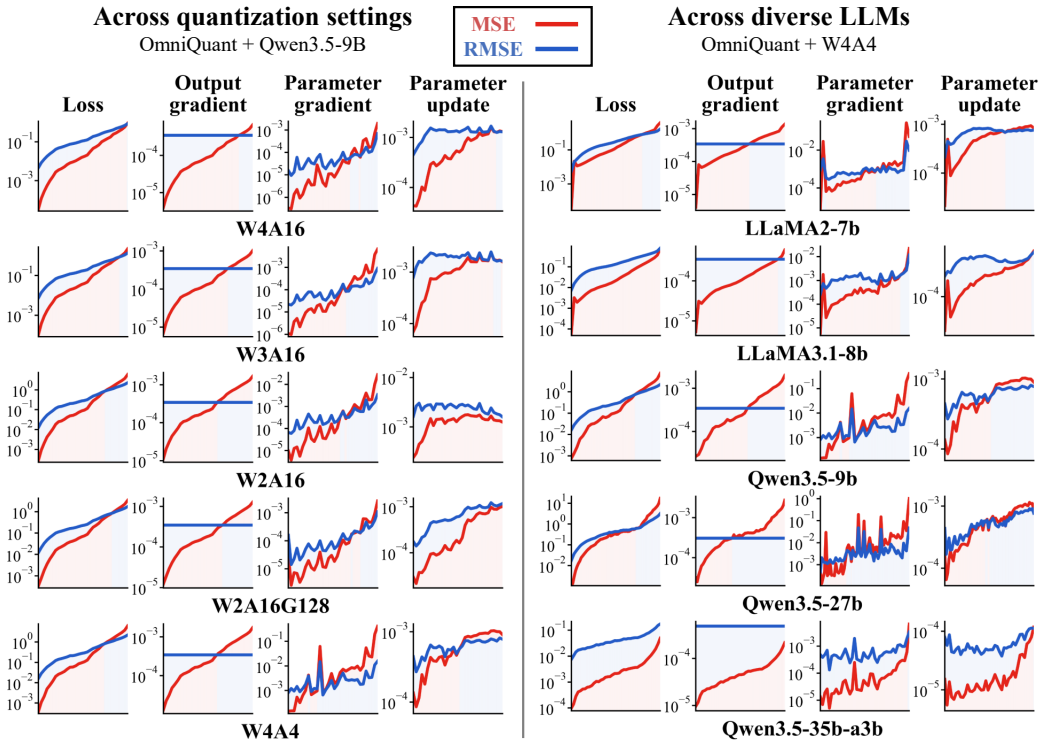

Figure 2: Optimization statistics under MSE and RMSE across quantization settings and diverse LLMs. The horizontal axis represents the quantization stages from early to late. For each stage, loss, output-space gradient, parameter-space gradient, and relative parameter update are averaged over all optimization steps. Parameter-space gradients and parameter updates are computed over trainable parameters only.

output-space gradients and further reflected in parameter-space gradients and relative parameter updates. In contrast, RMSE maintains constant output-space gradient magnitudes despite the broad loss scale, resulting in considerably more balanced gradients and updates across stages. Together, the performance gains and optimization statistics validate our analysis of Optimization Imbalance and establish balanced optimization as an effective design principle for learning-based PTQ.

Generalization Across PTQ Methods. We further test whether the preceding findings generalize beyond OmniQuant. Specifically, we extend our analysis to three additional representative learning-based PTQ methods, namely SliderQuant, DuQuant, and FlatQuant, which employ substantially different quantization designs. As shown in Table 2, RMSE yields consistent performance gains across all four methods despite their distinct designs. The corresponding statistics in Fig. 3 reveal the same optimization pattern: under MSE, the broad reconstruction loss scale is accompanied by uneven gradients and parameter updates, whereas RMSE produces considerably more balanced optimization. These results demonstrate that Optimization Imbalance is broadly shared across representative learning-based PTQ methods. Its consistent presence across different quantization designs further supports the loss-dependent gradient scaling of MSE as its source.

4.3 Ablation Studies

Comparing RMSE Variants. As introduced in Sec. 3.2, all four RMSE variants preserve the same cross-stage balancing property while applying gradient normalization to different feature groups within each unit. As shown in Table 3, all four variants consistently outperform MSE across different models and quantization settings, confirming the effectiveness of this principle across feature groupings. At the same time, their relative performance varies across model families: LLaMA-family models generally favor RMSE-Token, whereas Qwen-family models generally favor RMSE-Channel. This model-dependent preference further shows that feature grouping constitutes a meaningful design dimension for balanced optimization. We therefore adopt RMSE-Token for LLaMA models and RMSE-Channel for Qwen models throughout the main experiments. For brevity, we

Table 2: Generalization of RMSE across representative learning-based PTQ methods under W4A4 quantization on LLaMA2-7B and Qwen3.5-9B. Acc denotes the average accuracy over PIQA, ARC-Easy, ARC-Challenge, Winogrande, HellaSwag, and BoolQ.

Model	Method	Loss	PPL↓	Acc↑
LLaMA2-7B	Original	–	5.47	70.45
	OmniQuant	MSE	17.42	49.09
		RMSE	11.48	56.23
	SliderQuant	MSE	8.48	57.20
		RMSE	8.14	59.74
	DuQuant	MSE	6.07	66.91
		RMSE	6.01	67.69
	FlatQuant	MSE	5.82	67.72
		RMSE	5.78	68.76
Qwen3.5-9B	Original	–	8.65	75.10
	OmniQuant	MSE	69.74	45.43
		RMSE	51.88	49.37
	SliderQuant	MSE	21.41	60.43
		RMSE	20.64	61.92
	DuQuant	MSE	10.95	69.85
		RMSE	10.54	70.60
	FlatQuant	MSE	10.26	72.23
		RMSE	9.15	74.19

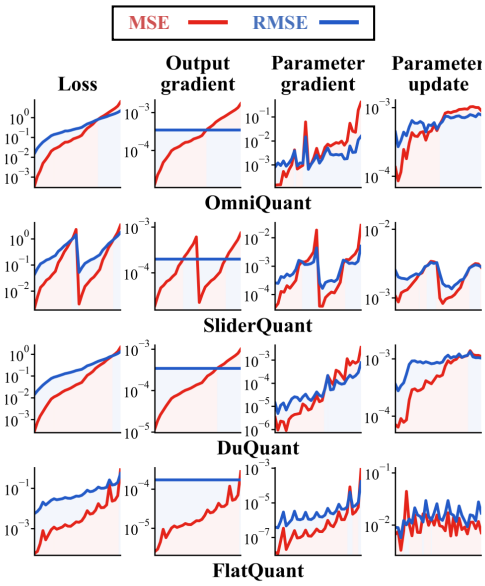

Figure 3: Optimization statistics under MSE and RMSE across representative learning-based PTQ methods on Qwen3.5-9B. The horizontal axis denotes the quantization stages from early to late.

