Why Do Some Inputs Break Low-Bit LLM Quantization?

Ting-Yun Chang **Muru Zhang** **Jesse Thomason** **Robin Jia**
University of Southern California, Los Angeles, CA, USA
{tingyun, muruzhan, jessetho, robinjia}@usc.edu

Abstract

Low-bit weight-only quantization significantly reduces the memory footprint of large language models (LLMs), but disproportionately affects certain examples. We analyze diverse 3–4 bit methods on LLMs ranging from 7B–70B in size and find that the quantization errors of 50 pairs of methods are strongly correlated (avg. $\rho = 0.82$) on FineWeb examples. Moreover, the residual stream magnitudes of full-precision models are indicative of future quantization errors. We further establish a hypothesis that relates the residual stream magnitudes to error amplification and accumulation over layers. Using LLM localization techniques, early exiting, and activation patching, we show that examples with large errors rely on precise residual activations in the later layers, and that the outputs of MLP gates play a crucial role in maintaining the perplexity. Our work reveals why certain examples result in large quantization errors and which model components are most critical for performance preservation.

1 Introduction

Many large language models (LLMs; AI@Meta, 2024; Gemma et al., 2024; OLMo et al., 2024; Qwen, 2025) use 16-bit precision to represent each parameter by default, requiring substantial GPU memory for inference. Quantization reduces memory usage by reducing the bit-precision of model weights or activations. Low-bit weight-only quantization (Frantar et al., 2023; Sheng et al., 2023; Park et al., 2024; Lee et al., 2024), which quantizes model weights to ≤ 4 bits and keeps the activations in 16 bits, greatly reduces the memory footprint while maintaining reasonable performance across various benchmarks (Kurtic et al., 2024).

However, Dettmers and Zettlemoyer (2023) find that at 3-bits, LLMs start to show noticeable accuracy degradation. Kumar et al. show that LLMs trained on more data become harder to quantize. Prior efforts focus on *outliers*, the abnormally large activations or weights that cause large errors in LLM quantization (Dettmers et al., 2022). Quantization methods that combat outliers lead to overall better performance (Xiao et al., 2023; Chee et al., 2023; Ashkboos et al., 2024; Luo et al., 2025); however, we observe that some examples still suffer from large quantization errors under various methods. Why does quantization disproportionately affect certain examples? What factors are indicative of quantization errors? Which parts of the LLMs lead to large errors? We study these underexplored questions across diverse 3- and 4-bit weight-only quantization methods.

First, we observe that the quantization errors of various pairs of methods are highly correlated (avg. $\rho = 0.82$) and that certain examples yield large errors across methods, where we define errors as the increase in negative log-likelihoods on sampled pretraining documents after quantization. A possible hypothesis is that the errors are largely predetermined by the full-precision model. Since Lin et al. (2024) show that activation outliers amplify the errors in quantized weights, we examine whether the activation magnitudes of the full-precision models are indicative of the future quantization errors.

Surprisingly, we find strong *negative* correlations between quantization errors and the magnitudes of the residual stream. For instance, the correlations computed on the residuals across the last 30 layers of Llama3-70B often exceed -0.6 and reach -0.8 in the final layer. The strong correlations support our hypothesis that the full-precision model largely decides the future quantization errors. Meanwhile, it is unexpected that smaller residual activations are associated with large quantization errors. Our further analysis reveals that the RMS LayerNorm (Zhang and Sennrich, 2019) tends to *reverse* the relative activation magnitudes. These findings altogether explain why certain examples have large quantization errors across methods: examples that inherently have smaller residual magni-

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 3411–3430
November 4-9, 2025

tudes will have larger post-RMSNorm magnitudes over multiple layers, which continually amplify the errors in the quantized weights and ultimately result in large quantization errors in outputs.

We further apply LLM localization techniques to study where quantization errors arise from. First, we use early exiting (nostalgebraist, 2020) to project the residual hidden states across layers to the output probabilities. Early exiting reveals that examples with large quantization errors rely heavily on later layers to adjust the model output distributions. However, restoring the weights of later layers to 16 bits only marginally reduces perplexity, implying that the problem lies with the accumulated errors in activations over layers. Next, we adapt cross-model activation patching (Prakash et al., 2024), which restores specific activations of the quantized model to the clean counterparts from the full-precision model. We find that restoring the outputs of the MLP gate projection (Shazeer, 2020) can substantially reduce the perplexity.

Finally, we investigate what kinds of data tend to have large quantization errors. We classify the large-error examples and find that they cover diverse topics. Because Ahia et al. (2021); Marchisio et al. (2024) show that model compression has a disparate impact on long-tail examples, we also study the relationship between quantization errors and long-tail examples. Surprisingly, our results on both FineWeb (Penedo et al., 2024) and PopQA (Mallen et al., 2023) datasets suggest that quantization has a milder impact on long-tail examples and that well-represented examples are not immune to large degradation in log-likelihood and accuracy.

Our work centers on why, where, and what leads to large quantization errors, offering systematic analyses across diverse methods. We provide a mechanistic explanation of why certain examples suffer from large quantization errors and identify key model components that are critical to maintaining the perplexity. Our study could motivate future methods to reduce errors from MLP gate projections or to introduce LayerNorm scaling factors (Wei et al., 2023; Lin et al., 2024) that consider the reversal effects in RMSNorm.[1]

2 Background

This section introduces the quantization methods in our study, covering diverse, widely-used approaches for low-bit weight-only quantization.

[1]Code: https://github.com/terarachang/QError

GPTQ. Frantar et al. (2023) propose GPTQ, an approximate second-order method that quantizes one weight at a time while updating not-yet-quantized weights to compensate for the quantization loss. GPTQ minimizes the loss on a small calibration set of C4 (Raffel et al., 2020) samples. We adopt common GPTQ settings, using activation sorting and a Hessian damping coefficient of 0.01 when not otherwise specified.

AWQ. Lin et al. (2024) observe that the activation outliers magnify the quantization errors in the corresponding weight dimensions. They propose AWQ, which smoothes the activations by introducing a scaling factor corresponding to each input dimension of weights. AWQ grid-searches the scaling factors based on the activation magnitudes.

NormalFloat. Dettmers et al. (2023) propose NormalFloat (NF), a 4-bit data type that is optimal for normally distributed weights based on information theory. NF employs non-uniform quantization with Gaussian quantiles and requires a lookup table. Guo et al. (2024) accelerate the lookup table operation and extend NF to lower bit precision.

EfficientQAT. Chen et al. (2024) propose EfficientQAT (EQAT), a lightweight quantization-aware fine-tuning (Liu et al., 2024) method. EQAT applies straight-through estimator (Bengio et al., 2013) for backpropagation.

LayerNorm in Quantization. Kovaleva et al. (2021); Wei et al. (2022) find that outliers in the LayerNorm parameters of BERT (Devlin et al., 2019) cause difficulties in model compression. Given the importance of LayerNorm, all the quantization methods we discuss above leave LayerNorm unquantized. Specifically, LLMs in Llama (Touvron et al., 2023), Mistral (Jiang et al., 2023), and Qwen (Bai et al., 2023) families use RMSNorm (Zhang and Sennrich, 2019):

$$\text{RMSNorm}(z) = \frac{z}{\text{RMS}(z)} \odot \gamma, \tag{1}$$

$$\text{RMS}(z) = \sqrt{\frac{1}{d}\sum_{i=1}^{d} z_i^2}, \tag{2}$$

where z is the d-dimensional input, $\odot$ denotes element-wise multiplication, and $\gamma \in \mathbb{R}^d$ is the unqauntized weights for affine transformation.

3 Problem Setups

3.1 Datasets

FineWeb. We create a dataset $\mathcal{D}$ of 10k examples sampled from the FineWeb corpus (Penedo et al., 2024), consisting of deduplicated English web documents. We evaluate the negative log-likelihood (NLL) of different models on $\mathcal{D}$. Formally, given a model θ and an example $x = [x_0, x_1, \ldots, x_T]$ of T tokens, prepended with a start of sentence token x_0, we define the NLL of x as:

$$\text{NLL}(x;\theta) = \frac{1}{T}\sum_{t=1}^{T} -\log p_\theta(x_t|x_{<t}), \quad (3)$$

where p_θ is the probability distribution of the model θ. We only sample from long documents and truncate all the sampled documents to 512 tokens.

Quantization Errors. NLL is a common loss term for LLM pretraining, and prior quantization research uses perplexity (exponentiated average NLL) as a standard metric. Therefore, we define the quantization error on a FineWeb example x as the increase in NLL. Formally, given the full-precision LLM θ and its quantized counterpart $\tilde{\theta}$, we define the quantization error of $\tilde{\theta}$ on x as:

$$e(x;\tilde{\theta}) = \text{NLL}(x;\tilde{\theta}) - \text{NLL}(x;\theta) \quad (4)$$

Large-Error Set and Control Set. For fine-grained analyses, we create a large-error set $\mathcal{D}_{\text{large}} \subset \mathcal{D}$, consisting of the top-10% (1k) examples by quantization errors and a control set $\mathcal{D}_{\text{ctrl}}$ of 1k examples sampled from the bottom-50%.[2]

PopQA. We use PopQA (Penedo et al., 2024) to study how quantization affects LLMs' factual knowledge of entities under different popularity levels. PopQA contains 14k questions, each supplemented with the entity popularity. Following Penedo et al. (2024), we use 15-shot prompting, greedy decoding, and exact match accuracy.

3.2 Models and Quantization Methods

We study four LLMs: Qwen2.5-7B (Qwen, 2024), Llama3-8B (AI@Meta, 2024), Mistral-Nemo-12B,[3] and Llama3-70B. We study quantization methods in the 3–4 bit, weight-only quantization regime: GPTQ, AWQ, NF, and EQAT (see §2). We follow original implementations and detail the calibration sets, quantization configurations, and perplexity of each method in appendix A.

[2] See appendix B for detailed dataset construction.

[3] https://mistral.ai/news/mistral-nemo

4 Do Methods Fail on the Same Inputs?

First, we investigate whether diverse quantization methods cause large errors on the same inputs. Recall that we define the example-level quantization error $e(x^{(i)};\tilde{\theta}) \in \mathbb{R}$ in Eq. 4. Given a dataset $\mathcal{D} = \{x^{(i)}\}_{i=1}^{N}$ of N examples, we define the errors of the quantized model $\tilde{\theta}$ on the dataset $\mathcal{D}$ as:

$$\mathcal{E}(\mathcal{D};\tilde{\theta}) = [e(x^{(1)};\tilde{\theta}), \ldots, e(x^{(N)};\tilde{\theta})] \in \mathbb{R}^N,$$

Let $\tilde{\theta}_1$ and $\tilde{\theta}_2$ be the quantized models of the same base LLM θ, produced using different methods. We compute their quantization errors on the dataset, $\mathcal{E}_1 \triangleq \mathcal{E}(\mathcal{D};\tilde{\theta}_1)$ and $\mathcal{E}_2 \triangleq \mathcal{E}(\mathcal{D};\tilde{\theta}_2)$, respectively, and report the Pearson correlation between the errors, denoted as $\rho(\mathcal{E}_1, \mathcal{E}_2)$.

