Rethinking the Outlier Distribution in Large Language Models: An In-depth Study

Anonymous ACL submission

Abstract

Investigating outliers in large language models (LLMs) is crucial due to their significant impact on various aspects of LLM performance, including quantization and compression. Outliers often cause considerable quantization errors, leading to degraded model performance. Identifying and addressing these outliers can enhance the accuracy and efficiency of the quantization process, enabling smoother deployment on edge devices or specialized hardware. Recent studies have identified two common types of outliers in LLMs: massive activations and channel-wise outliers. While numerous quantization algorithms have been proposed to mitigate their effects and maintain satisfactory accuracy, few have thoroughly explored the root causes of these outliers in depth.

In this paper, we conduct a comprehensive investigation into the formation mechanisms of these outliers and propose potential strategies to mitigate their occurrence. Ultimately, we introduce some efficient approaches to eliminate most massive activations and channel-wise outliers, facilitating efficient quantization.

1 Introduction

Large Language Models (LLMs) have emerged as a cornerstone in the field of natural language processing (NLP), transforming how we approach various linguistic tasks. These models, with their ability to understand and generate human-like text, have revolutionized applications ranging from conventional NLP tasks such as machine translation (Huang et al., 2023; Xu et al., 2024; Zhu et al., 2023), sentiment analysis (Miah et al., 2024; Wang et al., 2024; Deng et al., 2023) to advanced tasks such as code generation (Kazemitabaar et al., 2023; Thakur et al., 2024; Nakkab et al., 2024). However, the enormous size of LLMs, often reaching billions of parameters, presents substantial challenges for deployment, necessitating the use of techniques that enable efficient inference.

To address this, Post-Training Quantization (PTQ) (Frantar et al., 2022; Xiao et al., 2023; Lin et al., 2024a; Yao et al., 2022) provides a practical, low-cost approach for model quantization, either completely training-free or with minimal calibration effort (Cai et al., 2020; Li et al., 2021). In comparison to Quantization-Aware Training (QAT), which demands multiple fine-tuning iterations, PTQ incurs much lower computational costs, making it suitable for LLM. Unfortunately, outliers in LLM activations and KV vectors (Dettmers et al., 2022; Zeng et al., 2022) introduce significant magnitude variations among LLM elements, which in turn lead to a notable drop in model accuracy when low-precision PTQ is applied (Xiao et al., 2023; Tseng et al., 2024; Ashkboos et al., 2024b).

Prior research has identified two types of outliers in LLM activations. The first, massive activations (MAs), commonly appear across various LLMs and are typically linked to specific tokens in certain channels (Sun et al., 2024). The second type, channel-wise outliers (Dettmers et al., 2022; Xiao et al., 2023; Ashkboos et al., 2024b), manifests in bulk within specific channels. These findings have inspired a two-stage approach in modern quantization techniques: initially, methods are employed to eliminate outliers in the pretrained LLM, resulting in a model with a smoother value distribution in its activations. Subsequently, quantization algorithms such as GPTQ (Frantar et al., 2022) and OBQ (Frantar and Alistarh, 2022) are applied to produce low-precision LLMs, as shown in Figure 1.

Outlier smoothing is a crucial step in achieving efficient LLM quantization. Understanding the root causes of outliers is essential for developing effective quantization techniques and gaining deeper insights into model behavior and robustness. While prior studies have identified the presence of MAs and channel-wise outliers, and proposed methods to mitigate them (Sun et al., 2024; Liu et al., 2024; Bini et al., 2024; Xiong et al., 2024), none have

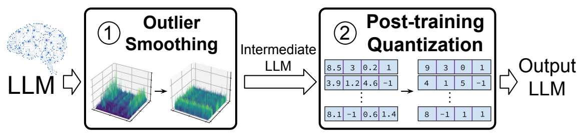

Figure 1: Given a pretrained LLM, techniques are first applied to smooth out the outliers in its activations. The resulting model is then quantized, achieving superior accuracy.

explored the fundamental reasons behind the existence of these outliers from a numerical perspective, particularly with operator-level granularity. This finer-grained understanding is crucial, as different layers and operators may contribute uniquely to the formation and propagation of LLM outliers, influencing both performance and accuracy in low-precision LLMs.

In this work, we investigate the underlying reasons for the existence of outliers in LLMs at the operator level through extensive empirical analysis. Our study provides valuable insights to guide the development of effective outlier smoothing algorithms. Building on these findings, we propose some novel methods to efficiently mitigate the majority of massive activations and channel-wise outliers without compromising model accuracy. This significantly reduces the complexity of subsequent LLM quantization processes. In summary, our findings on LLM outliers can be summarized as follows:

- We empirically demonstrate that massive activations (MAs) are predominantly generated in the initial layers the model. Once these MAs arise, they persist throughout the LLM, being propagated through subsequent layers via residual connections.

- Previous studies indicate that the removal of MAs can significantly impact the quantization process. Surprising, our empirical analysis shows that eliminating MAs introduced by residual connections has no measurable effect on the model's accuracy. Notably, these MAs constitute the majority of MAs in LLMs.

- Channel-wise outliers in LLMs initially emerge due to the normalization operations within the model. The rescaling operation within the normalization layer exacerbates this issue by introducing an increasing number of channel-wise outliers.