Table 3: Comparison of RMSE variants across quantization settings under OmniQuant. Results are reported as average accuracy (%) on downstream benchmarks.

#Bits	LLaMA2-7B					Qwen3.5-9B				
	MSE	RMSE				MSE	RMSE			
		Sample	Channel	Token	Element		Sample	Channel	Token	Element
W4A16	68.33	68.88	**69.33**	69.07	68.64	74.23	74.97	74.35	**75.08**	74.70
W3A16	64.05	64.17	65.88	**66.78**	66.58	71.10	**73.17**	72.79	72.47	72.92
W2A16	43.98	45.09	46.01	**49.97**	46.76	45.44	47.61	**54.78**	47.63	52.76
W2A16G128	51.66	56.58	57.36	57.29	**57.54**	53.54	**61.08**	60.03	59.89	58.63
W4A4	49.09	51.73	53.40	**56.23**	54.35	45.43	47.85	**49.37**	47.38	48.79

Table 4: Generalization of RMSE to mathematical reasoning and coding tasks under OmniQuant on Qwen3.5-9B. All results are reported as accuracy (%).

#Bits	Loss	Mathematical Reasoning		Coding		Avg
		GSM8K	MATH-500	HumanEval	MBPP	
Original	–	96.44	97.20	85.37	70.37	87.35
W4A16	MSE	95.15	94.20	75.61	57.14	80.53
	RMSE	95.98	94.80	75.00	62.96	**82.19**
W3A16	MSE	78.70	51.20	32.93	24.34	46.79
	RMSE	81.65	54.60	35.98	25.40	**49.41**
W2A16G128	MSE	1.67	2.20	0.00	0.00	0.97
	RMSE	7.35	7.00	0.00	0.26	**3.65**

present a subset of results here, *while comprehensive comparisons across additional models, quantization settings, and learning-based PTQ methods are provided in Appendix C.*

Generalization to Reasoning Tasks. To examine whether balanced optimization benefits complex reasoning, we evaluate the quantized models obtained under OmniQuant, without task-specific training or adaptation, on mathematical reasoning (GSM8K (Cobbe et al., 2021) and MATH-500 (Hendrycks et al., 2021)) and coding (HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021)). As shown in Table 4, RMSE consistently improves overall performance across quantization settings. Under W2A16G128, however, RMSE alone cannot fully recover the severely degraded performance, highlighting the need for specialized quantization designs in such aggressive settings.

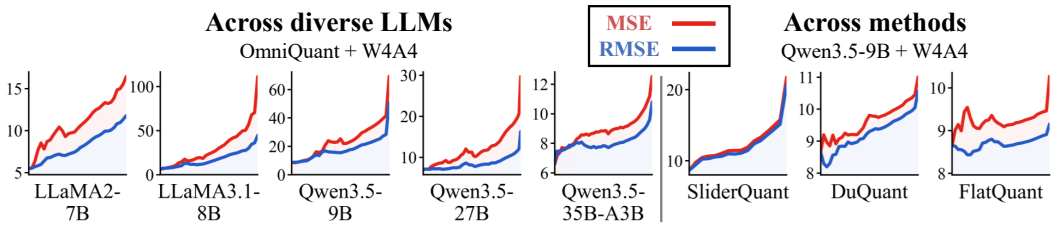

Figure 4: Perplexity evolution with quantization depth under MSE and RMSE. The horizontal axis represents the quantization stages from early to late. Left: OmniQuant with W4A4 quantization across diverse LLMs. Right: representative learning-based PTQ methods on Qwen3.5-9B under W4A4 quantization.

Table 5: Quantization process efficiency comparison between MSE and RMSE under OmniQuant. Runtime and peak GPU memory are measured over the entire sequential quantization process on a single NVIDIA L40 GPU with a batch size of 1.

Model	#Bits	Runtime (minutes)			Peak memory (GB)		
		MSE	RMSE	Overhead	MSE	RMSE	Overhead
LLaMA2-7B	W4A16	123.9	127.4	+2.82%	6.14	6.14	+0.00%
	W4A4	181.7	181.1	-0.33%	7.28	7.28	+0.00%
Qwen3.5-9B	W4A16	184.8	182.6	-1.19%	5.63	5.63	+0.00%
	W4A4	233.9	236.3	+1.02%	6.90	6.90	+0.00%

Perplexity Across Quantization Depth. To further connect optimization behavior with model performance, we record the perplexity of partially quantized models throughout sequential quantization. As shown in Fig. 4, RMSE consistently maintains lower perplexity than MSE, with the gap generally widening as quantization progresses. This trend highlights the increasing importance of balanced optimization as quantization errors accumulate.

Computational Overhead. As shown in Table 5, RMSE introduces negligible runtime and memory overhead as a drop-in replacement for MSE, enabling more balanced optimization while preserving computational efficiency.

5 Discussion

Existing learning-based PTQ research has primarily focused on developing advanced quantization designs. Our work suggests a complementary perspective: the choice of loss function itself fundamentally shapes optimization behavior across stages during sequential quantization. The consistent benefits of RMSE, particularly under aggressive quantization, indicate that Optimization Imbalance is not a minor artifact but a systematic phenomenon arising from the loss-dependent gradient scaling of MSE during sequential quantization. Its impact becomes increasingly pronounced as reconstruction becomes more difficult. This issue is particularly consequential in sequential quantization, where insufficient optimization at an early stage may influence all subsequent quantization stages. Furthermore, the distinct behavior of different feature grouping strategies indicates that feature grouping constitutes a meaningful design dimension, opening a broader space for balanced optimization beyond the variants studied here. Taken together, these findings suggest that optimization behavior should be treated as a fundamental design consideration in future learning-based PTQ research, alongside quantization design.

6 Conclusion

We identify *Optimization Imbalance* as a previously overlooked phenomenon arising from MSE-based reconstruction during sequential quantization in learning-based PTQ, where optimization strength across quantization stages is coupled with the reconstruction loss scale. To address this, we propose a family of RMSE loss functions that naturally decouple these two quantities and enable balanced optimization across quantization stages. These loss functions serve as drop-in replacements for MSE. Extensive experiments across learning-based PTQ methods, model families, model scales, architectures, and quantization settings demonstrate consistent improvements in quantization performance across language modeling, commonsense reasoning, mathematical reasoning, and coding tasks, with the gains becoming increasingly pronounced under more aggressive quantization.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, all experiments are conducted using publicly available models, datasets, and open-source PTQ frameworks, including OmniQuant, FlatQuant, DuQuant, and SliderQuant. Implementation details regarding our four RMSE variants based on different feature grouping strategies, experimental settings, hyperparameters, and evaluation protocols are provided in the main paper and appendix. We will release our implementation and experimental code upon publication.