4.1 Strong Correlations Between Methods

Table 1 shows the correlations between the quantization errors of every two methods. We include the results of Qwen2.5-7B and Llama3-8B in Table 6 in appendix. Across four LLMs, we observe high correlations, an average of $\rho(\mathcal{E}_1, \mathcal{E}_2) = 0.82$ over all 50 pairs of methods, although Qwen2.5-7B has lower correlations (avg. 0.66) compared to other LLMs. The overall strong correlation indicates that diverse methods have a similar effect on

Model	(Method1, Method2)	$\rho(\mathcal{E}_1, \mathcal{E}_2)$
Mistral-12B	(AWQ4 , NF4)	0.78
	(AWQ4 , GPTQ4)	0.86
	(AWQ4 , NF3)	0.80
	(AWQ4 , GPTQ3)	0.83
	(AWQ3 , NF3)	0.90
	(AWQ3 , GPTQ3)	0.95
	(NF4 , GPTQ4)	0.82
	(NF4 , AWQ3)	0.81
	(NF4 , GPTQ3)	0.78
	(NF3 , GPTQ3)	0.89
	(GPTQ4, AWQ3)	0.90
	(GPTQ4, NF3)	0.85
Llama3-70B	(AWQ4 , NF4)	0.80
	(AWQ4 , GPTQ4)	0.96
	(AWQ4 , GPTQ3)	0.86
	(AWQ4 , EQAT3)	0.81
	(AWQ3 , EQAT3)	0.85
	(NF4 , GPTQ4)	0.81
	(NF4 , GPTQ3)	0.80
	(NF4 , AWQ3)	0.81
	(NF4 , EQAT3)	0.75
	(GPTQ4, AWQ3)	0.94
	(GPTQ4, EQAT3)	0.83
	(GPTQ3 , AWQ3)	0.96
	(GPTQ3, EQAT3)	0.83

Table 1: The quantization errors of different methods are strongly correlated on the FineWeb. The numbers in the method names denote 3- or 4-bit precision.

LLMs' likelihoods. We further find that the top-10% large-error examples of different methods on the same LLMs are highly overlapped, with an average Jaccard similarity of 0.51, far above the ~ 0.05 expected values by chance.

Since the configurations of the GPTQ algorithm can greatly affect the quantization outcomes (Frantar et al., 2023; Chen et al., 2025), we further examine correlations in quantization errors across different GPTQ variants. We focus on two factors: (1) the damping coefficient that improves the numerical stability of the inverse Hessian, varied over the range $[0.005, 0.01, 0.02]$, and (2) whether activation-based sorting is applied to change the weight quantization order. We find that the 3-bit Qwen2.5-7B model variants produced by different GPTQ configurations show a strong average correlation (0.93) in quantization errors on FineWeb (see Table 7 in the appendix).

5 Why Do Examples Have Large Errors?

We further explore why certain examples have large quantization errors across methods. Inspired by Lin et al. (2024), we first study whether the activation magnitudes from the full-precision models are predictive of the quantization errors.

5.1 Notations

Formally, a Transformer layer (Vaswani et al., 2017) in modern LLMs consists of a multi-headed attention (MHA), an MLP, and two Pre-LNs (Child et al., 2019), LN_1 and LN_2:

$$r^{(l)} = z^{(l-1)} + \text{MHA}^{(l)}\left(\text{LN}_1^{(l)}(z^{(l-1)})\right), \quad (5)$$

$$z^{(l)} = r^{(l)} + \text{MLP}^{(l)}\left(\text{LN}_2^{(l)}(r^{(l)})\right), \quad (6)$$

where l is the layer index and $z^{(0)} \in \mathbb{R}^d$ is the input to the first layer. We observe that the representations after the residual operations, $r^{(l)}$ and $z^{(l)}$, are both indicative of quantization errors. As $r^{(l)}$ shows clearer patterns, we name it as residual state and include the results of $z^{(l)}$ in appendix. Note that all the studied methods quantize the weights in MHA and MLP, but keep the weights γ in LN_1 and LN_2 in 16 bits (see §2).

Given an example x of T tokens, we quantify the magnitude of its residual states at layer l by averaging the Euclidean norm of each token representation $r_t^{(l)} \in \mathbb{R}^d$ over all T tokens:

$$\text{norm}\left(x; r^{(l)}\right) = \frac{1}{T}\sum_{t=1}^{T} \|r_t^{(l)}\|_2 \quad (7)$$

We name $\text{norm}\left(x; r^{(l)}\right) \in \mathbb{R}$ as the residual magnitude of example x at layer l. Given a dataset $\mathcal{D} = \{x^{(i)}\}_{i=1}^N$ of N examples, we compute $\text{norm}\left(x; r^{(l)}\right)$ for each example $x^{(i)}$ and denote the residual magnitudes on $\mathcal{D}$ as $\mathbb{S}(\mathcal{D}; \theta; l) \in \mathbb{R}^N$.

Finally, given an full-precision model θ with a layer index l, a quantized model $\tilde{\theta}$, and a dataset $\mathcal{D}$ of N examples, we calculate the Pearson correlation between the residual magnitudes $\mathbb{S}(\mathcal{D}; \theta; l)$ and the quantization errors $\mathcal{E}(\mathcal{D}; \tilde{\theta})$, denoted as $\rho\left(\mathbb{S}(l), \mathcal{E}\right)$. Note that $\mathcal{E}(\mathcal{D}; \tilde{\theta})$ measures the final errors of the quantized model and thus does not depend on the layer index l.

5.2 Residual Magnitudes From Full-Precision Models Indicates Quantization Errors

We observe a strong negative correlation between the quantization error $\mathcal{E}(\mathcal{D}; \tilde{\theta})$ and the residual magnitudes $\mathbb{S}(\mathcal{D}; \theta; l)$ computed on the last few layers of the full-precision model, where $\mathcal{D}$ is the FineWeb dataset of 10k examples. For example, let $l := L$ be the last Transformer layer, we report the correlation $\rho\left(\mathbb{S}(L), \mathcal{E}\right)$ of different LLMs and quantization methods in Table 2. The average correlation over 3-bit methods on Qwen2.5-7B, Llama3-8B, Mistral-12B, and Llama3-70B are $-0.66, -0.76, -0.78$, and -0.80. As shown in

Model	Method	$\rho\left(\mathbb{S}(L), \mathcal{E}\right)$
Qwen2.5-7B	AWQ3	-0.66
	NF3	-0.58
	GPTQ3	-0.74
Llama3-8B	AWQ3	-0.76
	NF3	-0.76
	EQAT3	-0.75
Mistral-12B	AWQ3	-0.80
	NF3	-0.71
	GPTQ3	-0.83
Llama3-70B	AWQ3	-0.78
	GPTQ3	-0.81

Table 2: The final-layer residual magnitudes from the full-precision model have a strong negative correlation with the quantization errors of 3-bit methods.

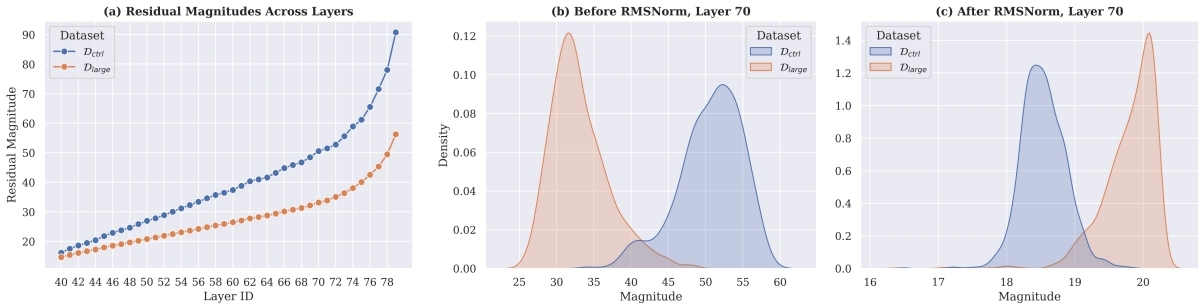

Figure 1: (a) The residual magnitudes of the large-error set $\mathcal{D}_{\text{large}}$ are smaller than those of $\mathcal{D}_{\text{ctrl}}$ across layers. (b) & (c) RMSNorm reverses the relative activation magnitudes. Before RMSNorm (b), the residual magnitudes of $\mathcal{D}_{\text{large}}$ are smaller than $\mathcal{D}_{\text{ctrl}}$, as shown in (a). After applying RMSNorm (c), the activation magnitudes of $\mathcal{D}_{\text{large}}$ become larger. We observe clear reversal effects in the last 10 layers of the full-precision Llama3-70B.

appendix C, the correlations become increasingly negative in upper layers. The strong negative correlations suggest that (1) the full-precision LLMs can indicate an example's future quantization error, and (2) small residual magnitudes in upper layers are associated with large quantization errors.

To further investigate the cause of large errors, we compare the large-error set $\mathcal{D}_{\text{large}}$ with the control set $\mathcal{D}_{\text{ctrl}}$ (§3.1). We compute the average residual magnitudes over examples in $\mathcal{D}_{\text{large}}$ and $\mathcal{D}_{\text{ctrl}}$, respectively, and then compare their average magnitudes across the upper half layers of Llama3-70B. In Fig. 1 (a), we find that for both datasets, the average residual magnitudes increase over layers; however, the $\mathcal{D}_{\text{large}}$ (orange) consistently has smaller magnitudes than $\mathcal{D}_{\text{ctrl}}$ (blue). This implies that having a small residual magnitude in the final layer is the cumulative result of continually having smaller residual magnitudes over the upper layers. Our observation seems to contradict the previous belief that *large* activations are detrimental to weight-only quantization (Lin et al., 2024). We investigate the cause of this discrepancy in the next section.

5.3 From Residuals to Errors

The reversal effect of RMSNorm. Why are smaller residual magnitudes in upper layers associated with larger quantization errors? The RMSNorm applied to the residual states before MLP (Eq. 6), plays a crucial role. We observe a "reversal effect", where the activations with smaller magnitudes often end up with larger magnitudes after applying RMSNorm. Formally, we denote the hidden state after RMSNorm as $h^{(l)} = \text{RMSNorm}^{(l)}(r^{(l)})$. Following Eq. 7, we can compute the magnitude of $h^{(l)}$, denoted as $\text{norm}\left(x; h^{(l)}\right)$. Fig. 1 (b) & (c) present the density of $\text{norm}\left(x; r^{(l)}\right)$ and $\text{norm}\left(x; h^{(l)}\right)$, respectively. Initially, $\mathcal{D}_{\text{large}}$ has smaller $\text{norm}\left(x; r^{(l)}\right)$ than $\mathcal{D}_{\text{ctrl}}$, but after RMSNorm, it has larger $\text{norm}\left(x; h^{(l)}\right)$. We observe the reversal effect in the upper layers of all four LLMs.

Activations amplify errors in weights. Since the hidden state after RMSNorm is the immediate input to the quantized MLP (Eq. 6), we hypothesize that a larger $\text{norm}\left(x; h^{(l)}\right)$ leads to amplification of errors in the quantized weights. Let $h^{(l)}$ and $\tilde{h}^{(l)}$ be the inputs to the l-th layer MLP in the full-precision and quantized models, respectively, and let $o^{(l)}$ and $\tilde{o}^{(l)}$ be the corresponding MLP outputs. We compute the mean squared error between $o^{(l)}$ and $\tilde{o}^{(l)}$ to quantify the layer-wise output error, denoted as $\text{MSE}^{(l)}$. We confirm that the input magnitudes $\text{norm}\left(x; h^{(l)}\right)$ have positive correlations (> 0.6) with $\text{MSE}^{(l)}$ in most upper layers.

Full Hypothesis. Combining our findings above, we establish a full hypothesis: certain examples inherently have smaller residual magnitudes over layers. Due to the reversal effect, they tend to have larger activation magnitudes after RMSNorms. When the model is quantized, large activations will amplify the quantization errors in the weights over

Notation		Definition
$r^{(l)}$	$\mathbb{R}^d$	layer l residual states (Eq. 5)
$h^{(l)}$	$\mathbb{R}^d$	$\text{RMSNorm}^{(l)}(r^{(l)})$
$o^{(l)}$	$\mathbb{R}^d$	$\text{MLP}^{(l)}(h^{(l)})$
γ	$\mathbb{R}^d$	RMSNorm weights (Eq. 1)
$\text{norm}\left(x; r^{(l)}\right)$	$\mathbb{R}$	layer l residual magnitude of x
$\mathbb{S}(\mathcal{D}; \theta; l)$	$\mathbb{R}^N$	layer l residual magnitudes of $\mathcal{D}$

Table 3: Notation summary.

multiple layers and eventually result in large quantization errors. Both the residual states and errors are cumulative over layers, which may explain why the final-layer residual magnitudes $\mathbb{S}(\mathcal{D}; \theta; L)$ have the strongest correlation with the final errors $\mathcal{E}(\mathcal{D}; \tilde{\theta})$.