- Certain channels within the weight matrices can also contribute to the emergence of channel-wise outliers in the intermediate results of LLMs.

2 Background and Related Work

2.1 LLM Operations

Modern LLMs (e.g., Llama series (Touvron et al., 2023a,b), GPT series (Radford et al., 2019; Brown, 2020)) are constructed as a stack of transformer decoders, with each decoder comprising two fundamental components: a Self-Attention (SA) block and a feed forward network (FFN), as depicted in Figure 2 (a). During the LLM serving process, the input to the Self-Attention (SA) block is first processed by a normalization operation (e.g., LayerNorm or RMSNorm). As detailed in Figure 3(d), this normalization consists of two key steps: standardization and rescaling. Specifically, the input X is normalized by subtracting its mean μ_X and dividing by its standard deviation σ_X. Subsequently, each channel of the standardized output is scaled by a learnable parameter γ and shifted by another learnable parameter β.

The output of the normalization operation is then multiplied with three weight matrices W_Q, W_K, and W_V, yielding the outputs referred to as query (q), key (k), and value (v), which is shown as x_3, x_4 and x_5 in Figure 2, respectively. The resulting q and k, in combination with v, will then undergo multiplication, Softmax, and residual addition to generate the SA output, as shown in Figure 2 (b).

The output from the SA will then be passed to the FFN for further processing, which typically involves a gated MLP (Radford, 2018; Radford et al., 2019) (Figure 2 (c)) or standard MLP (Liu et al., 2021; Touvron et al., 2023a,b) (Figure 2 (d)). The FFN consists of a normalization operation, multiple fully connected (FC) layers along with an intermediate activation function, such as GeLU (Hendrycks and Gimpel, 2016) or SiLU (Hendrycks and Gimpel, 2016).

2.2 Outlier in LLM

As prior studies have demonstrated (Dettmers et al., 2022; Zeng et al., 2022; Sun et al., 2024), outliers can be categorized into two types: *massive activations (MA)* and *channel-wise outliers (CO)*. The presence of outliers in LLM activations and KV vectors (Dettmers et al., 2022; Zeng et al., 2022) often causes a significant drop in model accuracy

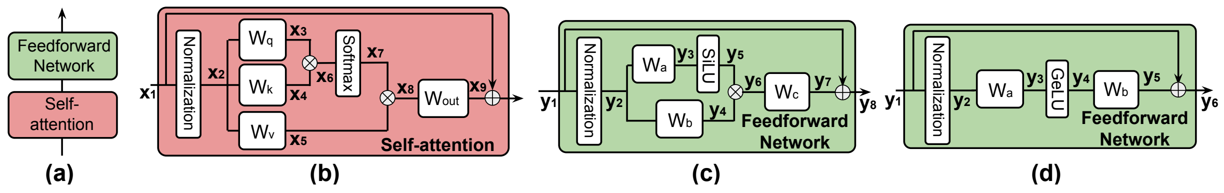

Figure 2: (a) Architecture of a LLM decoder block. (b), (c) and (d) show the architectures of self-attention block, standard FFN (conventional MLP), and gated FFN (GLU), respectively. The notations will be used throughout the rest sections.

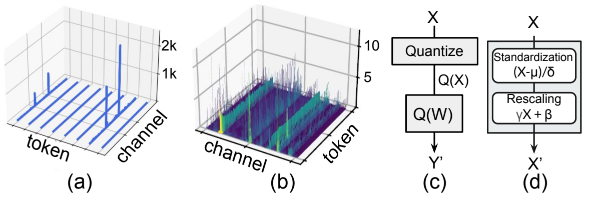

Figure 3: (a) One example of massive activation presented in the inputs x_1. (b) An example of outlier channel at position x_2 in the LLM. (c) The existence of outlier will lead to an output Y' different from the original output Y. (d) The normalization operations within LLM.

when low-precision PTQ is applied (Xiao et al., 2023; Tseng et al., 2024; Ashkboos et al., 2024b).

While earlier research, such as (Bondarenko et al., 2023), has shown that the attention mechanism can lead to excessive activations by concentrating too much on specific tokens, resulting in scenarios where the mechanism fails to remain inactive and creates an outlier problem, these studies mainly focus on BERT architectures. In contrast, our analysis expands the scope to include LLaMA, GPT, and Qwen models. This broader investigation provides new insights into the architectural changes that can give rise to activation outliers. As discussed in (Li et al., 2024), performing kurtosis on the activation tensor to reflect on MAs in GLUs, though our focus here remains on characterizing and categorizing the outlier phenomenon and providing a simpler method to remove MAs.

To demonstrate this, we profile the inputs x_2 to the W_Q, W_K, and W_V matrices within the self-attention (SA) block, as shown in Figure 2, using the Wikitext dataset (Merity et al., 2016) on the LLaMA-7B model. Following the notations in Figure 2, we record the input to the normalization operation, x_1. The results presented in Figure 3 (a) highlights the presence of MAs in x_1, with magnitudes often reaching thousands. Furthermore, these MAs propagate through the normalization operation, causing x_2 to also exhibit some outliers. Although the magnitude of these outliers is reduced after normalization, they remain significant. Additionally, Figure 3 (b) shows that distribution of the COs in x_2, corroborating earlier findings (Xiao et al., 2023; Ashkboos et al., 2024b,a; Frantar and Alistarh, 2023). To isolate the impact of MAs, we remove them from x_2 to better illustrate the distribution of COs.