AI USE STATEMENT

We used generative AI tools solely to assist with proofreading the manuscript. All AI-assisted content was reviewed and verified by the authors. The authors take full responsibility for the content of this work.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

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 *NeurIPS*, 2024.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning about physical commonsense in natural language. In *AAAI*, 2020.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *NeurIPS*, 2020.

Jiale Chen, Vage Egiazarian, Roberto L Castro, Torsten Hoefler, and Dan Alistarh. Wush: Near-optimal adaptive transforms for llm quantization. In *ICML*, 2026.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *NAACL*, 2019.

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. *arXiv preprint arXiv:1803.05457*, 2018.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Xin Ding, Xiaoyu Liu, Zhijun Tu, Yun Zhang, Wei Li, Jie Hu, Hanting Chen, Yehui Tang, Zhiwei Xiong, Baoqun Yin, et al. Cbq: Cross-block quantization for large language models. In *ICLR*, 2025.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. In *ICLR*, 2023.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language model evaluation harness, 2024. URL https://zenodo.org/records/12608602.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *NeurIPS*, 2021.

Yuhang Li, Ruokai Yin, Donghyun Lee, Shiting Xiao, and Priyadarshini Panda. GPTAQ: efficient finetuning-free quantization for asymmetric calibration. In *ICML*, 2025.

Zhikai Li, Zhen Dong, Xuewen Liu, Jing Zhang, and Qingyi Gu. Osaq: Outlier self-absorption for accurate low-bit llm quantization. In *ICML*, 2026.

Yesheng Liang, Haisheng Chen, Song Han, and Zhijian Liu. Paroquant: Pairwise rotation quantization for efficient reasoning llm inference. In *ICLR*, 2026.

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 *NeurIPS*, 2024a.

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

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, et al. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. In *NeurIPS*, 2023.

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 *ICLR*, 2025.

Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in 1.58 bits. *arXiv preprint arXiv:2402.17764*, 2024.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *ICLR*, 2017.

Qwen Team. Qwen3.5: Towards native multimodal agents, 2026. URL https://qwen.ai/blog?id=qwen3.5.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 2021.

Sai Sanjeet, Ian Colbert, Pablo Monteagudo-Lago, Giuseppe Franco, Yaman Umuroglu, and Nicholas J Fraser. Pushing the limits of block rotations in post-training quantization. In *ICML*, 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 *ICLR*, 2024.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, et al. Flatquant: Flatness matters for llm quantization. In *ICML*, 2025.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Boris van Breugel, Yelysei Bondarenko, Paul Whatmough, and Markus Nagel. Fptquant: Function-preserving transforms for llm quantization. In *ICML*, 2026.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, 2017.

Shigeng Wang, Chao Li, Yangyuxuan Kang, Jiawei Fan, Zhonghong Ou, and Anbang Yao. Sliderquant: Accurate post-training quantization for llms. In *ICLR*, 2026.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *ICML*, 2023.

Yuzhuang Xu, Xu Han, Zonghan Yang, Shuo Wang, Qingfu Zhu, Zhiyuan Liu, Weidong Liu, and Wanxiang Che. Onebit: Towards extremely low-bit large language models. In *NeurIPS*, 2024.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. Q8bert: Quantized 8bit bert. In *EMC2-NeurIPS*, 2019.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *ACL*, 2019.

Jiaqi Zhao, Miao Zhang, Chao Zeng, Ming Wang, Xuebo Liu, and Liqiang Nie. Lrquant: Learnable and robust post-training quantization for large language models. In *ACL*, 2024.

APPENDIX

A IMPLEMENTATION DETAILS

A.1 EXPERIMENTAL DETAILS

We follow the official implementations and hyperparameter settings of each PTQ method. For a fair comparison, RMSE is used as a drop-in replacement for the original MSE loss function, while all other components, including trainable parameters, optimizer, learning rate, reconstruction steps, and quantization configurations, are kept unchanged. When an official implementation does not directly support a model family, we implement the corresponding quantization pipeline following the original method design. All methods are calibrated using 128 samples from WikiText2, each with a sequence length of 2048, following the standard protocol used in prior PTQ work. For each MSE/RMSE comparison, we use exactly the same calibration samples to ensure that the observed differences come from the loss function rather than data variation.

For reasoning-intensive evaluation, we evaluate GSM8K and MATH-500 using LM-Evaluation-Harness, and HumanEval and MBPP using EvalPlus (Liu et al., 2023). We adopt the officially recommended sampling settings of Qwen3.5 for reasoning generation. All reasoning evaluations are conducted on the same quantized models without task-specific training or adaptation.

All quantization experiments are conducted on a single NVIDIA L40 GPU. For larger models, including Qwen3.5-27B and Qwen3.5-35B-A3B, evaluation is performed using two L40 GPUs to accommodate model inference. The runtime and peak memory reported in Table 5 are measured during reconstruction on a single NVIDIA L40 GPU with a batch size of 1.

A.2 DEFINITIONS OF RMSE VARIANTS

We provide the concrete definitions of the RMSE variants used in our experiments. Let $E = \widehat{X}_{i+1} - X_{i+1} \in \mathbb{R}^{B\times T\times C}$ denote the reconstruction error tensor at a quantization stage, where B, T, and C denote the batch size, number of tokens, and number of channels, respectively. The official implementations adopt different reconstruction batch sizes ($B = 1$ for OmniQuant and DuQuant, $B = 4$ for FlatQuant, and $B = 3$ for SliderQuant). We therefore define all grouping strategies within each sample, making the sample level the coarsest formulation shared consistently across methods.