Why does $\mathcal{D}_{\text{large}}$ have smaller residual magnitudes? We find that examples in the large-error set have fewer/weaker outliers in the residual states $r^{(l)}$. Following Bondarenko et al. (2023); Gong et al. (2024), we use the Kurtosis value to quantify the severity of outliers. We measure the Kurtosis value $\kappa^{(l)}$ of the l-th layer residual states $r^{(l)}$, where $\kappa^{(l)} = \mathbb{E}\left[\left(\frac{r^{(l)}-\mu}{\sigma}\right)^4\right]$. A large $\kappa^{(l)}$ indicates more outliers (Liu et al., 2025a). Fig. 6 in appendix shows that the Kurtosis values of $\mathcal{D}_{\text{large}}$ (orange) are smaller than those of $\mathcal{D}_{\text{ctrl}}$ (blue) across the upper layers of LLMs, suggesting that $\mathcal{D}_{\text{large}}$ has fewer outliers in the residual states. Because outliers dominate the activation magnitudes, having fewer outliers leads to smaller residual magnitudes.

How does RMSNorm reverse the relative magnitudes? We observe that the RMSNorm weights $\gamma \in \mathbb{R}^d$ suppress many outliers in the residual states by assigning smaller weight values to the corresponding dimensions. Specifically, we rank the entries in γ by their absolute values and find that in the last few layers, the largest outliers[4] often have the smallest RMSNorm weights across all four LLMs. We include the statistics and discussion on other outlier dimensions in appendix D. We summarize the mechanism behind the reversal effect: a residual state $r^{(l)}$ with fewer outliers has a smaller magnitude, but RMSNorm normalizes the magnitude and downweights outliers with γ, yielding a hidden state $h^{(l)}$ with a relatively larger magnitude. Since $\mathcal{D}_{\text{large}}$ examples have smaller kurtosis values but larger $\text{norm}\left(x; h^{(l)}\right)$, the large quantization errors are driven by the overall larger activations in $h^{(l)}$.

6 Where Do Errors Arise From?

We study which parts of LLMs lead to large quantization errors with localization techniques: early exiting (nostalgebraist, 2020; Geva et al., 2022) and activation patching (Meng et al., 2022).

[4]The "most-activated" dimension in $r^{(l)}$ that has the highest average absolute activation over the FineWeb dataset.

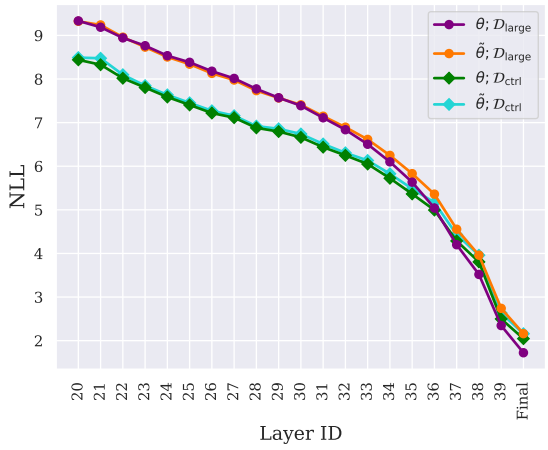

Figure 2: Early exiting across layers shows that the NLLs of $\mathcal{D}_{\text{large}}$ (purple and orange) dramatically decrease in the late layers, suggesting that $\mathcal{D}_{\text{large}}$ heavily relies on the late layers to adjust the output probabilities. Meanwhile, the quantized model (orange) does not show obvious deviation from the full-precision one (purple) until layer 33.

6.1 Early Exiting

nostalgebraist (2020) first introduces logit lens, an early exiting technique that projects hidden representations into the vocabulary space using the LLMs' output embeddings. We use early exiting to measure the degree to which a residual state is ready for decoding. Formally, let $r^{(l)}$ be the residual state at layer l and $\phi(\cdot)$ be the final RMSNorm operation applied before the output embedding matrix U. We define the probability distribution from decoding $r^{(l)}$ as:

$$p^{(l)} = \text{Softmax}\left(U \cdot \phi(r^{(l)})\right)$$

Let $p_\theta^{(l)}$ and $p_{\tilde{\theta}}^{(l)}$ be the early-exiting probabilities from the full-precision and quantized models. We use $p_\theta^{(l)}$ and $p_{\tilde{\theta}}^{(l)}$ to compute the average negative log-likelihood (NLL) over dataset $\mathcal{D}$ at each layer, denoted $\text{nll}_\theta^{(l)}(\mathcal{D})$ and $\text{nll}_{\tilde{\theta}}^{(l)}(\mathcal{D})$, respecitvely. A $\text{nll}_\theta^{(l)}(\mathcal{D})$ value near the final model NLL means that the residual state at layer l contains sufficient information for decoding; otherwise, further layers are needed to refine the output probability.

We investigate two key questions: (1) Can we observe patterns from the early exiting dynamics that distinguish $\mathcal{D}_{\text{large}}$ and $\mathcal{D}_{\text{ctrl}}$? (2) At which layer do $\text{nll}_\theta^{(l)}(\mathcal{D})$ and $\text{nll}_{\tilde{\theta}}^{(l)}(\mathcal{D})$ begin to diverge? To answer these questions, we perform four runs of early exiting, one for each combination of model

Model	Method	16-bit	3-bit	Patch h_{down}	Patch h_{gate}	Patch h_{up}	Patch h_{attn}
Qwen2.5-7B	AWQ3	9.42	12.26 (+2.84)	9.73 (+0.31)	10.20 (+0.78)	12.64 (+3.22)	11.12 (+1.70)
	NF3		12.53 (+3.11)	9.66 (+0.24)	10.61 (+1.19)	12.97 (+3.55)	11.35 (+1.93)
Llama3-8B	AWQ3	5.62	10.68 (+5.06)	5.95 (+0.33)	6.42 (+0.80)	9.97 (+4.35)	9.27 (+3.65)
	NF3		11.02 (+5.40)	5.92 (+0.30)	6.43 (+0.81)	10.48 (+4.86)	9.49 (+3.87)
Mistral-12B	AWQ3	5.60	8.67 (+3.07)	6.05 (+0.45)	6.13 (+0.53)	12.49 (+6.89)	7.61 (+2.01)
	NF3		10.14 (+4.54)	5.92 (+0.32)	7.12 (+1.52)	13.75 (+8.15)	8.71 (+3.11)
Llama3-70B	AWQ3	2.66	5.25 (+2.59)	2.73 (+0.07)	2.94 (+0.28)	3.24 (+0.58)	4.73 (+2.07)

Table 4: The perplexity (↓) on $\mathcal{D}_{\text{large}}$ before/after patching activations from 16-bit full-precision models into 3-bit models, quantized with AWQ3 and NF3 methods, respectively. Both patching the entire MLP outputs (h_{down}) and patching only the MLP gate outputs (h_{gate}) can substantially close the gap between the 3-bit and 16-bit models.

type (full-precision and quantized) and dataset split ($\mathcal{D}_{\text{large}}$ and $\mathcal{D}_{\text{ctrl}}$). We then compute $\text{nll}_{\theta}^{(l)}(\mathcal{D}_{\text{large}})$, $\text{nll}_{\tilde{\theta}}^{(l)}(\mathcal{D}_{\text{large}})$, $\text{nll}_{\theta}^{(l)}(\mathcal{D}_{\text{ctrl}})$, and $\text{nll}_{\tilde{\theta}}^{(l)}(\mathcal{D}_{\text{ctrl}})$ across the upper half layers.

Fig. 2 compares how the four NLL values evolve across the upper layers of Mistral-12B, where the quantized model $\tilde{\theta}$ is produced by AWQ3. We observe a clear distinction between $\mathcal{D}_{\text{large}}$ and $\mathcal{D}_{\text{ctrl}}$ on the full-precision model: $\mathcal{D}_{\text{large}}$ (purple) has higher NLLs than $\mathcal{D}_{\text{ctrl}}$ (green) until layer 36, around which its NLLs begin to decrease sharply. This observation supports our claim in §5 that the properties of the full-precision model reflect the future quantization errors. Moreover, the sharp NLL decrease in later layers suggests that $\mathcal{D}_{\text{large}}$ heavily relies on these layers to make accurate predictions. This reliance requires the quantized model to keep the residual states precise through to the later layers, which is a challenging task because quantization errors propagate over layers. Otherwise, the model cannot decode the noisy representations, resulting in large quantization errors. By comparison, $\mathcal{D}_{\text{ctrl}}$ relies less on the later layers, and the quantized model (cyan) shows less deviation from the full-precision one (green).

An alternative hypothesis is that the precision of weights in the later layers is crucial to $\mathcal{D}_{\text{large}}$. To test this hypothesis, we keep all the weights above layer 32 in full-precision, because Fig. 2 shows that on $\mathcal{D}_{\text{large}}$, the quantized model (orange) starts to diverge from the full-precision model (purple) at layer 33. This experiment restores 7 layers (18%) from 3 bits to 16 bits. However, the perplexity on $\mathcal{D}_{\text{large}}$ only slightly decreases from 8.67 to 8.05, versus 5.60 for the full-precision model. We show similar results on other LLMs in appendix E.

In summary, our early exiting experiment suggests that various methods struggle with $\mathcal{D}_{\text{large}}$ because it requires the quantized models to maintain precise residual states up to the late layers. We also find that leaving late layers unquantized has little effect on NLLs, implying that the cumulative error in residual states, not the precision of upper-layer weights, is the primary cause for large errors.

6.2 Cross-Model Activation Patching

Prakash et al. (2024) introduce cross-model activation-patching (CMAP) to identify how fine-tuned models improve over the pretrained ones. We adapt CMAP to track where the quantized models differ from the full-precision ones, aiming to identify the sources of quantization errors. For a given example from $\mathcal{D}_{\text{large}}$, we first run a forward pass on the full-precision model and cache its activations. We then run the same example through the quantized model, but replace the activations of specific modules with those from the full-precision model. We patch the outputs of MHA, denoted h_{attn}, and the outputs of different modules in MLP. Specifically, all LLMs we study implement MLP with Gated Linear Units (Dauphin et al., 2017):

$$
\begin{aligned}
h_{\text{gate}} &= \sigma(W_{\text{gate}}\, h), \\
h_{\text{up}} &= W_{\text{up}}\, h, \\
h_{\text{down}} &= W_{\text{down}}\, (h_{\text{gate}} \odot h_{\text{up}}),
\end{aligned}
$$

where h is the post-RMSNorm input to MLP, W_{gate}, W_{up}, and W_{down} are the MLP weights, h_{down} is the output of MLP, and $\sigma(\cdot)$ is the activation function. To trace quantization errors, we perform four runs of CMAP, corresponding to patching h_{gate}, h_{up}, h_{down}, and h_{attn}, in the upper half layers of the models. Patching h_{down} is a much stronger intervention than patching either h_{up} or h_{down}, as it overrides the final outputs of MLPs.

We apply AWQ3 and NF3 methods and report the perplexity on $\mathcal{D}_{\text{large}}$ in Table 4. First, we find

that patching the outputs of MLP h_{down} greatly narrows the perplexity gaps between the 3-bit and 16-bit full-precision models, bringing the gaps below 0.5 across all LLMs. In contrast, patching the outputs of MHA h_{attn} only leads to marginal effects, suggesting that the quantization errors from MLPs are dominating the results. We further find that inside MLPs, patching h_{up} alone is ineffective, but patching h_{gate} results in substantial perplexity reduction, approaching the effect of patching the entire MLP outputs h_{down}. These results highlight the importance of having precise outputs from MLP gates, which control the information passed on to subsequent layers.

We conduct two additional experiments to understand the source of quantization errors in MLP gates: (1) only patching the inputs to W_{gate}, and (2) restoring all W_{gate} weights to 16-bit precision. We find that patching the gates' inputs (1) performs nearly as well as patching the outputs, h_{gate}, while restoring the weights (2) is ineffective. For instance, on Llama-3-8B quantized with NF3, (1) achieves a perplexity of 6.53 compared to 10.45 for (2). These results further support our conclusion in §6.1 that noisy activations are the primary source of large quantization errors.