Figure 3 (c) illustrates that when the input X contains both types of outliers, its quantized version $Q(X)$ experiences significant quantization error. As a result, the output Y', derived from the quantized input $Q(X)$ and quantized weight $Q(W)$, deviates considerably from the original output $Y = XW$, leading to a noticeable degradation in accuracy.

2.3 Outlier Smoothing for Low-precision LLM Quantization

Reducing quantization error is crucial for achieving effective low-precision model quantization. However, as highlighted by LLM.int8() (Dettmers et al., 2022), directly quantizing LLMs to INT8 leads to significant accuracy loss due to the presence of outliers. To address these outliers, LLM.int8() employs a mixed-precision decomposition scheme. While this approach preserves model accuracy, its fine-grained decomposition introduces computational overhead and potential performance bottlenecks.

Olive (Guo et al., 2023) addresses the impact of MAs on low-precision quantization by proposing a hybrid quantization scheme that quantizes MAs separately from the remaining elements. Similarly, PrefixQuant (Chen et al., 2024) groups tokens with MAs and jointly quantizes them, resulting in reduced quantization error. This approach has also been applied to KV cache quantization (Zhang et al., 2024a), following the same principle. Collectively, these studies highlight the critical importance of understanding outlier behavior within LLMs to develop more effective quantization strate-

Table 1: TMA values distribution within different LLMs. Initial Top-1 and Initial Top-2 denote the MAs with the largest and second largest magnitudes within the initial LLM layers. Last-1 and Last-2 denote the last and second last layers within LLM. The N/A for smaller models indicate that the balancing of signs observed in True Massive Activations (TMAs) is handled only by the last layer.

	LLaMA3.2-3B		**LLaMA3.1-8B**		**LLaMA2-13B**		**GPT-2**		**Qwen2.5-7B**	
Massive Activations	**Value**	**Position**	**Value**	**Position**	**Value**	**Position**	**Value**	**Position**	**Value**	**Position**
Initial Top-1	-328.25	(0, 588)	-300.5	(0, 788)	-1211.0	(0, 4743)	-449.82	(0, 1591)	-9057.43	(0, 458)
Initial Top-2	-303.25	(0, 1016)	-274.75	(0, 1384)	-708.0	(0, 2100)	-388.98	(0, 506)	-5757.42	(0, 2570)
Last-2 Top-1	N/A	N/A	N/A	N/A	414.75	(0, 4743)	169.89	(0, 1591)	9178.38	(0, 458)
Last-2 Top-2	N/A	N/A	N/A	N/A	288.25	(0, 2100)	159.61	(0, 506)	4645.87	(0, 2570)
Last-1 Top-1	262.5	(0, 1016)	299.75	(0, 788)	824.0	(0, 4743)	277.06	(0, 1591)	2688.36	(0, 458)
Last-1 Top-2	249.5	(0, 588)	273.5	(0, 1384)	477.0	(0, 2100)	243.73	(0, 506)	2609.71	(0, 2570)

gies.

On the other hand, to eliminate the channel-wise outliers, SmoothQuant (Xiao et al., 2023) proposes migrating the quantization challenge from activations to weights using scale invariance. This allows INT8 quantization for both weights and activations across all matrix multiplications in LLMs. Outlier Suppression+ (Wei et al., 2023) further enhances quantization by introducing a fast and stable scheme for calculating scaling values, effectively balancing the quantization burden.

To reduce manual intervention and improve performance under extremely low-bit quantization, OmniQuant (Shao et al., 2023) introduces Learnable Weight Clipping and Learnable Equivalent Transformation, optimizing both weight-only and weight-activation quantization processes. In W4A8 quantization with weight clipping, QQQ (Zhang et al., 2024b) dynamically manages outliers through adaptive smoothing. Additionally, QServe (Lin et al., 2024b) introduces SmoothAttention to mitigate accuracy degradation caused by 4-bit KV quantization. Both QQQ and QServe have greatly improved LLM accuracy under W4A8 quantization.

While most previous studies focus on mitigating the impact of channel-wise outliers during the quantization process, this work investigates the root causes of both MAs and COs. We propose some insights to address these outliers by targeting and removing them at their fundamental level.

3 Empirical Study on Massive Activation

3.1 Settings

To investigate the formation of massive activations (MAs), we conduct experiments on various LLMs, including the LLaMA series (Touvron et al., 2023a,b), GPT-2 (Achiam et al., 2023), and Qwen (Yang and et al., 2024), using two datasets: WikiText (Merity et al., 2016) and C4 (Huggingface, 2022). Each experiment is averaged over 100 random samples from the dataset. LLM performance is evaluated using the perplexity (PPL) metric.

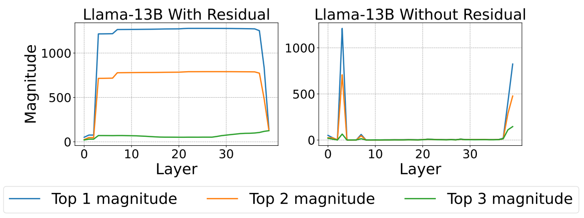

Figure 4: Left: TMAs and FMAs within the input of LLaMA-13B across each layer. Right: after removing the MAs in residual connection, only TMA left.