RMSE-Sample. RMSE-Sample computes RMSE independently for each calibration sample and then averages over samples:

$$\mathcal{L}_{\text{Sample}} = \frac{1}{B}\sum_{b=1}^{B}\sqrt{\frac{1}{TC}\sum_{t=1}^{T}\sum_{c=1}^{C}E_{b,t,c}^2}. \tag{9}$$

RMSE-Channel. RMSE-Channel computes RMSE independently for each channel within each sample and then averages over all sample-channel pairs:

$$\mathcal{L}_{\text{Channel}} = \frac{1}{BC}\sum_{b=1}^{B}\sum_{c=1}^{C}\sqrt{\frac{1}{T}\sum_{t=1}^{T}E_{b,t,c}^2}. \tag{10}$$

RMSE-Token. RMSE-Token computes RMSE independently for each token and then averages over all tokens:

$$\mathcal{L}_{\text{Token}} = \frac{1}{BT}\sum_{b=1}^{B}\sum_{t=1}^{T}\sqrt{\frac{1}{C}\sum_{c=1}^{C}E_{b,t,c}^2}. \tag{11}$$

RMSE-Element. RMSE-Element computes RMSE independently for each feature element:

$$\mathcal{L}_{\text{Element}} = \frac{1}{BTC}\sum_{b=1}^{B}\sum_{t=1}^{T}\sum_{c=1}^{C}\sqrt{E_{b,t,c}^2}. \tag{12}$$

Since each group contains only one element, RMSE-Element is equivalent to the mean absolute reconstruction error. All variants share the same gradient normalization principle: within each group, the gradient norm depends on the group size rather than the local reconstruction loss magnitude. As the groups are equal-sized within each variant, the overall output-space gradient magnitude remains constant at $1/\sqrt{BTC}$. The variants therefore differ only in the feature grouping over which this normalization is applied.

B Full Downstream Evaluation Results

Table 6: Full downstream results under OmniQuant across diverse LLMs and quantization settings. Avg denotes the average accuracy over PIQA, ARC-Easy (ARC-e), ARC-Challenge (ARC-c), Winogrande (WG), HellaSwag (HS), and BoolQ.

Model	#Bits	Loss	PPL↓	PIQA↑	ARC-e↑	ARC-c↑	WG↑	HS↑	BoolQ↑	Avg↑
LLaMA2-7B	Original	–	5.47	79.11	74.54	46.25	69.06	76.00	77.71	70.45
	W4A16	MSE	5.75	78.35	71.63	42.83	69.53	73.90	73.73	68.33
		RMSE	5.70	78.02	73.57	43.43	68.35	74.46	76.57	69.07
	W3A16	MSE	6.62	75.90	65.87	38.57	65.43	69.70	68.87	64.05
		RMSE	6.38	76.77	69.99	42.49	67.32	72.04	72.08	66.78
	W2A16	MSE	36.92	57.73	35.48	25.68	50.59	35.97	58.44	43.98
		RMSE	16.06	63.93	43.22	26.88	56.20	47.15	62.45	49.97
	W2A16G128	MSE	11.05	66.21	46.89	27.73	56.75	51.65	60.73	51.66
		RMSE	9.10	70.62	59.05	31.74	59.67	57.01	65.66	57.29
	W4A4	MSE	17.42	62.35	46.38	27.73	54.46	46.94	56.67	49.09
		RMSE	11.48	69.42	57.41	33.70	55.01	59.86	61.96	56.23
LLaMA3.1-8B	Original	–	6.24	81.18	80.89	53.67	73.56	78.87	82.08	75.04
	W4A16	MSE	7.19	80.52	78.11	50.43	71.43	76.45	78.62	72.59
		RMSE	6.92	80.14	78.07	50.43	71.19	77.45	80.06	72.89
	W3A16	MSE	12.38	72.47	63.17	37.12	58.33	63.62	68.59	60.55
		RMSE	9.46	77.15	68.90	42.24	68.82	71.71	74.19	67.17
	W2A16	MSE	1021.33	54.03	31.10	21.84	50.59	26.57	37.86	37.00
		RMSE	140.63	60.01	39.86	25.09	50.83	32.15	46.73	42.44
	W2A16G128	MSE	124.49	54.41	29.67	21.84	53.83	29.83	38.84	38.07
		RMSE	20.21	62.68	46.68	27.56	53.51	44.03	57.16	48.60
	W4A4	MSE	110.88	54.46	32.07	22.87	51.46	35.33	44.34	40.09
		RMSE	44.29	53.32	33.84	24.57	52.72	44.11	52.48	43.51
Qwen3.5-9B	Original	–	8.65	80.09	74.33	55.80	73.01	78.15	89.20	75.10
	W4A16	MSE	9.28	78.89	76.64	54.52	71.19	76.22	87.89	74.23
		RMSE	9.14	80.25	75.67	55.89	71.59	76.43	86.27	74.35
	W3A16	MSE	11.56	76.77	77.61	52.13	68.82	70.33	80.92	71.10
		RMSE	11.21	77.04	79.63	54.18	69.46	72.06	84.40	72.79
	W2A16	MSE	99.05	57.18	42.47	25.77	52.96	32.23	62.02	45.44
		RMSE	51.56	63.77	59.43	36.86	56.75	45.64	66.24	54.78
	W2A16G128	MSE	23.61	64.20	55.47	30.38	54.70	45.91	70.58	53.54
		RMSE	17.98	68.55	65.57	36.69	59.59	54.62	75.17	60.03
	W4A4	MSE	69.74	58.05	44.87	27.90	51.93	40.62	49.24	45.43
		RMSE	51.88	62.84	47.10	31.14	51.85	47.61	55.66	49.37
Qwen3.5-27B	Original	–	6.90	82.21	79.76	61.69	78.85	83.43	76.70	77.10
	W4A16	MSE	7.17	82.21	79.92	61.43	77.74	82.62	82.32	77.71
		RMSE	7.09	82.64	81.94	64.16	77.98	82.37	81.10	78.37
	W3A16	MSE	8.42	81.39	84.55	62.63	76.01	79.77	78.38	77.12
		RMSE	8.26	81.61	84.68	63.57	76.64	80.20	82.11	78.13
	W2A16	MSE	37.51	68.55	71.34	44.80	60.62	53.46	63.91	60.45
		RMSE	22.20	74.59	78.11	52.47	68.90	67.99	83.67	70.96
	W2A16G128	MSE	11.73	74.76	80.39	52.82	66.93	67.65	82.72	70.88
		RMSE	11.29	75.73	81.73	55.55	70.48	70.01	85.05	73.09
	W4A4	MSE	29.45	68.34	63.34	39.93	56.75	60.12	68.26	59.46
		RMSE	16.13	74.86	73.15	50.17	63.69	69.50	73.67	67.51
Qwen3.5-35B-A3B	Original	–	6.59	83.68	79.17	62.37	74.43	82.31	88.07	78.34
	W4A16	MSE	7.22	82.54	81.23	61.01	73.17	80.84	86.91	77.61
		RMSE	7.06	81.77	80.43	61.52	73.09	81.28	89.82	77.98
	W3A16	MSE	9.87	79.76	78.62	58.62	70.24	75.03	75.99	73.05
		RMSE	8.83	79.43	77.23	57.59	69.93	77.39	85.41	74.50
	W2A16	MSE	73.79	61.26	42.17	26.11	49.33	37.52	64.16	46.76
		RMSE	21.85	69.80	56.78	37.63	56.51	56.34	68.47	57.59
	W2A16G128	MSE	24.66	69.10	57.53	35.49	53.59	54.84	68.26	56.47
		RMSE	13.90	74.81	74.07	49.15	61.09	62.02	79.42	66.76
	W4A4	MSE	12.42	72.52	68.10	44.54	59.51	71.40	77.74	65.63
		RMSE	10.82	75.08	66.75	46.33	63.54	74.36	78.81	67.48

Table 7: Full downstream results across representative learning-based PTQ methods under W4A4 quantization. Avg denotes the average accuracy over PIQA, ARC-Easy, ARC-Challenge, Winogrande, HellaSwag, and BoolQ.