7 What Data Cause Large Errors?

Understanding what kinds of data lead to large quantization errors is crucial for future improvement. This section studies whether examples in $\mathcal{D}_{\text{large}}$ cluster in certain domains or are unusual, long-tail data.

7.1 Categorizing Examples in $\mathcal{D}_{\text{large}}$

To better understand the distribution of $\mathcal{D}_{\text{large}}$, we apply the topic classifier from Wettig et al. (2025), which categorizes web pre-training data into 24 topics.[5] We find that the $\mathcal{D}_{\text{large}}$ sets of both Mistral-12B and Llama3-70B cover all the 24 topics, and we plot the topic distributions in Fig. 3. For both models, `Entertainment`, `Finance & Business`, `Politics`, and `Health` are among the top five topics, although none account for more than 12% of the dataset. We also manually inspect $\mathcal{D}_{\text{large}}$ examples and do not find them unusual.

7.2 Quantization Errors vs. Long-Tail Data

Ogueji et al. (2022); Marchisio et al. (2024) show that model compression has a disparate effect for long-tail data; thus, we explore the relationship between quantization errors and data long-tailness.

[5] https://huggingface.co/WebOrganizer/TopicClassifier-NoURL

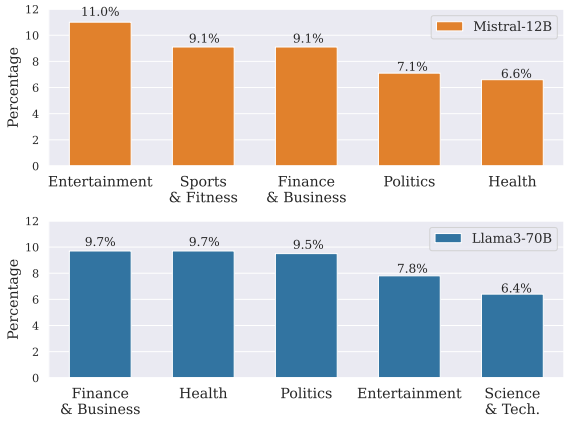

Figure 3: Top five topics of $\mathcal{D}_{\text{large}}$ for Mistral-12B (up; orange) and Llama3-70B (down; blue), respectively.

FineWeb. We use the data likelihood assigned by a strong language model, Llama3-70B, to estimate the long-tail characteristics of the examples. We call the FineWeb examples that have small log-likelihoods (large NLLs) on the full-precision Llama3-70B as long-tail examples. Fig. 4 shows that long-tail examples (x-axis ≥ 4) have small quantization errors under both AWQ3 (avg. 0.14) and GPTQ3 (avg. 0.30) methods. On the other hand, examples that suffer from large errors have widespread NLLs before quantization, and many "head data" (small NLLs; x-axis ≤ 1) have large quantization errors (avg. 0.67 for AWQ3 and 1.00 for GPTQ3). We observe the same findings on the other LLMs (see Fig. 16 in appendix).

PopQA. We investigate how quantization affects LLMs' factual knowledge on entities with different popularity levels. We partition the PopQA dataset into buckets based on the examples' $\log_{10}$(popularity) and calculate the accuracies within each bucket. Fig. 5 presents the accuracy degradation after quantization under different buckets, where we apply NF3 (blue) and AWQ3 (orange) to quantize Mistral-12B into 3 bits. We find that when the examples have the lowest and highest popularities (the leftmost and rightmost buckets, respectively), the 3-bit models show the lowest degradation. In contrast, examples in the middle buckets (log popularity between 2.8 and 4.8) are most affected by quantization, with accuracies dropped by $> 10\%$. Note that only 6% of examples have log popularity greater than 4.8, meaning that quantization can severely degrade

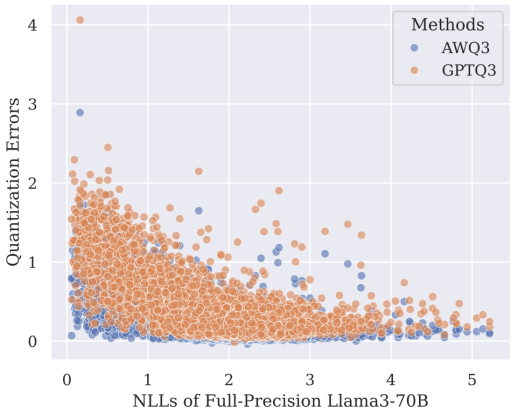

Figure 4: Long-tail samples with large NLLs on the full-precision Llama3-70B have small quantization errors. Each dot represents a FineWeb example. We quantize Llama3-70B with AWQ3 and GPTQ3 methods and compute their quantization errors with Eq. 4.

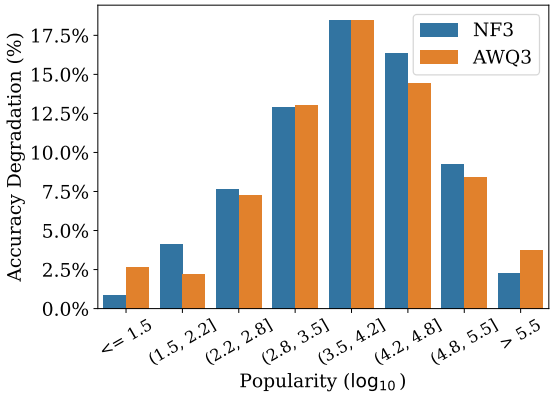

Figure 5: The accuracy degradation of PopQA under different levels of popularity after 3-bit quantization. Samples with log popularity between 2.8 and 4.8 show the greatest degradation.

LLMs' knowledge on entities that have intermediate to high popularity. We include results from other LLMs in Fig. 17, showing that low popularity examples usually have milder degradation.

Summary. We find that long-tail examples are relatively less affected by quantization on both FineWeb and PopQA. Meanwhile, well-represented data are not immune to large errors.

8 Discussion

We thoroughly study why certain examples are disproportionately affected by 3–4 bit, weight-only quantization methods. We reveal the distinct nature of large-error examples before quantization, showing that these examples have smaller residual magnitudes and rely on upper layers to refine output probabilities. Our patching and weight recovery experiments further highlight the importance of having precise activations at MLP gates, and reject naive mixed-precision approaches. Our kurtosis value analysis finds that large-error examples actually have fewer outliers in residual states across upper layers. We further discover the reversal effects of RMSNorm, showing that large quantization errors may stem from overall larger activations across dimensions, rather than from a few outlier dimensions. Finally, we explore the relationship between quantization errors and long-tail data, showing that long-tail examples tend to have less degradation after quantization than other examples.

Overall, our work sheds light on why and where large quantization errors occur. Our findings may inspire quantization error prediction frameworks and encourage future post-training methods or quantization-friendly architectures that specifically address MLP gates and RMSNorm; for instance, by avoiding the Gated Linear Units or redesigning the LayerNorm (Zhu et al., 2025).

9 Limitations

We conduct our main experiments on FineWeb and use NLL as our evaluation metric, which does not consider downstream tasks. We believe this is a reasonable choice because Jin et al. (2024) show that perplexity (exponentiated average NLL) is a reliable performance indicator for quantized LLMs on various evaluation benchmarks. We invite future work beyond intrinsic analysis to extrinsic, downstream tasks.

We do not cover sub-3-bit settings because we observe that the methods in our study suffer from extremely large errors when quantized below 3 bits (with perplexity greater than 10^3 and near-random accuracy on commonsense QA tasks), making it hard to pinpoint the cause of errors. Because prior work also marks 3 bits as a turning point in quantization (Dettmers and Zettlemoyer, 2023; Liu et al., 2025b), our study of quantization errors under the 3–4 bit regime could be valuable to future efforts to improve quantization methods to lower bits.

Acknowledgements

We thank Yuqing Yang for helpful discussions, Ming Zhong for guidance on table coloring, and the anonymous reviewers for their valuable feedback. This work was supported in part by the National Science Foundation under Grant No. IIS-2403436.

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Orevaoghene Ahia, Julia Kreutzer, and Sara Hooker. 2021. The low-resource double bind: An empirical study of pruning for low-resource machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3316–3333, Punta Cana, Dominican Republic. Association for Computational Linguistics.

AI@Meta. 2024. Llama 3 model card.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. 2024. Quarot: Outlier-free 4-bit inference in rotated llms. *Advances in Neural Information Processing Systems*, 37:100213–100240.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q. Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2023. Llemma: An open language model for mathematics. *Preprint*, arXiv:2310.10631.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Yoshua Bengio, Nicholas Léonard, and Aaron Courville. 2013. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. 2023. Quantizable transformers: Removing outliers by helping attention heads do nothing. *Advances in Neural Information Processing Systems*, 36:75067–75096.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M De Sa. 2023. Quip: 2-bit quantization of large language models with guarantees. *Advances in Neural Information Processing Systems*, 36:4396–4429.

Jiale Chen, Torsten Hoefler, and Dan Alistarh. 2025. The geometry of llm quantization: Gptq as babai's nearest plane algorithm. *arXiv preprint arXiv:2507.18553*.

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. 2024. Efficientqat: Efficient quantization-aware training for large language models. *arXiv preprint arXiv:2407.11062*.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509*.

Together Computer. 2023. Redpajama: An open source recipe to reproduce llama training dataset.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, pages 30318–30332. Curran Associates, Inc.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. QLoRA: Efficient finetuning of quantized LLMs. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Tim Dettmers and Luke Zettlemoyer. 2023. The case for 4-bit precision: k-bit inference scaling laws. In *International Conference on Machine Learning*, pages 7750–7774. PMLR.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2023. GPTQ: Accurate post-training quantization for generative pre-trained transformers. In *The Eleventh International Conference on Learning Representations*.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, and 1 others. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Team Gemma, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, and 1 others. 2024. Gemma: Open models based on gemini research and technology. *arXiv preprint arXiv:2403.08295*.

Mor Geva, Avi Caciularu, Kevin Wang, and Yoav Goldberg. 2022. Transformer feed-forward layers build predictions by promoting concepts in the vocabulary space. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 30–45, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ruihao Gong, Yang Yong, Shiqiao Gu, Yushi Huang, Chengtao Lv, Yunchen Zhang, Xianglong Liu, and Dacheng Tao. 2024. Llmc: Benchmarking large language model quantization with a versatile compression toolkit. *arXiv preprint arXiv:2405.06001*.

Han Guo, William Brandon, Radostin Cholakov, Jonathan Ragan-Kelley, Eric P. Xing, and Yoon Kim. 2024. Fast matrix multiplications for lookup table-quantized LLMs. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12419–12433, Miami, Florida, USA. Association for Computational Linguistics.

Paul Jaccard. 1912. The distribution of the flora in the alpine zone. 1. *New phytologist*, 11(2):37–50.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

R Jin, J Du, W Huang, W Liu, J Luan, B Wang, and D Xiong. 2024. A comprehensive evaluation of quantization strategies for large language models. arxiv.

Olga Kovaleva, Saurabh Kulshreshtha, Anna Rogers, and Anna Rumshisky. 2021. BERT busters: Outlier dimensions that disrupt transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3392–3405, Online. Association for Computational Linguistics.

Tanishq Kumar, Zachary Ankner, Benjamin Frederick Spector, Blake Bordelon, Niklas Muennighoff, Mansheej Paul, Cengiz Pehlevan, Christopher Re, and Aditi Raghunathan. Scaling laws for precision. In *The Thirteenth International Conference on Learning Representations*.

Eldar Kurtic, Alexandre Marques, Shubhra Pandit, Mark Kurtz, and Dan Alistarh. 2024. " give me bf16 or give me death"? accuracy-performance trade-offs in llm quantization. *arXiv preprint arXiv:2411.02355*.

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

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. 2024. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. In *Proceedings of Machine Learning and Systems*, volume 6, pages 87–100.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2024. LLM-QAT: Data-free quantization aware training for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 467–484, Bangkok, Thailand. Association for Computational Linguistics.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. 2025a. Spinquant: LLM quantization with learned rotations. In *The Thirteenth International Conference on Learning Representations*.