Following the definition of MAs from (Sun et al., 2024), an activation is considered **massive** if its magnitude exceeds 100 and is at least 1,000 times greater than the median activation magnitude.

3.2 Observations on Massive Activation

In our experiments, we investigate the existence of MAs in the hidden state tensors within the attention and MLP blocks. Next, we modify the inference process of LLMs by directly intervening in the layers where massive activations emerge. Specifically, for any hidden state exhibiting massive activations, we manually set those activations to fixed values. The modified hidden state is then passed to the subsequent layer, with the remaining computations proceeding as usual. As a result of these studies, we have the following surprising observations that differ from or were not reported in earlier literature, summarized as follows:

Massive Activations are first appeared in the FFN Block: We found that for all LLMs, MAs first appear within the feed-forward network (FFN) of first layer. Specifically, in models using gated MLPs, such as the LLaMA series and Qwen, MAs

Table 2: Impact of MAs on the performances (in perplexity) of LLaMA, GPT-2, and Qwen models.

	LLaMA3.2-3B		LLaMA3.1-8B		LLaMA2-13B		GPT-2		Qwen2.5-7B	
Intervention	**WikiText**	**C4**	**WikiText**	**C4**	**WikiText**	**C4**	**WikiText**	**C4**	**WikiText**	**C4**
Original	5.567	10.790	6.941	9.046	4.355	6.405	14.795	19.460	6.520	11.773
TMAs to mean at y_7	1124111.75	21046.82	21281.49	1301562.25	1301562.25	6469.42	14.841	19.560	71216.17	66588.86
TMAs to zeroes at y_7	1138151.23	21951.41	21601.10	1302018.53	1309211.61	7128.32	14.911	19.928	71835.61	67518.35
TMAs to mean at y_6	6.053	14.423	7.026	10.046	4.355	6.405	14.795	19.460	6.537	11.797
TMAs to mean at y_6	6.237	14.767	7.147	10.255	4.371	6.498	14.831	19.565	6.642	13.021

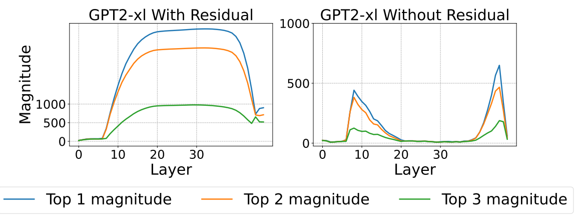

Figure 5: Left: TMAs and FMAs within the input of GPT-2 across each layer. Right: after removing the MAs in residual connection, only TMA left.

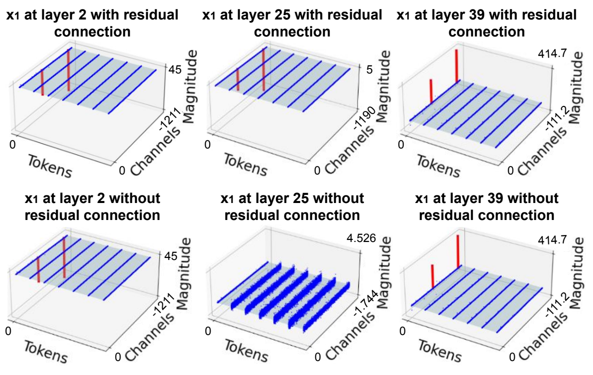

Figure 6: The red lines drawn to the origin plane indicate the MAs. Top three figures are the TMA and FMA of Llama-13B. The bottom three figures are the TMA of Llama 13B model after removing MAs in the residual connection. Layer 4 and Layer 39 have TMA for the same channel and token but with the opposite sign. The MAs of layer 25 is eliminated after the removal of the residual connection.

emerge in y_6, the product of y_4 and y_5, as illustrated in Figure 2 (c). In contrast, for LLMs with conventional MLPs, like GPT-2, MAs are first produced immediately after the GeLU activation, represented by y_4 in Figure 2 (d).

Most of MAs are caused by residual connections within LLM: Among the MAs observed across LLM layers, most are propagated through residual connections in both the self-attention (SA) and FFN blocks. Specifically, after initially appearing in the FFN, the residual links carry these MAs through the inputs of SA and FFN blocks across the middle layers of the LLM. These MAs are not newly generated but are instead carried forward from previously produced MAs through the intermediate layers via residual connections. For the final layers (e.g., 39th and 40th layers in LLaMA), MAs are generated spontaneously and are not caused by residual connections. To differentiate these MAs, we call the MAs that are caused by the residual link **Fake MAs** (FMAs), and rest of MA **True MAs** (TMAs).

To illustrate the presence of TMAs and FMAs, we conduct experiments on LLaMA-13B and GPT-2. The left side of Figure 4 and Figure 5 show the top three elements with the highest magnitudes, identified as MAs, across the input of each layer. Building on this, we remove the residual connections for both the SA and FFN layers throughout the entire LLM. The right side of Figure 4 and Figure 5 present the results after these residual connections are removed from all layers. Our observations show that the TMAs at y_6 effectively eliminates all TMAs and FMAs. Due to space constraints, we present results only for LLaMA-13B and GPT-2, although similar behaviors are observed in other LLMs.