Model	Method	Loss	PPL↓	PIQA↑	ARC-e↑	ARC-c↑	WG↑	HS↑	BoolQ↑	Avg↑
LLaMA2-7B	Original	–	5.47	79.11	74.54	46.25	69.06	76.00	77.71	70.45
	OmniQuant	MSE	17.42	62.35	46.38	27.73	54.46	46.94	56.67	49.09
		RMSE	11.48	69.42	57.41	33.70	55.01	59.86	61.96	56.23
	SliderQuant	MSE	8.48	70.46	54.92	32.85	58.09	60.60	66.30	57.20
		RMSE	8.14	72.09	60.69	34.98	60.77	62.81	67.09	59.74
	DuQuant	MSE	6.07	76.71	69.11	42.49	67.88	72.01	73.24	66.91
		RMSE	6.01	77.31	70.62	43.77	67.01	72.72	74.71	67.69
	FlatQuant	MSE	5.82	77.42	71.55	43.17	66.30	74.15	73.70	67.72
		RMSE	5.78	77.20	73.61	43.52	68.03	74.29	75.93	68.76
Qwen3.5-9B	Original	–	8.65	80.09	74.33	55.80	73.01	78.15	89.20	75.10
	OmniQuant	MSE	69.74	58.05	44.87	27.90	51.93	40.62	49.24	45.43
		RMSE	51.88	62.84	47.10	31.14	51.85	47.61	55.66	49.37
	SliderQuant	MSE	21.41	69.64	64.39	40.44	58.96	58.64	70.49	60.43
		RMSE	20.64	69.21	66.58	42.24	62.19	59.94	71.35	61.92
	DuQuant	MSE	10.95	76.06	73.86	47.18	66.38	71.68	83.95	69.85
		RMSE	10.54	77.48	73.11	50.09	67.64	72.77	82.51	70.60
	FlatQuant	MSE	10.26	77.75	73.65	53.50	68.59	74.92	84.95	72.23
		RMSE	9.15	79.16	76.56	55.38	70.88	76.15	87.00	74.19

In the main paper, we report the average downstream accuracy for compactness. In this section, we provide the full per-task results on PIQA, ARC-Easy, ARC-Challenge, Winogrande, HellaSwag, and BoolQ. These results provide a more detailed view of the downstream behavior of MSE and RMSE across different models, quantization settings, and PTQ methods.

C Additional Results of RMSE Variants

Tables 8, 9, 10, 11, and 12 provide additional comparisons of RMSE-Sample, RMSE-Channel, RMSE-Token, and RMSE-Element across learning-based PTQ methods, model families, and quantization settings. All four variants consistently outperform MSE across the evaluated configurations, further supporting the effectiveness of the shared gradient normalization principle across different feature groupings. While the best-performing variant varies across individual configurations, we observe an overall model-family-level preference: RMSE-Token generally performs favorably on LLaMA-family models, whereas RMSE-Channel performs favorably on Qwen-family models. Accordingly, unless otherwise specified, we adopt these variants as the respective defaults in the main experiments. These defaults reflect their overall empirical performance rather than per-configuration optimality. We report broad representative comparisons rather than exhaustively evaluating every possible configuration.

Table 8: Comparison of RMSE variants with DuQuant under W4A4 quantization. Avg denotes the average accuracy over PIQA, ARC-Easy, ARC-Challenge, Winogrande, HellaSwag, and BoolQ.

Model	Loss	PPL↓	PIQA↑	ARC-e↑	ARC-c↑	WG↑	HS↑	BoolQ↑	Avg↑
LLaMA2-7B	MSE	6.07	76.71	69.11	42.49	67.88	72.01	73.24	66.91
	RMSE-Sample	6.02	76.28	70.03	42.49	67.88	72.88	74.10	67.28
	RMSE-Channel	6.01	78.02	71.55	43.69	66.06	72.76	73.61	67.61
	RMSE-Token	6.01	77.31	70.62	43.77	67.01	72.72	74.71	67.69
	RMSE-Element	6.01	77.31	70.54	42.66	65.59	72.68	73.55	67.06
Qwen3.5-9B	MSE	10.95	76.06	73.86	47.18	66.38	71.68	83.95	69.85
	RMSE-Sample	10.41	77.20	75.13	52.65	67.32	72.56	82.05	71.15
	RMSE-Channel	10.54	77.48	73.11	50.09	67.64	72.77	82.51	70.60
	RMSE-Token	10.38	77.26	74.12	51.54	66.93	72.86	78.81	70.25
	RMSE-Element	10.37	77.37	73.65	50.60	66.14	73.34	80.24	70.22

Table 9: Comparison of RMSE variants with OmniQuant on LLaMA2-7B across quantization settings. Avg denotes the average accuracy over PIQA, ARC-Easy, ARC-Challenge, Winogrande, HellaSwag, and BoolQ.