Zechun Liu, Changsheng Zhao, Hanxian Huang, Sijia Chen, Jing Zhang, Jiawei Zhao, Scott Roy, Lisa Jin, Yunyang Xiong, Yangyang Shi, and 1 others. 2025b. Paretoq: Scaling laws in extremely low-bit llm quantization. *arXiv preprint arXiv:2502.02631*.

Haozheng Luo, Chenghao Qiu, Maojiang Su, Zhihan Zhou, Zoe Mehta, Guo Ye, Jerry Yao-Chieh Hu, and Han Liu. 2025. Fast and low-cost genomic foundation models via outlier removal. In *Forty-second International Conference on Machine Learning*.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. 2023. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9802–9822, Toronto, Canada. Association for Computational Linguistics.

Kelly Marchisio, Saurabh Dash, Hongyu Chen, Dennis Aumiller, Ahmet Üstün, Sara Hooker, and Sebastian Ruder. 2024. How does quantization affect multilingual LLMs? In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15928–15947, Miami, Florida, USA. Association for Computational Linguistics.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372.

nostalgebraist. 2020. interpreting GPT: the logit lens.

Kelechi Ogueji, Orevaoghene Ahia, Gbemileke Onilude, Sebastian Gehrmann, Sara Hooker, and Julia Kreutzer. 2022. Intriguing properties of compression on multilingual models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9092–9110, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Team OLMo, Pete Walsh, Luca Soldaini, Dirk Groeneveld, Kyle Lo, Shane Arora, Akshita Bhagia, Yuling Gu, Shengyi Huang, Matt Jordan, Nathan Lambert, Dustin Schwenk, Oyvind Tafjord, Taira Anderson, David Atkinson, Faeze Brahman, Christopher Clark, Pradeep Dasigi, Nouha Dziri, and 21 others. 2024. 2 olmo 2 furious.

Gunho Park, Baeseong park, Minsub Kim, Sungjae Lee, Jeonghoon Kim, Beomseok Kwon, Se Jung Kwon, Byeongwook Kim, Youngjoo Lee, and Dongsoo Lee. 2024. LUT-GEMM: Quantized matrix multiplication based on LUTs for efficient inference in large-scale generative language models. In *The Twelfth International Conference on Learning Representations*.

Keiran Paster, Marco Dos Santos, Zhangir Azerbayev, and Jimmy Ba. 2023. Openwebmath: An open dataset of high-quality mathematical web text. *Preprint*, arXiv:2310.06786.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. 2024. The fineweb datasets: Decanting the web for the finest text data at scale. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Nikhil Prakash, Tamar Rott Shaham, Tal Haklay, Yonatan Belinkov, and David Bau. 2024. Fine-tuning enhances existing mechanisms: A case study on entity tracking. In *Proceedings of the 2024 International Conference on Learning Representations*. ArXiv:2402.14811.

Team Qwen. 2024. Qwen2.5: A party of foundation models.

Team Qwen. 2025. Qwen3.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67.

Noam Shazeer. 2020. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Ying Sheng, Lianmin Zheng, Binhang Yuan, Zhuohan Li, Max Ryabinin, Beidi Chen, Percy Liang, Christopher Re, Ion Stoica, and Ce Zhang. 2023. FlexGen: High-throughput generative inference of large language models with a single GPU. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 31094–31116. PMLR.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and 1 others. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. 2023. 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*, pages 1648–1665.

Xiuying Wei, Yunchen Zhang, Xiangguo Zhang, Ruihao Gong, Shanghang Zhang, Qi Zhang, Fengwei Yu, and Xianglong Liu. 2022. Outlier suppression: Pushing the limit of low-bit transformer language models. *Advances in Neural Information Processing Systems*, 35:17402–17414.

Alexander Wettig, Kyle Lo, Sewon Min, Hannaneh Hajishirzi, Danqi Chen, and Luca Soldaini. 2025. Organize the web: Constructing domains enhances pretraining data curation. In *Forty-second International Conference on Machine Learning*.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pages 38087–38099. PMLR.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. 2020. On layer normalization in the transformer architecture. In *International conference on machine learning*, pages 10524–10533. PMLR.

Biao Zhang and Rico Sennrich. 2019. Root mean square layer normalization. *Advances in Neural Information Processing Systems*, 32.

Jiachen Zhu, Xinlei Chen, Kaiming He, Yann LeCun, and Zhuang Liu. 2025. Transformers without normalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

A Implementation Details

We aim to answer whether different quantization methods make similar errors; thus, we follow the original implementations and do not unify their quantization configurations.

GPTQ. We use the `GPTQModel`[6] package for implementation. The calibration set consists of samples from English C4.[7] The quantization group size is 64 for 3-bit models and 128 for 4-bit models. We do not include the results of GPTQ3 with Llama-3-8B because we observe severe perplexity degradation with the latest `GPTQModel` package.

AWQ. We follow the original implementation[8] and use the released model checkpoints when available. The calibration set consists of samples from the Pile (Gao et al., 2020). The quantization group size is 128 for both 3- and 4-bit models. Note that because AWQ applies scaling transformation, in our patching experiments §6.2, we also apply the same scaling factors on the *full-precision* models, which does not change their outputs but aligns their activations with the AWQ-quantized models.

NF. We use the official packages: `bitsandbytes`[9] for 4-bit models and `flute`[10] for 3-bit model. The quantization group size is 64 for both 3- and 4-bit models. We do not have the results of NF3 on Llama-3-70B due to kernel incompatibility.

EQAT. EQAT train on thousands of samples from RedPajama (Computer, 2023). We use the officially released 3-bit model checkpoints,[11] which only cover the Llama3 models. The quantization group size is 128.

Perplexity. We report the perplexity of different 3- and 4-bit quantization methods in Table 5. We follow the standard implementation of WikiText perplexity that sets the sequence length to 2048. The sequence length of FineWeb is 512.

Data. We use FineWeb for our main experiments because it is a diverse, cleaned dataset, and none of the quantization methods considered above sample calibration data from it, making FineWeb an unbiased corpus for our study. We also explore more technical domains, Arxiv and OpenWebMath (Azerbayev et al., 2023; Paster et al., 2023). Our findings that the quantization errors of different methods are strongly correlated (§4) also hold on these domains. Specifically, the average correlation across 8 pairs of methods on Llama3-70B is 0.69 on ArXiv and 0.89 on OpenWebMath.

Model	Method	Fineweb	Wiki
Qwen2.5-7B	16 bits	11.06	6.85
	AWQ4	11.35	7.09
	NF4	11.37	7.10
	GPTQ4	11.33	7.16
	AWQ3	12.77	8.20
	NF3	13.47	8.51
	GPTQ3	12.26	8.11
Llama3-8B	16 bits	9.57	6.14
	AWQ4	10.13	6.56
	NF4	10.11	6.56
	GPTQ4	10.12	6.86
	AWQ3	12.13	8.19
	NF3	12.45	8.44
	EQAT3	10.77	7.10
Mistral-12B	16 bits	8.67	5.75
	AWQ4	9.09	6.01
	NF4	9.13	6.06
	GPTQ4	9.02	6.04
	AWQ3	10.32	7.03
	NF3	12.48	8.75
	GPTQ3	10.44	7.20
Llama3-70B	16 bits	7.15	2.86
	AWQ4	7.42	3.27
	NF4	7.88	3.22
	GPTQ4	7.52	3.58
	GPTQ3	9.64	6.51
	AWQ3	8.40	4.74

Table 5: The perplexity of different quantization methods on the 10k FineWeb dataset and WikiText. Our FineWeb examples have shorter sequence lengths than WikiText (512 vs. 2048), leading to higher perplexity.

Definition of Error. We also explore an alternative definition of quantization errors based on KL divergence. Let p_θ, the probability distribution of the full-precision model, be the true distribution. We can define errors as the KL divergence between the quantized and full-precision model, $\mathrm{KL}\left(p_\theta \,\|\, p_{\tilde{\theta}}\right)$. The results of these two definitions are highly correlated ($\rho \geq 0.97$). Thus, we adopt Eq. 4 for its lower computation costs.

B The Construction of the $\mathcal{D}_{\text{large}}$

To ensure the diversity of $\mathcal{D}_{\text{large}}$ examples, we apply aggressive data deduplication based on Jaccard similarity (Jaccard, 1912), filtering out ex-

[6] https://github.com/ModelCloud/GPTQModel
[7] en/c4-train.00001-of-01024.json.gz
[8] https://github.com/mit-han-lab/llm-awq
[9] https://pypi.org/project/bitsandbytes
[10] https://github.com/HanGuo97/flute
[11] https://github.com/OpenGVLab/EfficientQAT

amples with 5-gram Jaccard similarity > 0.1. We create a separate $\mathcal{D}_{\text{large}}$ for each quantization method and observe similar findings across them. Besides, the $\mathcal{D}_{\text{large}}$ sets of different methods under the same LLMs often have overlapping examples, with the average Jaccard similarity $\frac{|A \cap B|}{|A \cup B|} = 0.30, 0.49, 0.66, 0.66$ on Qwen2.5-7B, Meta-Llama-3-8B, Mistral-12B, and Meta-Llama-3-70B, respectively. Therefore, in the main content, we always use the one derived from AWQ3 as $\mathcal{D}_{\text{large}}$ for simplicity.

C Correlations Between Residual Magnitudes and Quantization Errors

We present the correlations between the quantization errors $\mathcal{E}(\mathcal{D}; \tilde{\theta})$ and the residual magnitudes $\mathbb{S}(\mathcal{D}; \theta; l)$ across layers of Qwen2.5-7B, Llama3-8B, Mistral-12B, and Llama3-70B in Fig. 8, 9, 10, 11, respectively. We find that different models and methods all show that the correlations become increasingly negative in deeper layers.

D RMSNorm and Outliers

First, we select the serverest outlier dimensions in $r^{(l)}$, namely, the most-activated dimensions in $r^{(l)}$ that have the highest average absolute activation over the 10k FineWeb dataset. Next, we rank the entries in RMSNorm weights γ by their absolute values. We compare the median of γ with those entries corresponding to the outlier dimensions. Table 8 shows that in the upper half layers of Llama3-70B, all outlier dimensions have much smaller weights than the median. On the other hand, in Mistral-12B, the most severe outliers have small weights (rank 1 or 2) across layers, but other outlier dimensions sometimes have much larger weights than the median. These results show that RMSNorm weights suppress the largest outlier dimensions but sometimes also promote other outlier dimensions. In Fig. 6 and Fig. 7, we show the kurtosis values before and after RMSNorm, respectively. Comparing the scale of the kurtosis values (y-axis) of the two figures, we find that the scale after RMSNorm is much smaller, implying that RMSNorm in the upper layers suppresses outliers overall.

Similar to Wei et al. (2022), we find that the LayerNorm parameters modulate activation outliers; however, we observe mostly opposite effects, likely due to the different architectural designs (pre-LN vs. post-LN Xiong et al., 2020).

Model	(Method1, Method2)	$\rho(\mathcal{E}_1, \mathcal{E}_2)$
Qwen2.5-7B	(AWQ4 , NF4)	0.52
	(AWQ4 , GPTQ4)	0.66
	(AWQ4 , NF3)	0.61
	(AWQ4 , GPTQ3)	0.65
	(AWQ3 , NF3)	0.78
	(AWQ3 , GPTQ3)	0.84
	(NF4 , GPTQ4)	0.57
	(NF4 , AWQ3)	0.57
	(NF4 , GPTQ3)	0.55
	(NF3 , GPTQ3)	0.75
	(GPTQ4, AWQ3)	0.75
	(GPTQ4, NF3)	0.70
Llama3-8B	(AWQ4 , NF4)	0.95
	(AWQ4 , GPTQ4)	0.96
	(AWQ4 , NF3)	0.88
	(AWQ4 , EQAT3)	0.92
	(AWQ3 , NF3)	0.98
	(AWQ3 , EQAT3)	0.97
	(NF4 , GPTQ4)	0.95
	(NF4 , AWQ3)	0.87
	(NF4 , EQAT3)	0.90
	(NF3 , EQAT3)	0.96
	(GPTQ4, AWQ3)	0.90
	(GPTQ4, NF3)	0.90
	(GPTQ4, EQAT3)	0.93

Table 6: The quantization errors of different methods are strongly correlated on the FineWeb. The numbers in the method names denote 3 or 4-bit precision.