Trends on TMA Magnitudes: Across various models, TMAs exhibit consistent behavior: their magnitude remains fixed within specific channels, regardless of the input sequence tokens. Analyzing the sign of these TMAs reveals a clear pattern: in the final layers, TMAs have a similar magnitude but opposite sign compared to those in the initial layers, occurring at the same channel positions. This indicates that TMAs generated in the early layers are effectively suppressed in the later layers. Table 1 presents the average magnitudes of TMAs across multiple LLMs, highlighting their presence in the first initial layers and the last two layers. It also shows the top two MAs with the largest magnitudes in each layer's input, along with their corresponding token and channel indices, shown in the first and second number within the bracket. While models like GPT-2 and Qwen display multiple initial and final layers with high activation magnitudes, the observed magnitude and sign trends persist. Figure 6 shows that layer 2 of LLaMA-13B has a negative TMA while at layer 39 there is a positive TMA at the same channel and token position.

3.3 Impact of Massive Activation Values on LLM Accuracy

Building on the presence of TMAs, and FMAs, we next analyze their impact on LLM accuracy. Specifically, we replace all TMAs, which are located at y_6 of FFN with either zero or the mean value of their respective tensors. As shown in Table 2, the results remain comparable to the original LLM. Notably, for LLaMA2-13B, GPT-2, and Qwen, the PPL values are nearly identical to those of the original LLM on both WikiText-2 and C4, demonstrating that TMAs, and FMAs can be effectively eliminated without any negative impact on accuracy performance.

In contrast, the removal of TMAs located at y_7 of the FFN results in disastrous effects on LLM performance. As shown in Table 2, replacing TMAs with mean or zero values significantly increases PPL across models, with the exception of GPT-2. Thus, we conclude that most FMAs can be safely removed by replacing them with either zero or the mean of the tensor containing them. However, TMAs are essential and must be retained to preserve LLM functionality.

3.4 Insights for MA Smoothing

The presence of MAs is widely acknowledged as a major challenge in LLM quantization, particularly when aiming to enable efficient matrix multiplication within SA. As demonstrated in Section 3.2 and Section 3.3, all FMAs can be effectively eliminated by replacing them with either the mean value or zero, with negligible impact on LLM performance. This makes the corresponding activation matrices significantly easier to quantize.

In contrast, removing TMAs directly leads to severe performance degradation. As a result, existing outlier smoothing techniques, such as mathematical invariance transformations (Ashkboos et al., 2024b) and (Xiao et al., 2023), are typically applied exclusively to these outliers. Since mathematical invariance transformations (e.g., Hadamard transform) introduce additional computational overhead for outlier smoothing, limiting their application to the small number of TMAs significantly reduces the overall computational cost.

4 Empirical Study on Channel-wise Outliers

4.1 Settings

In addition to the presence of MAs, channel-wise outliers (CO) are also observed within the intermediate results of LLMs, as noted in several prior studies (Xiao et al., 2023; Ashkboos et al., 2024b; Tseng et al., 2024; Liu et al., 2024). These outliers significantly degrade the performance of low-precision LLM quantization. Following our study on MAs, we examine the presence and formation of channel-wise outliers in various LLMs (LLaMA series and GPT-2) using two datasets: WikiText and C4. LLM performance is assessed using the perplexity (PPL) metric, with each experiment averaged over 100 random samples. Since no formal study has been conducted on channel-wise outlier before, we use the following criteria to search for the channel-wise outlier.

For each channel A_j within an activation matrix A, it is classified as an **outlier channel** if it satisfies the following criteria:

- The mean of A_j exceeds the overall average of the tensor by more than $m\sigma_A$, where m is a predefined parameter and σ_A is the standard deviation of elements within A.

- The standard deviation of A_j is below a threshold β.

The first criterion ensures that the average value of the entire channel is sufficiently high to qualify as an outlier, while the second criterion ensures that all elements within the channel have similar magnitudes, aligning with outlier channel behavior. Without loss of generality, in the following experiment, m is set to 4, and β is set to $1/3$. We also present the results under different settings in the subsequent sections.

4.2 Observations on channel-wise Outliers

We examine the presence of outlier channels in the input, output, and hidden state tensors within the SA and MLP blocks. These correspond to the inputs and outputs of the SA and FFN blocks (e.g., x_1, x_2, y_1, and y_2) as well as intermediate results (e.g., x_3, x_4, y_3, and y_5) depicted in Figure 2 (b), (c), and (d). Next, we delve deeper into normalization operation and attention weight matrix multiplications, looking at how the learned model weights