#Bits	Loss	PPL↓	PIQA↑	ARC-e↑	ARC-c↑	WG↑	HS↑	BoolQ↑	Avg↑
W4A16	MSE	5.75	78.35	71.63	42.83	69.53	73.90	73.73	68.33
	RMSE-Sample	5.70	79.05	73.23	43.52	68.51	74.19	74.77	68.88
	RMSE-Channel	5.70	78.40	73.53	44.11	69.46	74.33	76.15	69.33
	RMSE-Token	5.70	78.02	73.57	43.43	68.35	74.46	76.57	69.07
	RMSE-Element	5.71	78.29	72.90	42.06	68.35	74.68	75.54	68.64
W3A16	MSE	6.62	75.90	65.87	38.57	65.43	69.70	68.87	64.05
	RMSE-Sample	6.49	75.14	63.72	38.31	66.14	71.30	70.40	64.17
	RMSE-Channel	6.45	77.04	68.10	41.21	67.64	71.56	69.72	65.88
	RMSE-Token	6.38	76.77	69.99	42.49	67.32	72.04	72.08	66.78
	RMSE-Element	6.43	77.04	68.31	41.72	67.80	72.30	72.32	66.58
W2A16	MSE	36.92	57.73	35.48	25.68	50.59	35.97	58.44	43.98
	RMSE-Sample	25.58	57.73	37.50	26.19	52.80	38.19	58.13	45.09
	RMSE-Channel	22.46	63.00	38.59	25.68	52.57	39.70	56.51	46.01
	RMSE-Token	16.06	63.93	43.22	26.88	56.20	47.15	62.45	49.97
	RMSE-Element	22.78	61.04	41.75	26.11	53.12	41.45	57.09	46.76
W4A4	MSE	17.42	62.35	46.38	27.73	54.46	46.94	56.67	49.09
	RMSE-Sample	14.15	63.44	47.10	29.18	54.38	53.00	63.30	51.73
	RMSE-Channel	13.14	67.52	52.57	31.48	53.28	57.31	58.23	53.40
	RMSE-Token	11.48	69.42	57.41	33.70	55.01	59.86	61.96	56.23
	RMSE-Element	13.64	67.90	55.13	34.39	55.41	55.58	57.68	54.35
W2A16G128	MSE	11.05	66.21	46.89	27.73	56.75	51.65	60.73	51.66
	RMSE-Sample	9.20	70.02	55.64	31.23	60.14	56.61	65.84	56.58
	RMSE-Channel	9.13	71.33	57.24	32.68	60.30	57.13	65.50	57.36
	RMSE-Token	9.10	70.62	59.05	31.74	59.67	57.01	65.66	57.29
	RMSE-Element	9.22	70.89	57.58	32.17	61.72	56.26	66.61	57.54

Table 10: Comparison of RMSE variants with OmniQuant on LLaMA3.1-8B across quantization settings. Avg denotes the average accuracy over PIQA, ARC-Easy, ARC-Challenge, Winogrande, HellaSwag, and BoolQ.

#Bits	Loss	PPL↓	PIQA↑	ARC-e↑	ARC-c↑	WG↑	HS↑	BoolQ↑	Avg↑
W4A16	MSE	7.19	80.52	78.11	50.43	71.43	76.45	78.62	72.59
	RMSE-Sample	6.96	80.90	77.95	49.66	71.74	77.17	79.20	72.77
	RMSE-Channel	6.96	79.92	78.45	51.11	71.67	77.34	78.75	72.87
	RMSE-Token	6.92	80.14	78.07	50.43	71.19	77.45	80.06	72.89
	RMSE-Element	6.96	79.76	77.27	50.43	71.27	77.23	79.97	72.65
W3A16	MSE	12.38	72.47	63.17	37.12	58.33	63.62	68.59	60.55
	RMSE-Sample	9.88	76.82	71.42	43.26	67.48	70.16	68.87	66.34
	RMSE-Channel	9.56	77.42	67.34	42.83	67.56	71.29	75.66	67.02
	RMSE-Token	9.46	77.15	68.90	42.24	68.82	71.71	74.19	67.17
	RMSE-Element	9.58	76.61	71.80	44.28	67.01	71.81	74.59	67.68
W2A16	MSE	1021.33	54.03	31.10	21.84	50.59	26.57	37.86	37.00
	RMSE-Sample	941.96	54.46	32.24	22.27	49.33	26.58	38.07	37.16
	RMSE-Channel	194.77	56.47	36.57	24.74	51.14	30.76	45.17	40.81
	RMSE-Token	140.63	60.01	39.86	25.09	50.83	32.15	46.73	42.44
	RMSE-Element	368.30	55.98	34.05	22.35	50.83	27.15	38.41	38.13
W4A4	MSE	110.88	54.46	32.07	22.87	51.46	35.33	44.34	40.09
	RMSE-Sample	68.36	55.50	32.32	22.27	48.54	39.49	50.95	41.51
	RMSE-Channel	55.16	53.65	30.35	24.15	52.64	43.85	49.05	42.28
	RMSE-Token	44.29	53.32	33.84	24.57	52.72	44.11	52.48	43.51
	RMSE-Element	59.20	57.07	34.68	23.21	52.64	37.88	52.17	42.94
W2A16G128	MSE	124.49	54.41	29.67	21.84	53.83	29.83	38.84	38.07
	RMSE-Sample	20.52	63.11	45.79	26.45	52.72	43.92	61.74	48.96
	RMSE-Channel	20.91	64.74	47.14	27.05	55.41	45.55	58.20	49.68
	RMSE-Token	20.21	62.68	46.68	27.56	53.51	44.03	57.16	48.60
	RMSE-Element	23.72	61.26	45.03	26.02	53.12	43.86	55.20	47.42

Table 11: Comparison of RMSE variants with OmniQuant on Qwen3.5-9B across quantization settings. Avg denotes the average accuracy over PIQA, ARC-Easy, ARC-Challenge, Winogrande, HellaSwag, and BoolQ.

#Bits	Loss	PPL↓	PIQA↑	ARC-e↑	ARC-c↑	WG↑	HS↑	BoolQ↑	Avg↑
W4A16	MSE	9.28	78.89	76.64	54.52	71.19	76.22	87.89	74.23
	RMSE-Sample	9.14	79.82	78.62	56.23	70.48	76.46	88.20	74.97
	RMSE-Channel	9.14	80.25	75.67	55.89	71.59	76.43	86.27	74.35
	RMSE-Token	9.20	79.16	78.87	58.11	71.03	76.49	86.79	75.08
	RMSE-Element	9.20	79.05	77.40	56.57	70.88	76.45	87.83	74.70
W3A16	MSE	11.56	76.77	77.61	52.13	68.82	70.33	80.92	71.10
	RMSE-Sample	11.32	77.75	80.01	53.84	68.82	71.62	87.00	73.17
	RMSE-Channel	11.21	77.04	79.63	54.18	69.46	72.06	84.40	72.79
	RMSE-Token	11.16	77.69	77.19	52.65	69.06	71.51	86.73	72.47
	RMSE-Element	11.29	77.58	77.82	53.24	70.24	72.11	86.54	72.92
W2A16	MSE	99.05	57.18	42.47	25.77	52.96	32.23	62.02	45.44
	RMSE-Sample	82.46	58.32	51.52	26.96	52.57	34.12	62.17	47.61
	RMSE-Channel	51.56	63.77	59.43	36.86	56.75	45.64	66.24	54.78
	RMSE-Token	88.41	58.16	49.62	28.41	52.25	34.37	62.97	47.63
	RMSE-Element	61.76	62.24	56.48	33.11	57.62	43.24	63.85	52.76
W4A4	MSE	69.74	58.05	44.87	27.90	51.93	40.62	49.24	45.43
	RMSE-Sample	66.52	59.47	48.91	30.97	50.51	42.26	54.98	47.85
	RMSE-Channel	51.88	62.84	47.10	31.14	51.85	47.61	55.66	49.37
	RMSE-Token	62.15	59.14	46.34	29.44	51.22	42.04	56.12	47.38
	RMSE-Element	54.12	61.48	46.59	29.44	51.38	46.66	57.19	48.79
W2A16G128	MSE	23.61	64.20	55.47	30.38	54.70	45.91	70.58	53.54
	RMSE-Sample	17.85	67.36	67.05	37.46	61.01	53.91	79.69	61.08
	RMSE-Channel	17.98	68.55	65.57	36.69	59.59	54.62	75.17	60.03
	RMSE-Token	18.06	67.74	65.66	37.88	61.17	53.87	73.00	59.89
	RMSE-Element	18.68	68.50	61.62	35.92	58.80	53.17	73.76	58.63