Variant Pairs	ρ
(damp-0.005-sort, damp-0.01-sort)	0.95
(damp-0.005-sort, damp-0.02-sort)	0.95
(damp-0.01 -sort, damp-0.02-sort)	0.95
(damp-0.005-sort, damp-0.01-no_sort)	0.92
(damp-0.01 -sort, damp-0.01-no_sort)	0.92
(damp-0.02 -sort, damp-0.01-no_sort)	0.92

Table 7: Quantization error correlations of 3-bit Qwen2.5-7B GPTQ variants with different damping coefficients and activation-sorting settings. All pairs are strongly correlated.

E More Early Exiting Results

We apply early exiting on $z^{(l)}$ and $r^{(l)}$, the hidden states after the residual operations (see Eq. 6), across layers. Figure 12, 13, 14, and 15 shows the results on Qwen2.5-7B, Llama3-8B, Mistral-12B, and Llama3-70B, respectively, where we use AWQ3 to produce the quantized models. All the LLMs show that the NLLs of $\mathcal{D}_{\text{large}}$ (purple) decrease dramatically in the later layers, suggesting that $\mathcal{D}_{\text{large}}$ relies more on the late layers to decode the representations than $\mathcal{D}_{\text{ctrl}}$ (green).

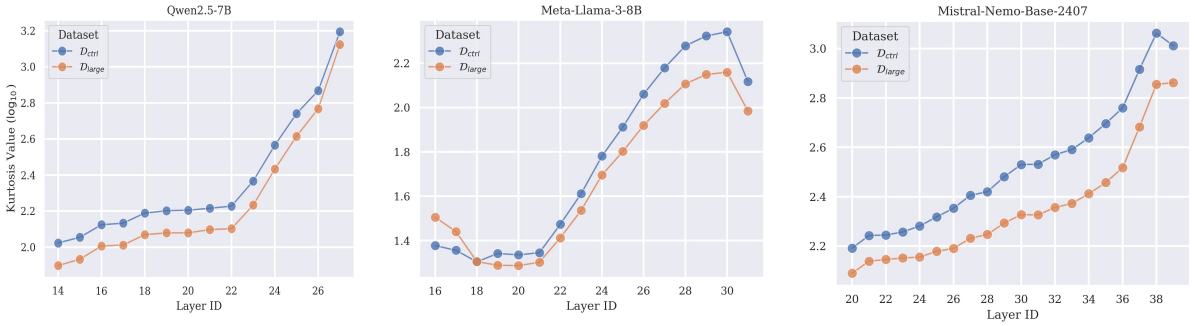

Figure 6: Kurtosis values of the residual states $r^{(l)}$ from different full-precision LLMs (log scale). The large-error set $\mathcal{D}_{\text{large}}$ shows lower kurtosis values than the control set $\mathcal{D}_{\text{ctrl}}$ across layers, indicating that $\mathcal{D}_{\text{large}}$ contains fewer or less extreme outliers in the residual states.

Figure 7: Kurtosis values of the post-RMSNorm hidden states $h^{(l)}$ from different full-precision LLMs (log scale). The large-error set $\mathcal{D}_{\text{large}}$ shows lower kurtosis values than the control set $\mathcal{D}_{\text{ctrl}}$, except for the last layer of Llama3-8B, indicating that overall $\mathcal{D}_{\text{large}}$ contains fewer or less extreme outliers in the hidden states after RMSNorm. In addition, compared with Fig. 6 (before RMSNorm), the scale of the kurtosis values in this figure is much smaller (y-axis), implying that RMSNorm in the upper layers tends to suppress outliers.

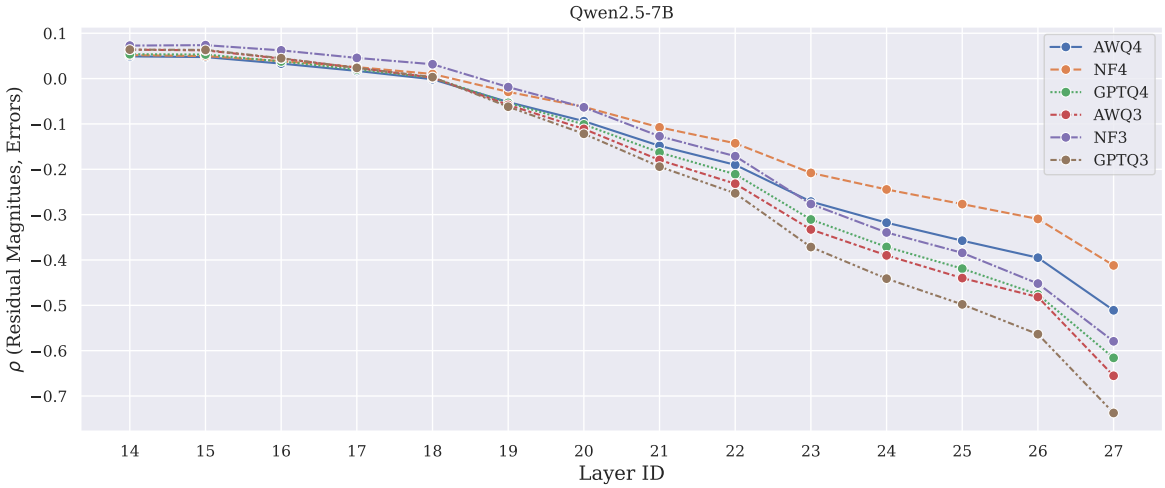

Figure 8: Correlations between the quantization errors $\mathcal{E}(\mathcal{D};\tilde{\theta})$ and the residual magnitudes $\mathbb{S}(\mathcal{D};\theta;l)$ across the upper half layers of Qwen2.5-7B. All methods show negative correlations in the last few layers.

F Full Results on PopQA

Fig. 17 and 18 show the accuracy degradation and absolute accuracy values of different models, respectively. Fig. 19 shows the popularity histogram.

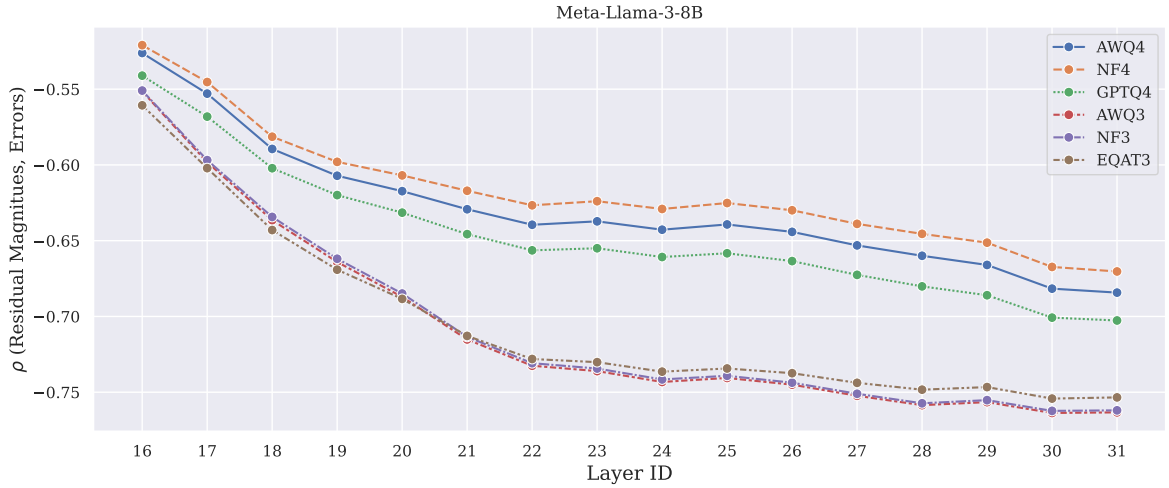

Figure 9: Correlations between the quantization errors $\mathcal{E}(\mathcal{D};\tilde{\theta})$ and the residual magnitudes $\mathbb{S}(\mathcal{D};\theta;l)$ across the upper half layers of Llama3-8B. All methods show negative correlations in the last few layers.

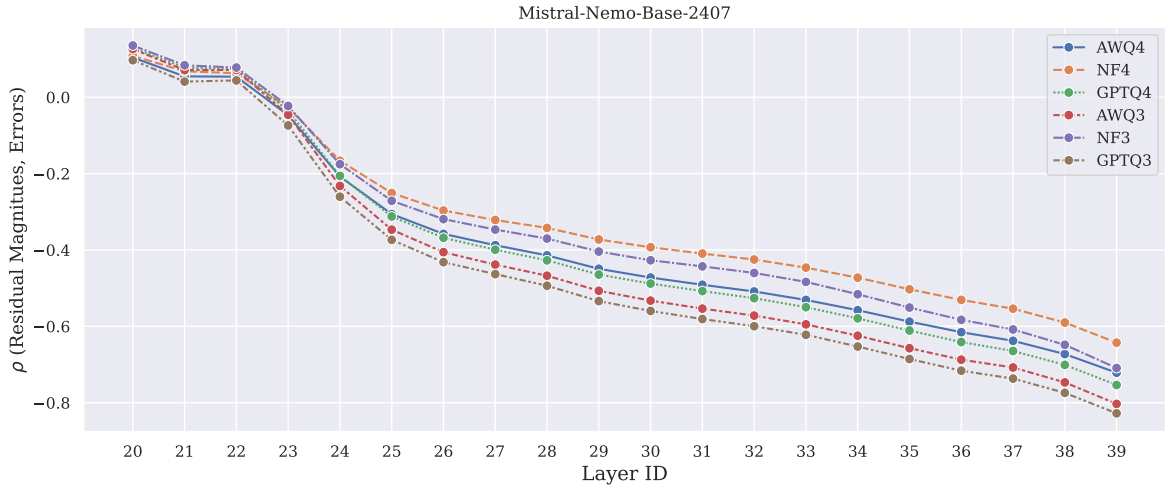

Figure 10: Correlations between the quantization errors $\mathcal{E}(\mathcal{D};\tilde{\theta})$ and the residual magnitudes $\mathbb{S}(\mathcal{D};\theta;l)$ across the upper half layers of Mistral-12B. All methods show negative correlations in the last few layers.

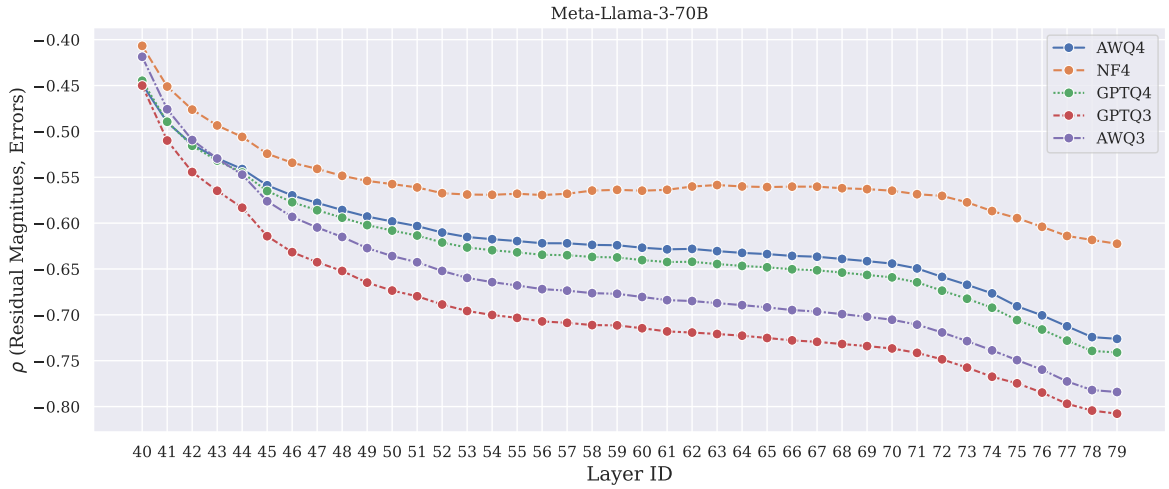

Figure 11: Correlations between the quantization errors $\mathcal{E}(\mathcal{D};\tilde{\theta})$ and the residual magnitudes $\mathbb{S}(\mathcal{D};\theta;l)$ across the upper half layers of Llama3-70B. All methods show negative correlations in the last few layers.