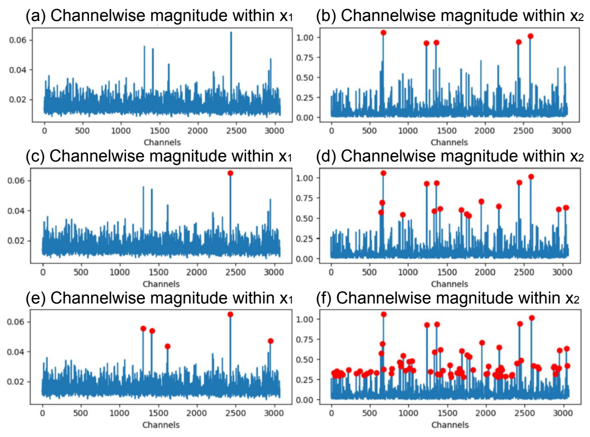

Figure 7: Changes on number of channel-wise outlier after the input x_1 of the first layer of LLaMA-13B passing through the RMSNorm. Outlier channels denoted by red dots. (a) and (b) shows the results by setting m=6, (c) and (d) for m=4, and (e) and (f) for m=2, respectively.

associated with each of these transformations affect the occurrence of outlier channels in the output activations. Specifically, we observe the effects of smoothening the outlier channels within these weights by replacing them with fixed values. The observations are summarized below.

Channel-wise outliers first arise after the normalization operation in first layer: We observe that in all evaluated LLMs, outlier channels first emerge during the initial normalization operation preceding the SA block. Figure 7 illustrates the average magnitude of each channel in the input and output of the normalization operation within the first layer of the SA block. Red dots represent the outlier channels. The results are shown by varying the criteria for channel-wise outliers, with m set to 2, 4, and 6, respectively. Notably, the number of outlier channels increases largely after the normalization operation.

Learned rescaling operations inside Normalization block produces outlier channel: As shown in Figure 3 (d), the normalization operation with in LLM, such as LayerNorm or RMSNorm, are further contains two components: standardization and rescaling. For example, in LayerNorm, the input is first normalized by subtracting its mean μ and dividing by its standard deviation σ. Each channel of the normalized output is then scaled by a learnable parameter γ and shifted by another learnable parameter β.

We conduct an outlier analysis of tensors within the normalization block, as illustrated in Figure 8. To isolate the effects of channel-wise outliers, we first eliminate massive activations (MAs) from the input, allowing for a clearer visualization of outlier channels. In the normalization process, the inputs undergo token-wise standardization followed by a rescaling operation. Our findings reveal that the standardization step does not introduce additional channel-wise outliers (Figure 8 (c)). However, the rescaling operation has a channel-specific impact, which can lead to an increase in channel-wise outliers, as depicted in Figure 8 (d).

To further validate the impact of the rescaling operation, we modify the rescaling factor vector γ by identifying the indices associated with the outlier channels in the output of the normalization operation. This modification was applied to the normalization layers within both SA and FFN layers. Specifically, the rescaling factor elements at these indices were replaced with either the mean of the rescaling vector or zero. Both modifications result in a noticeable reduction in the number of outlier channels in the subsequent outputs, as shown in Figure 9.

4.3 Observations on channel-wise Outliers in weight matrix multiplications

In this section, we examine the presence of channel-wise outliers during matrix multiplication with weight tensors. As a case study, we focus on the Query weight matrix ($\mathbf{W_q}$) within the SA block, and Key and Value matrices have the same trends. When examining the output activations ($\mathbf{x_3}$), new channel-wise outliers emerge that are absent in the input activations ($\mathbf{x_2}$). Specifically, $\mathbf{x_3}$ can be computed as follows:

$$\mathbf{x_3} = \mathbf{W_q} \cdot \mathbf{x_2} \tag{1}$$

If channel-wise outliers are observed in $\mathbf{x_3}$ but not in the corresponding input activations channel $\mathbf{x_2}$, we hypothesize that specific channels (rows) in $\mathbf{W_q}$ are responsible for the existence of new channel-wise outlier. These channels, which constitute approximately 1% of all channels within W_q, appear to hold greater numerical importance compared to others on LLM accuracy. We call it *Outlier Triggering Channels* (OTC).

An important but subtle observation is that these OTCs do not exhibit outlier characteristics when analyzing $\mathbf{W_q}$ alone, based on mean and standard deviation statistics. However, their interaction with $\mathbf{x_2}$ gives rise to outlier activations in $\mathbf{x_3}$. This finding highlights the critical role of OTCs in outlier

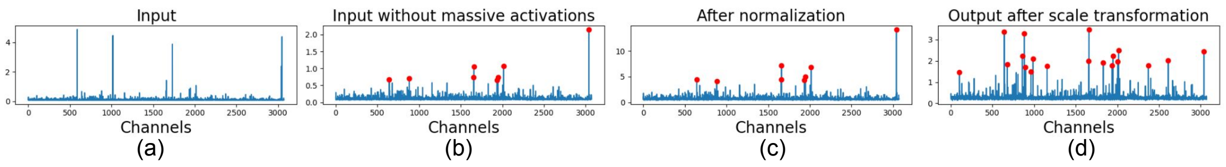

Figure 8: Changes on number of channel-wise outliers within a normalization block of SA. Outlier channels denoted by red dots. The channel-wise means of (a) the input x_1, (b) x_1 after removing the MAs, (c) the output of the standardization operation, and (d) the output of normalization x_2 are plotted. A similar observation has been observed for the normalization block within FFN.