Table 12: Comparison of RMSE variants with OmniQuant on Qwen3.5-27B under W2A16 quantization. Avg denotes the average accuracy over PIQA, ARC-Easy, ARC-Challenge, Winogrande, HellaSwag, and BoolQ.

#Bits	Loss	PPL↓	PIQA↑	ARC-e↑	ARC-c↑	WG↑	HS↑	BoolQ↑	Avg↑
W2A16	MSE	37.51	68.55	71.34	44.80	60.62	53.46	63.91	60.45
	RMSE-Sample	24.28	72.96	77.90	50.60	66.46	61.73	69.36	66.50
	RMSE-Channel	22.20	74.59	78.11	52.47	68.90	67.99	83.67	70.96
	RMSE-Token	25.04	72.58	76.14	47.70	65.98	59.64	72.08	65.69
	RMSE-Element	24.53	72.85	74.62	49.83	68.11	64.07	84.28	68.96

D Additional Analysis of Optimization Dynamics

Complementing the optimization statistics presented in Figs. 2 and 3, Figs. 5, 6, and 7 provide additional representative results across learning-based PTQ methods, model families, and quantization settings. These figures further compare all four RMSE variants, demonstrating consistent balanced optimization behavior across different feature groupings.

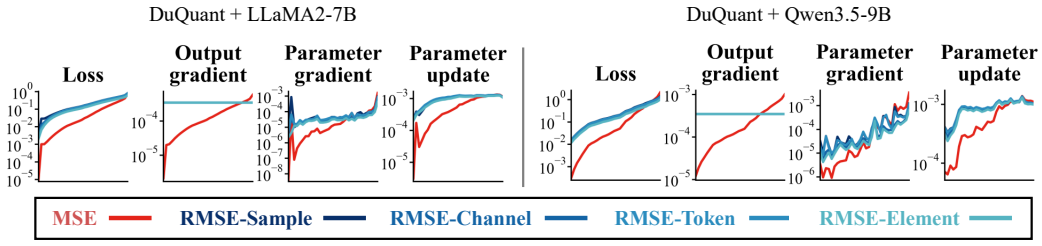

Figure 5: Optimization statistics under MSE and different RMSE variants with DuQuant on LLaMA2-7B and Qwen3.5-9B under W4A4 quantization. The horizontal axis denotes the quantization stages from early to late.

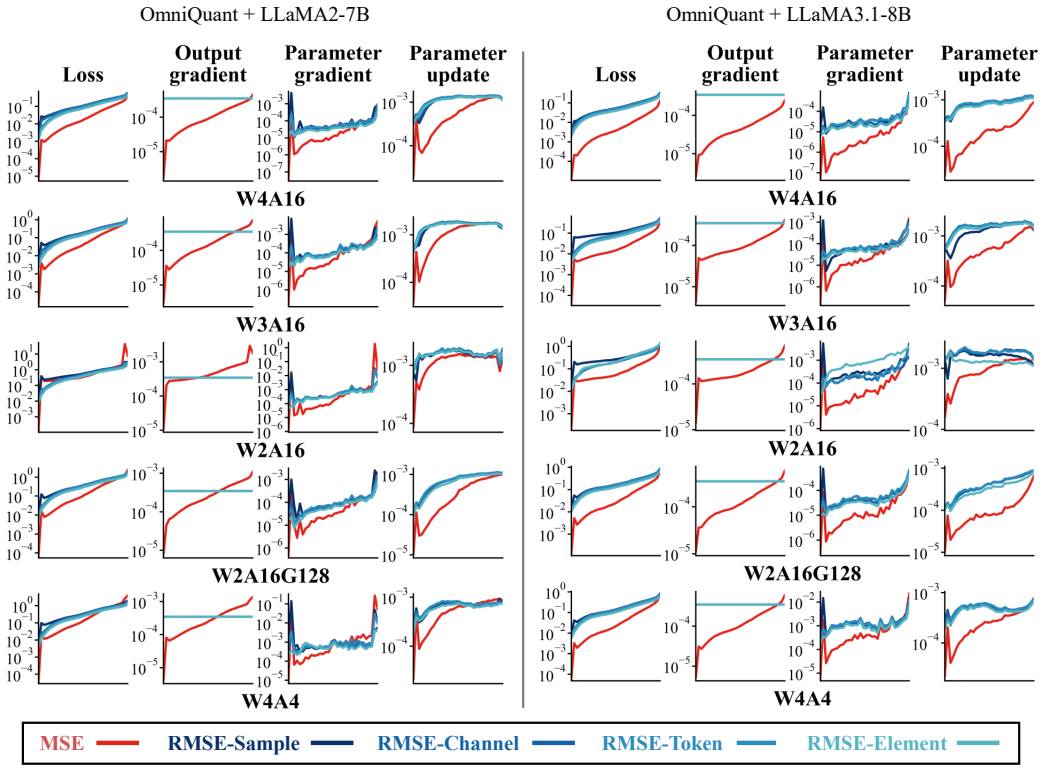

Figure 6: Optimization statistics under MSE and different RMSE variants with OmniQuant on LLaMA2-7B and LLaMA3.1-8B across quantization settings. The horizontal axis denotes the quantization stages from early to late.