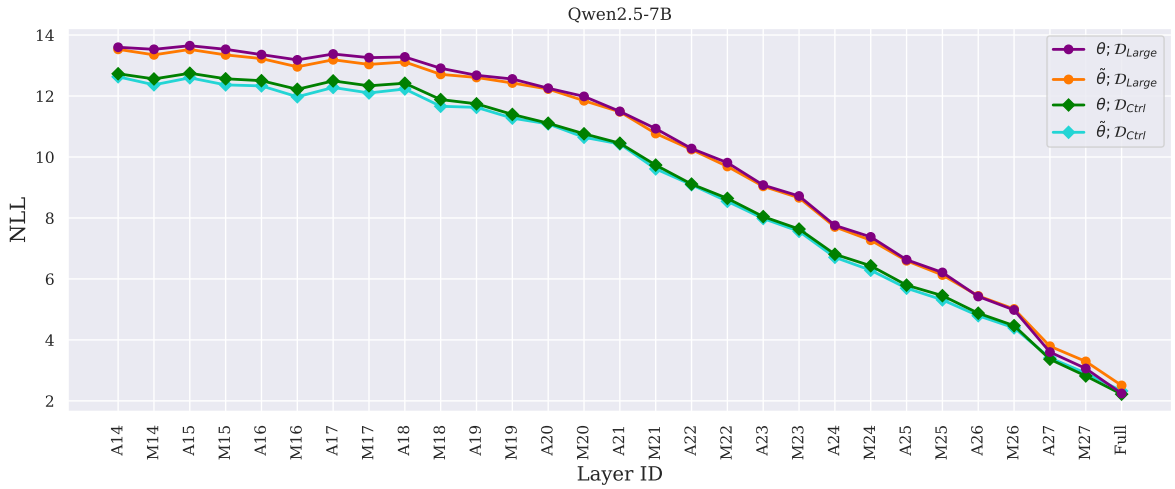

Figure 12: Early Exiting results on Llama3-8B. We decode the $z^{(l)}$ (A#) and $r^{(l)}$ (M#) across layers.

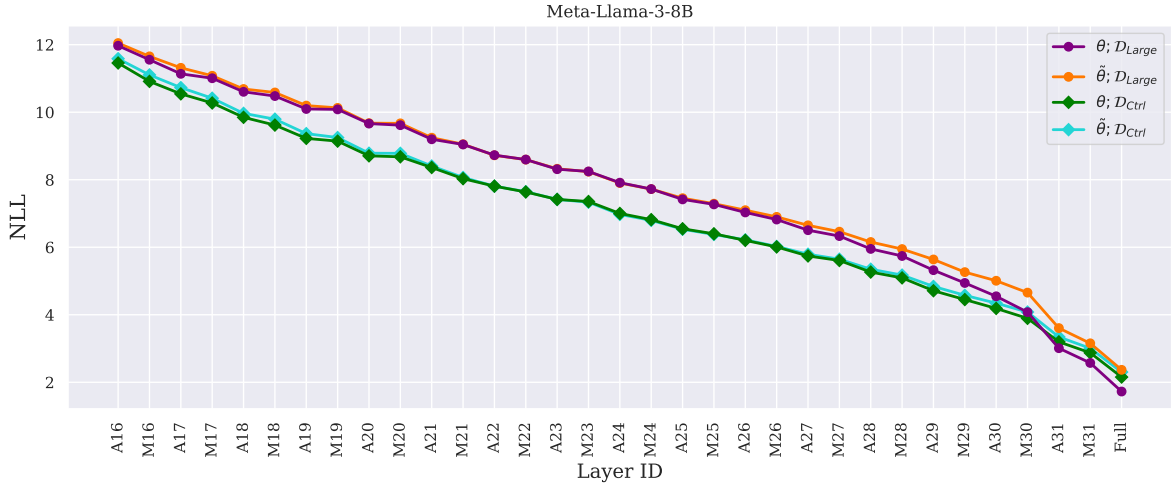

Figure 13: Early Exiting results on Llama3-8B. We decode the inputs to $z^{(l)}$ (A#) and $r^{(l)}$ (M#) across layers.

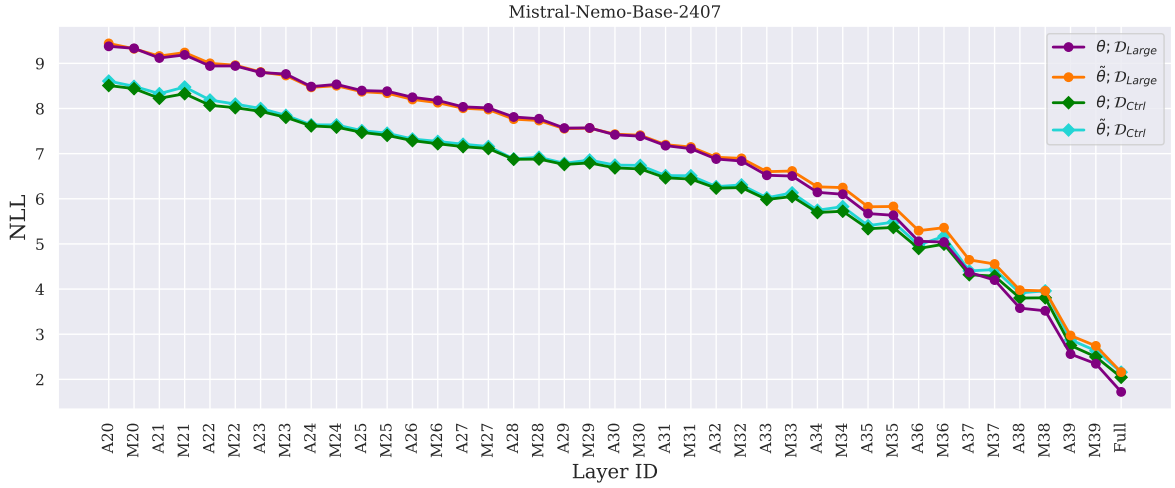

Figure 14: Early Exiting results on Llama3-8B. We decode the $z^{(l)}$ (A#) and $r^{(l)}$ (M#) across layers.

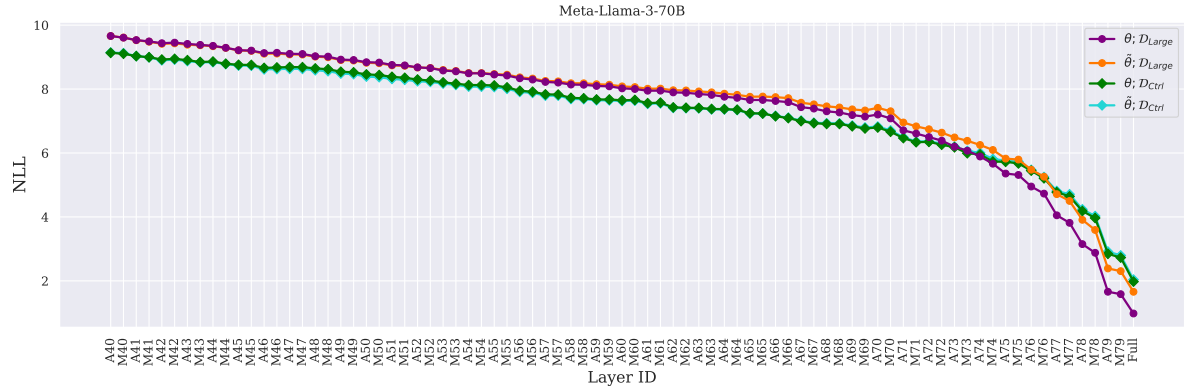

Figure 15: Early Exiting results on Llama3-8B. We decode$z^{(l)}$ (A#) and $r^{(l)}$ (M#) across layers.

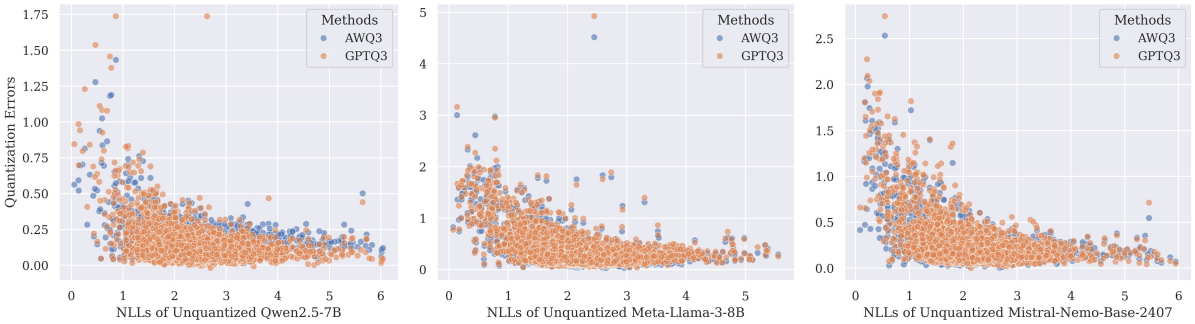

Figure 16: NLLs before quantization vs. quantization errors on different LLMs. Each dot represents a sample from FineWeb. In general, the low-frequency data (x-axis values ≥ 4) have smaller errors. On the other hand, many samples that suffer from large quantization errors have small NLLs before quantization.

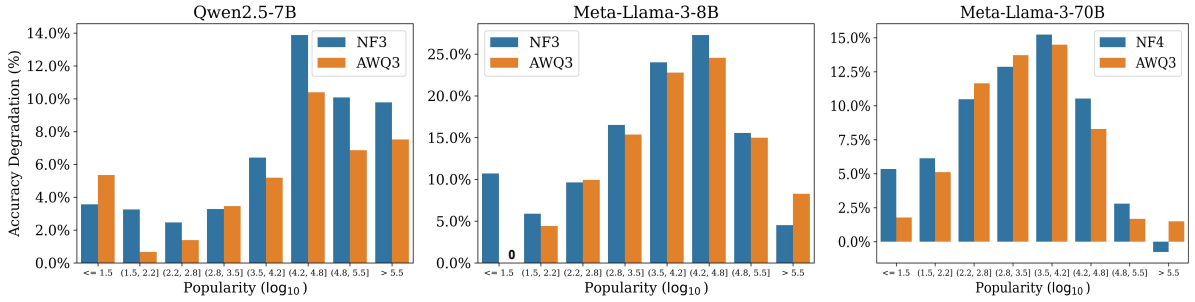

Figure 17: Accuracy degradation of different models on PopQA.