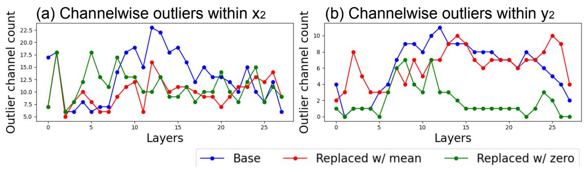

Figure 9: The blue line shows the number of outlier channels in the normalization layer inputs for each LLM layer. To identify the source of these outliers, we examine the corresponding rescaling factors γ that contribute to the channel-wise outliers. These rescaling factors are then replaced with either their mean values (red lines) or zeros (green lines).

Table 3: Analysis on the Importance of OTC, other LLMs also have similar trends

	LLaMA3.2-3B		LLaMA3.1-8B		LLaMA2-13B	
Intervention	**WikiText**	**C4**	**WikiText**	**C4**	**WikiText**	**C4**
base model	5.567	10.790	6.941	9.046	4.355	6.405
Remove OTCs	38.924	165.396	480.8123	465.2235	774.7298	15398.1279
Remove random channels	7.5094	11.990	7.1700	18.602	4.4455	6.682

formation, despite their seemingly unremarkable statistical profile in isolation. To test this hypothesis, we evaluate model performance by modifying $\mathbf{W_q}$ in two ways: (a) setting all elements within the OTCs, which comprise approximately 1% of the total number of channels in $\mathbf{W_q}$, to their mean values, and (b) setting a random 1% of channels to their mean values. The results of these interventions are presented in Table 3. The modification on OTC will cause a greater accuracy drop than that on equivalent amount of random channels. This comparison highlights the importance of specific weight channels that contribute to the presence of channel-wise outliers on the LLM accuracy. Similar studies have been performed on the key matrix and observe the trend being similar to query matrix, while the value matrix does not follow this trend and remain unaffected even after removing the OTC.

4.4 Insights for channel-wise Outlier Smoothing

Based on the results presented in Section 4.2 and Section 4.3, we conclude that the rescaling factor γ in the rescaling operations within the normalization layer plays a significant role in determining the number of channel-wise outliers in x_2 and y_2. These outliers are subsequently propagated into the matrix multiplication processes. To effectively mitigate channel-wise outliers in the input, a great strategy is to fine-tune the rescaling factors γ to reduce their variation. This adjustment results in x_2 having fewer outlier channels. However, simply setting the corresponding rescaling factor to a fix value will lead to significant accuracy drop.

OTCs within the weight matrices greatly contribute to channel-wise outliers in the intermediate results of LLMs. A potential solution to address this issue is to adopt parameter-efficient fine-tuning techniques, which can effectively eliminate OTCs without requiring extensive changes to the model.

5 Conclusion

Outliers in LLMs are crucial to address because of their significant impact on the accuracy of quantized LLMs. In this paper, we undertake a detailed investigation into the mechanisms behind the formation of outliers and develop strategies to mitigate their effects. We explore the causes of these outliers and propose practical approaches for their elimination, setting the stage for more efficient quantization processes.

Our comprehensive analysis not only highlights the challenges posed by outliers but also provides innovative solutions that could be pivotal for the advancement of quantization techniques in LLMs. We hope our findings make a valuable contribution to the ongoing research within the LLM community, especially in addressing the complexities of quantization challenges presented by outliers.

Limitations

While this survey offers a comprehensive overview of outliers within LLMs, it is important to acknowledge some limitations. The study of outliers is specifically tailored to LLMs, and there is scope for extending this research to other types of large models that handle multimodal inputs. Further investigation in these areas could provide a broader understanding of outlier effects across different model architectures and enhance the robustness of multimodal systems.

References

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

Saleh Ashkboos, Maximilian L Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. 2024a. Slicegpt: Compress large language models by deleting rows and columns. *arXiv preprint arXiv:2401.15024*.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian L Croci, Bo Li, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. 2024b. Quarot: Outlier-free 4-bit inference in rotated llms. *arXiv preprint arXiv:2404.00456*.

Lorenzo Bini, Marco Sorbi, and Stephane Marchand-Maillet. 2024. Characterizing massive activations of attention mechanism in graph neural networks. *arXiv preprint arXiv:2409.03463*.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. 2023. Quantizable transformers: Removing outliers by helping attention heads do nothing. *Preprint*, arXiv:2306.12929.

Tom B Brown. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.

Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2020. Zeroq: A novel zero shot quantization framework. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13169–13178.

Mengzhao Chen, Yi Liu, Jiahao Wang, Yi Bin, Wenqi Shao, and Ping Luo. 2024. Prefixquant: Static quantization beats dynamic through prefixed outliers in llms. *arXiv preprint arXiv:2410.05265*.

Xiang Deng, Vasilisa Bashlovkina, Feng Han, Simon Baumgartner, and Michael Bendersky. 2023. Llms to the moon? reddit market sentiment analysis with large language models. In *Companion Proceedings of the ACM Web Conference 2023*, pages 1014–1019.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022. Gpt3. int8 (): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35:30318–30332.

Elias Frantar and Dan Alistarh. 2022. Optimal brain compression: A framework for accurate post-training quantization and pruning. *Advances in Neural Information Processing Systems*, 35:4475–4488.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pages 10323–10337. PMLR.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*.