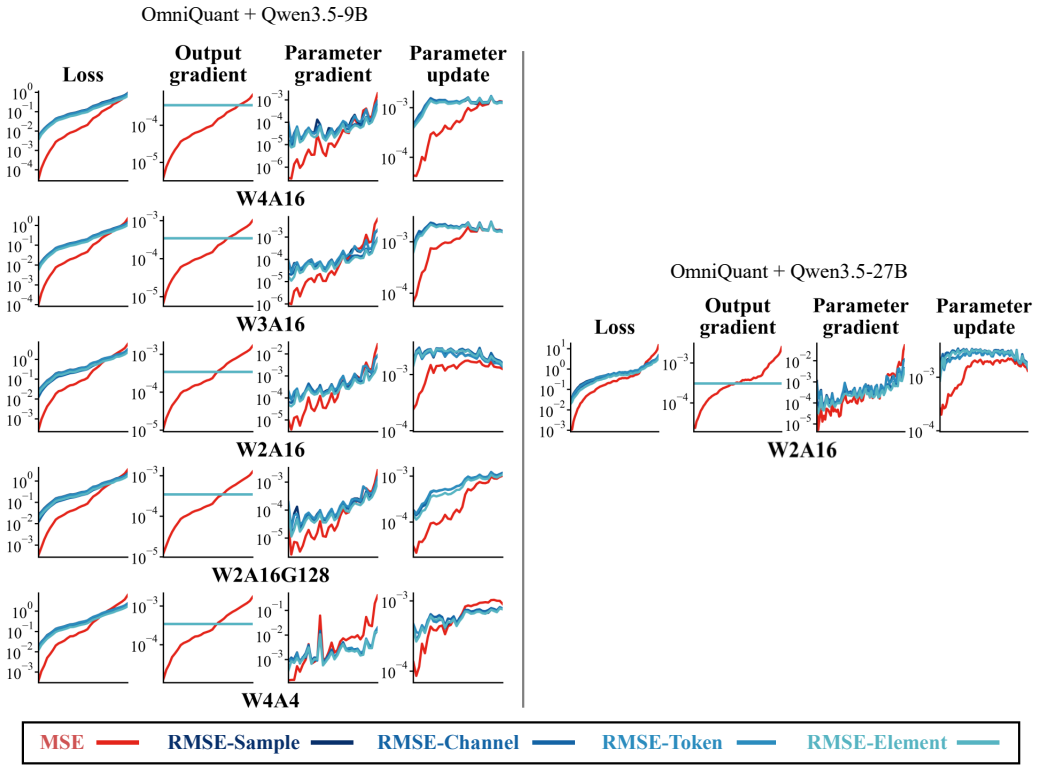

Figure 7: Optimization statistics under MSE and different RMSE variants with OmniQuant. Results are reported across multiple quantization settings on Qwen3.5-9B and under W2A16 quantization on Qwen3.5-27B. The horizontal axis denotes the quantization stages from early to late.

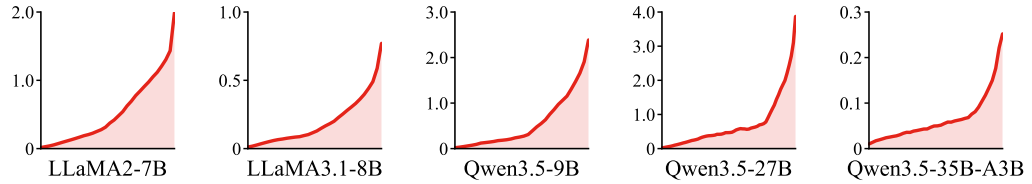

Figure 8: Mean absolute magnitudes of high-precision block outputs across model depth. The horizontal axis denotes the block order from early to late, and the vertical axis reports the mean absolute output magnitude, $\|X_{i+1}\|_1/N$, of each block. Statistics are collected from the original high-precision models using 128 WikiText2 samples with a sequence length of 2048.

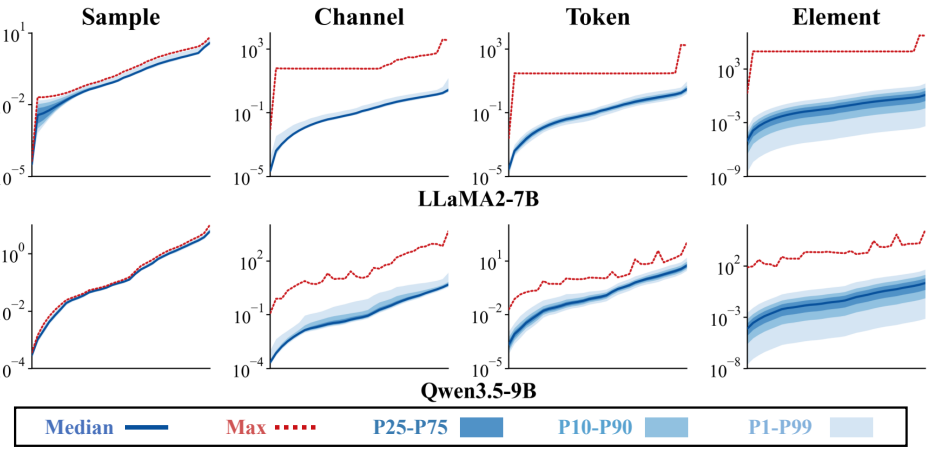

Figure 9: Distributions of group-wise reconstruction losses across blocks. For the fully quantized model, we partition the high-precision and quantized outputs of each block into groups at the sample, channel, token, and element levels, and compute the MSE loss independently within each group. The horizontal axis denotes the block order from early to late, and the vertical axis reports the group-wise MSE. The solid line denotes the median across groups, the shaded regions show different percentile intervals, and the red dotted line denotes the maximum. Results are collected from OmniQuant models trained with MSE under W4A4 quantization.

E Output Magnitudes Across Blocks

As discussed in Sec. 2.2, reconstruction losses are computed directly over unit outputs, making their magnitudes inherently sensitive to the magnitudes of the reconstructed features. To examine this factor directly, we measure the mean absolute magnitude of the high-precision block outputs,

$$\frac{1}{N}\|X_{i+1}\|_1\,, \tag{13}$$

using 128 WikiText2 calibration samples. As shown in Fig. 8, output magnitudes vary substantially across blocks and generally increase with model depth across all evaluated model families, scales, and architectures. Since these statistics are collected from high-precision models without quantization, the observed variation is intrinsic to the model outputs rather than introduced by the quantization process. This cross-block variation provides a natural source of the broad reconstruction loss scale observed during sequential quantization.

F Reconstruction Loss Magnitude Variation Within Stages

To examine whether reconstruction loss magnitudes also vary within each unit, we analyze OmniQuant models trained with MSE and compute the group-wise MSE between the high-precision and quantized unit outputs using the four feature groupings introduced in Sec. 3.2. As shown in Fig. 9, reconstruction loss magnitudes vary substantially across samples, with increasingly dispersed distributions under the channel, token, and element groupings. This within-unit variation further demonstrates that feature grouping is a meaningful design dimension for applying gradient normalization within each unit.