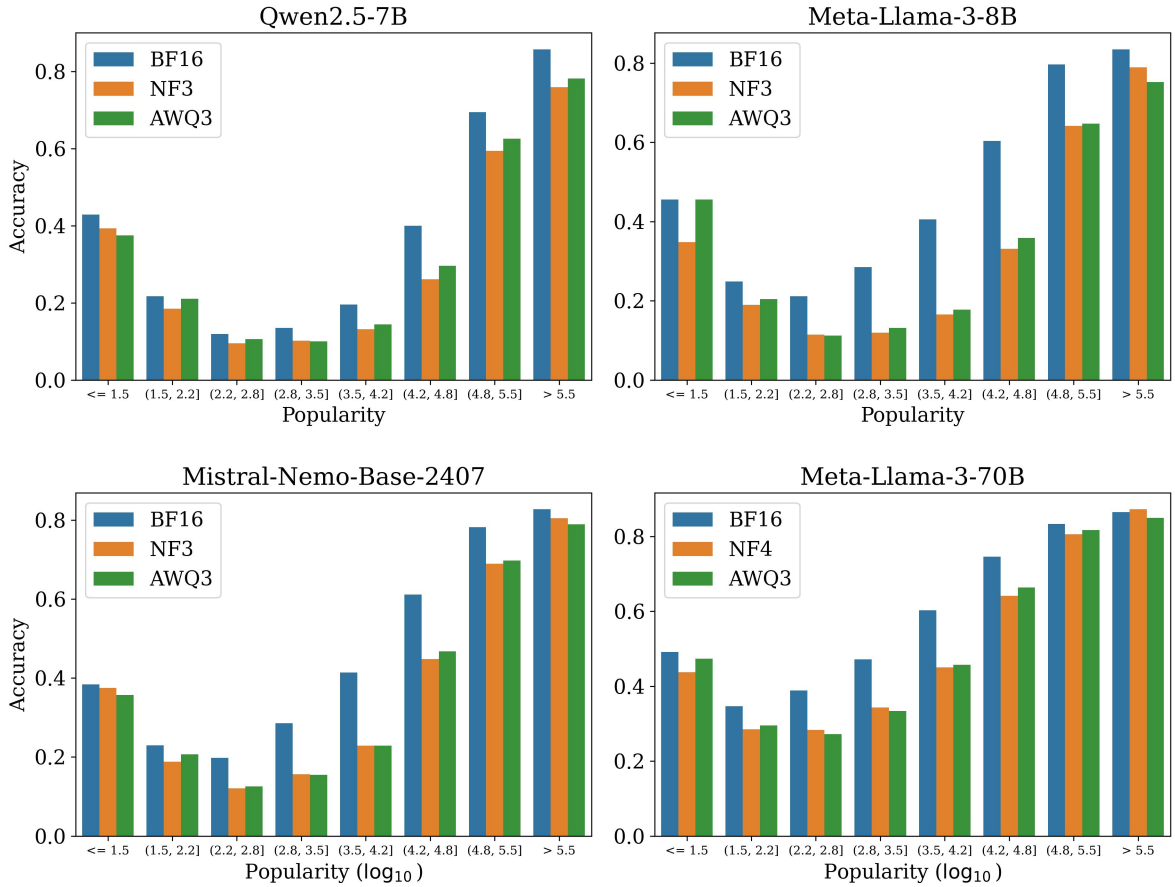

Figure 18: Absolute accuracies of different models on PopQA.

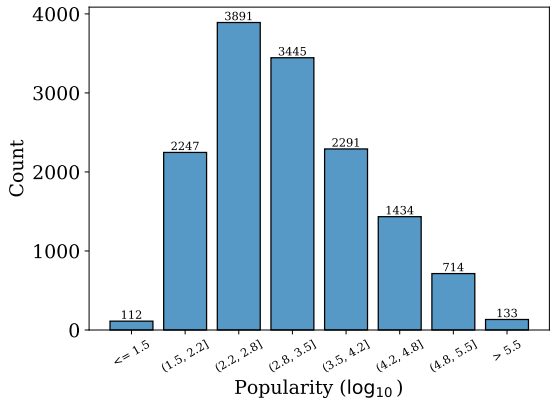

Figure 19: The histogram of the PopQA dataset. We partition the dataset into 8 buckets based on the sample popularity and ensure that each partition contains at least 100 samples.

	Layer	Rank of Outliers' $\lvert\gamma\rvert$	Median $\lvert\gamma\rvert$	Outliers' $\lvert\gamma\rvert$
Llama3-70B	40	[2, 9, 1, 10, 13]	0.1553	[0.0195, 0.0376, 0.0186, 0.0386, 0.0413]
	41	[2, 8, 1, 9, 10]	0.1582	[0.0170, 0.0312, 0.0142, 0.0320, 0.0330]
	42	[5, 1, 7, 10, 8]	0.1602	[0.0215, 0.0128, 0.0271, 0.0312, 0.0292]
	43	[1, 6, 2, 8, 9]	0.1621	[0.0117, 0.0275, 0.0154, 0.0315, 0.0317]
	44	[1, 7, 2, 9, 8]	0.1631	[0.0171, 0.0369, 0.0209, 0.0396, 0.0374]
	45	[1, 7, 10, 2, 7]	0.1660	[0.0130, 0.0383, 0.0420, 0.0209, 0.0383]
	46	[1, 7, 14, 12, 2]	0.1689	[0.0088, 0.0342, 0.0405, 0.0396, 0.0223]
	47	[1, 7, 8, 3, 14]	0.1699	[0.0090, 0.0320, 0.0364, 0.0206, 0.0396]
	48	[1, 6, 8, 2, 9]	0.1709	[0.0098, 0.0359, 0.0420, 0.0247, 0.0427]
	49	[1, 5, 12, 3, 12]	0.1738	[0.0128, 0.0354, 0.0476, 0.0297, 0.0476]
	50	[1, 6, 13, 3, 8]	0.1758	[0.0100, 0.0327, 0.0437, 0.0273, 0.0369]
	51	[1, 6, 11, 3, 8]	0.1777	[0.0117, 0.0369, 0.0471, 0.0312, 0.0427]
	52	[1, 7, 8, 3, 19]	0.1787	[0.0126, 0.0422, 0.0454, 0.0315, 0.0591]
	53	[1, 6, 11, 3, 14]	0.1816	[0.0136, 0.0337, 0.0422, 0.0255, 0.0461]
	54	[1, 6, 9, 2, 11]	0.1826	[0.0138, 0.0293, 0.0376, 0.0248, 0.0388]
	55	[1, 5, 9, 3, 11]	0.1846	[0.0110, 0.0330, 0.0427, 0.0277, 0.0454]
	56	[1, 6, 9, 3, 13]	0.1865	[0.0138, 0.0378, 0.0439, 0.0297, 0.0471]
	57	[1, 6, 9, 15, 3]	0.1885	[0.0124, 0.0344, 0.0408, 0.0459, 0.0282]
	58	[1, 6, 9, 15, 14]	0.1904	[0.0121, 0.0322, 0.0388, 0.0457, 0.0437]
	59	[1, 5, 12, 23, 14]	0.1934	[0.0206, 0.0420, 0.0559, 0.0684, 0.0583]
	60	[1, 6, 12, 27, 14]	0.1953	[0.0197, 0.0461, 0.0547, 0.0776, 0.0554]
	61	[1, 7, 12, 24, 14]	0.1982	[0.0177, 0.0459, 0.0532, 0.0693, 0.0562]
	62	[1, 6, 22, 14, 20]	0.2002	[0.0160, 0.0435, 0.0659, 0.0579, 0.0635]
	63	[1, 6, 12, 22, 15]	0.2031	[0.0199, 0.0530, 0.0640, 0.0781, 0.0679]
	64	[1, 8, 26, 18, 16]	0.2070	[0.0272, 0.0608, 0.0903, 0.0757, 0.0723]
	65	[1, 2, 9, 8, 26]	0.2109	[0.0225, 0.0403, 0.0635, 0.0603, 0.0903]
	66	[1, 9, 5, 6, 22]	0.2119	[0.0259, 0.0591, 0.0496, 0.0520, 0.0825]
	67	[1, 10, 6, 26, 7]	0.2158	[0.0266, 0.0625, 0.0537, 0.0918, 0.0559]
	68	[1, 9, 5, 27, 6]	0.2227	[0.0293, 0.0762, 0.0659, 0.1079, 0.0674]
	69	[1, 9, 2, 36, 7]	0.2266	[0.0294, 0.0732, 0.0425, 0.1201, 0.0688]
	70	[1, 12, 28, 6, 7]	0.2285	[0.0311, 0.0737, 0.1167, 0.0654, 0.0669]
	71	[1, 28, 16, 6, 7]	0.2344	[0.0312, 0.1328, 0.1006, 0.0771, 0.0776]
	72	[1, 33, 11, 9, 3]	0.2441	[0.0430, 0.1650, 0.0947, 0.0918, 0.0649]
	73	[1, 36, 9, 23, 3]	0.2520	[0.0483, 0.1729, 0.1089, 0.1455, 0.0723]
	74	[1, 36, 8, 18, 3]	0.2598	[0.0459, 0.1846, 0.1060, 0.1367, 0.0767]
	75	[1, 44, 11, 10, 3]	0.2676	[0.0796, 0.2139, 0.1250, 0.1230, 0.0923]
	76	[1, 43, 3, 15, 15]	0.3047	[0.0486, 0.2217, 0.0981, 0.1533, 0.1533]
	77	[1, 2, 52, 4, 5]	0.2891	[0.0554, 0.0820, 0.2080, 0.0869, 0.0913]
	78	[1, 4, 2, 20, 6]	0.2812	[0.0781, 0.0972, 0.0923, 0.1562, 0.1138]
	79	[9, 4, 1, 3, 13]	0.2236	[0.0688, 0.0500, 0.0309, 0.0454, 0.0747]
Mistral-12B	20	[1, 6, 22, 376, 4000]	0.8594	[0.001640, 0.137695, 0.330078, 0.648438, 0.902344]
	21	[1, 5, 22, 289, 2463]	0.9062	[0.001648, 0.113770, 0.324219, 0.644531, 0.906250]
	22	[1, 2, 22, 283, 2402]	0.9453	[0.002487, 0.044434, 0.369141, 0.687500, 0.941406]
	23	[1, 2, 22, 308, 3740]	0.9570	[0.001511, 0.089355, 0.410156, 0.726562, 0.988281]
	24	[2, 1, 261, 21, 4599]	1.0078	[0.080078, 0.001457, 0.730469, 0.365234, 1.062500]
	25	[2, 1, 167, 20, 1615]	1.0469	[0.102539, 0.017334, 0.691406, 0.359375, 1.023438]
	26	[2, 1, 184, 21, 1145]	1.0859	[0.092285, 0.001183, 0.722656, 0.390625, 1.039062]
	27	[5, 1, 151, 21, 840]	1.1172	[0.140625, 0.025146, 0.722656, 0.410156, 1.046875]
	28	[2, 1, 150, 22, 792]	1.1484	[0.094727, 0.005585, 0.757812, 0.429688, 1.070312]
	29	[2, 1, 175, 21, 945]	1.1875	[0.121582, 0.016113, 0.785156, 0.453125, 1.132812]
	30	[2, 1, 181, 19, 1080]	1.2266	[0.116211, 0.009583, 0.855469, 0.458984, 1.187500]
	31	[2, 1, 211, 3486, 21]	1.2578	[0.129883, 0.000584, 0.933594, 1.273438, 0.488281]
	32	[2, 1, 212, 4982, 18]	1.2812	[0.153320, 0.000881, 1.000000, 1.335938, 0.458984]
	33	[2, 1, 200, 5098, 18]	1.2812	[0.154297, 0.001419, 1.054688, 1.421875, 0.542969]
	34	[1, 2, 5111, 666, 21]	1.3125	[0.000793, 0.210938, 1.640625, 1.273438, 0.636719]
	35	[1, 2, 5112, 5054, 5110]	1.3594	[0.110352, 0.213867, 1.773438, 1.484375, 1.742188]
	36	[1, 2, 5114, 5111, 5064]	1.3906	[0.001472, 0.277344, 1.984375, 1.906250, 1.578125]
	37	[1, 19, 5116, 5107, 5095]	1.3828	[0.001205, 0.808594, 2.328125, 2.109375, 1.867188]
	38	[1, 5119, 5115, 5118, 5103]	1.4219	[0.316406, 3.078125, 2.921875, 3.031250, 2.359375]
	39	[1, 28, 61, 40, 5120]	1.6172	[0.601562, 1.265625, 1.476562, 1.406250, 5.437500]

Table 8: We rank the RMSNorm weights γ in descending order by the absolute value of each entry. We then compare the median weight with those of the outlier dimensions. We list the ranks and weight values corresponding to the 1st through 5th severest outliers. In Llama3-70B, all outlier dimensions have weights far below the median, suggesting that the model uses RMSNorm weights to suppress outliers. In Mistral-12B, the severest outliers have small weights (rank 1 or 2) across layers, but the other outliers sometimes have large weights (rank > 5000).