Cong Guo, Jiaming Tang, Weiming Hu, Jingwen Leng, Chen Zhang, Fan Yang, Yunxin Liu, Minyi Guo, and Yuhao Zhu. 2023. Olive: Accelerating large language models via hardware-friendly outlier-victim pair quantization. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*, pages 1–15.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Hui Huang, Shuangzhi Wu, Xinnian Liang, Bing Wang, Yanrui Shi, Peihao Wu, Muyun Yang, and Tiejun Zhao. 2023. Towards making the most of llm for translation quality estimation. In *CCF International Conference on Natural Language Processing and Chinese Computing*, pages 375–386. Springer.

Huggingface. 2022. C4 dataset. https://huggingface.co/datasets/legacy-datasets/c4.

Majeed Kazemitabaar, Xinying Hou, Austin Henley, Barbara Jane Ericson, David Weintrop, and Tovi Grossman. 2023. How novices use llm-based code generators to solve cs1 coding tasks in a self-paced learning environment. In *Proceedings of the 23rd Koli Calling International Conference on Computing Education Research*, pages 1–12.

Shiyao Li, Xuefei Ning, Luning Wang, Tengxuan Liu, Xiangsheng Shi, Shengen Yan, Guohao Dai, Huazhong Yang, and Yu Wang. 2024. Evaluating quantized large language models. *Preprint*, arXiv:2402.18158.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. 2021. Brecq: Pushing the limit of post-training quantization by block reconstruction. *arXiv preprint arXiv:2102.05426*.

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

Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. 2024b. Qserve: W4a8kv4 quantization and system co-design for efficient llm serving. *arXiv preprint arXiv:2405.04532*.

Hanxiao Liu, Zihang Dai, David So, and Quoc V Le. 2021. Pay attention to mlps. *Advances in neural information processing systems*, 34:9204–9215.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. 2024. Spinquant–llm quantization with learned rotations. *arXiv preprint arXiv:2405.16406*.

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

Md Saef Ullah Miah, Md Mohsin Kabir, Talha Bin Sarwar, Mejdl Safran, Sultan Alfarhood, and MF Mridha. 2024. A multimodal approach to cross-lingual sentiment analysis with ensemble of transformer and llm. *Scientific Reports*, 14(1):9603.

Andre Nakkab, Sai Qian Zhang, Ramesh Karri, and Siddharth Garg. 2024. Rome was not built in a single step: Hierarchical prompting for llm-based chip design. In *Proceedings of the 2024 ACM/IEEE International Symposium on Machine Learning for CAD*, pages 1–11.

Alec Radford. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. 2023. Omniquant: Omnidirectionally calibrated quantization for large language models. *arXiv preprint arXiv:2308.13137*.

Mingjie Sun, Xinlei Chen, J Zico Kolter, and Zhuang Liu. 2024. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*.

Shailja Thakur, Baleegh Ahmad, Hammond Pearce, Benjamin Tan, Brendan Dolan-Gavitt, Ramesh Karri, and Siddharth Garg. 2024. Verigen: A large language model for verilog code generation. *ACM Transactions on Design Automation of Electronic Systems*, 29(3):1–31.

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

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

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. 2024. Quip#: Even better llm quantization with hadamard incoherence and lattice codebooks. *Preprint*, arXiv:2402.04396.

Zeyu Wang, Yue Zhu, Shuyao He, Hao Yan, and Ziyi Zhu. 2024. Llm for sentiment analysis in e-commerce: A deep dive into customer feedback. *Applied Science and Engineering Journal for Advanced Research*, 3(4):8–13.

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 optimal shifting and scaling. *arXiv preprint arXiv:2304.09145*.

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.

Jing Xiong, Jianghan Shen, Fanghua Ye, Chaofan Tao, Zhongwei Wan, Jianqiao Lu, Xun Wu, Chuanyang Zheng, Zhijiang Guo, Lingpeng Kong, et al. 2024. Uncomp: Uncertainty-aware long-context compressor for efficient large language model inference. *arXiv preprint arXiv:2410.03090*.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *arXiv preprint arXiv:2401.08417*.

An Yang and et al. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. 2022. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Advances in Neural Information Processing Systems*, 35:27168–27183.

Aohan Zeng, Xiao Liu, Zhengxiao Du, Zihan Wang, Hanyu Lai, Ming Ding, Zhuoyi Yang, Yifan Xu, Wendi Zheng, Xiao Xia, et al. 2022. Glm-130b:

An open bilingual pre-trained model. *arXiv preprint arXiv:2210.02414*.

Yichi Zhang, Bofei Gao, Tianyu Liu, Keming Lu, Wayne Xiong, Yue Dong, Baobao Chang, Junjie Hu, Wen Xiao, et al. 2024a. Pyramidkv: Dynamic kv cache compression based on pyramidal information funneling. *arXiv preprint arXiv:2406.02069*.

Ying Zhang, Peng Zhang, Mincong Huang, Jingyang Xiang, Yujie Wang, Chao Wang, Yineng Zhang, Lei Yu, Chuan Liu, and Wei Lin. 2024b. Qqq: Quality quattuor-bit quantization for large language models. *arXiv preprint arXiv:2406.09904*.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. 2023. Multilingual machine translation with large language models: Empirical results and analysis. *arXiv preprint arXiv:2304.04675*.