

# Compensate Quantization Errors: Make Weights Hierarchical to Compensate Each Other

Yifei Gao<sup>1</sup>, Jie Ou<sup>1</sup>, Lei Wang<sup>2\*</sup>, Yuting Xiao<sup>3</sup>, Zhiyuan Xiang<sup>4</sup>, Ruiting Dai<sup>1</sup>, Jun Cheng<sup>2</sup>

<sup>1</sup>University of Electronic Science and Technology of China

<sup>2</sup>Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences

<sup>3</sup>Beijing Normal University

<sup>4</sup>Tianjin University

yilei.jin123@gmail.com, lei.wang1@siat.ac.cn

## Abstract

Emergent Large Language Models (LLMs) use their extraordinary performance and powerful deduction capacity to discern from traditional language models. However, the expenses of computational resources and storage for these LLMs are stunning, quantization then arises as a trending conversation. To address accuracy decay caused by quantization, two streams of works in post-training quantization methods stand out. One uses other weights to compensate existing quantization error, while the other transfers the quantization difficulty to other parts in the model. Combining both merits, we introduce Learnable Singular value Increment (LSI) as an advanced solution. LSI uses Singular Value Decomposition to extract singular values of the weights and make them learnable to help weights compensate each other conditioned on activation. Incorporating LSI with existing techniques, we achieve state-of-the-art performance in diverse quantization settings, no matter in weight-only, weight-activation or extremely low bit scenarios. By unleashing the potential of LSI, efficient finetuning on quantized model is no longer a prohibitive problem.

## 1 Introduction

Large language models (LLMs) have garnered significant attention for their remarkable performance across a wide range of downstream tasks and their ability to exhibit emergent behavior (Bubeck et al., 2023; Touvron et al., 2023). Furthermore, their prowess in understanding natural language and deductive reasoning can be extended to multimodal domains through alignment training (Mu et al., 2023; Xu et al., 2023; Zhang et al., 2023). Nevertheless, the training and upkeep of such LLMs are highly resource-intensive, with many GPUs being able to support only a single model or parts of one. Quantization, as a central paradigm in this field,

emerges as a solution that addresses both memory footprint and computational challenges.

Quantization methods are typically divided into two categories based on the quantization period. Quantization-Aware Training (QAT) (Liu et al., 2023) involves tuning the model during training to optimize its compatibility with quantization. Although QAT can yield superior results compared to Post-Training Quantization (PTQ), the significant computational costs associated with the training process are a notable challenge. Consequently, PTQ methods have gained widespread acceptance and become increasingly popular in recent times.

Within the PTQ field, there are various methodologies to explore. For instance, the GPTQ series (Frantar and Alistarh, 2022; Frantar et al., 2023; Dettmers et al., 2023; Lee et al., 2023) employ unquantized weights to gradually offset the quantization errors introduced by previously quantized weights. In this paper, for the sake of convenience in reading, all references to "weight" pertain to the *weight matrices* in the linear layers of the model. They achieve this by solving a Lagrange equation to obtain a new Hessian matrix to update. On the other hand, methods like SmoothQuant (Xiao et al., 2023) and OmniQuant (Shao et al., 2023) focus on altering the distribution of weights and activations to mitigate the challenges of quantization. Meanwhile, through extensive research and experiments, it has been revealed that a significant portion of errors proposed during the quantization process are caused by a small number of outliers with distinctive weight values. Hence, several studies (Dettmers et al., 2023; Wei et al., 2022, 2023; Lee et al., 2023) concentrate on reducing or mitigating these outliers to minimize the disruption caused by their presence.

After thorough research, we have summarized the reasons for the success of previous methods. For the GPTQ series, in most cases, the weights obtained through manual calculation are not dif-

\*Corresponding author.

ferent from those obtained by the simple uniform quantization method, detailed in Sec 2.2. However, in some later stages of the quantization phase, GPTQ can alter the inherent hierarchy of the original weights, approximating the weights to other quantization intervals, to achieve a globally optimized solution. For methods like SmoothQuant, the concept of transferring the difficulty of weight quantization is relatively easy to understand. For example, if the equation  $1.3 \times 15.4 \approx 20$  is going to be quantized through rounding, quantizing 1.3 to 1 would require changing 15.4 to 20 to maintain the original value. However, if the two multipliers in the equation are scaled to  $4.3 \times 4.7 \approx 20$  and then quantized, it would only be necessary to quantize it to  $4 \times 5$  to keep the original value unchanged. In this way, the difficulty of quantization is greatly simplified. Both types of methods have their own advantages, but they cannot both enjoy the benefits of the other. Additionally, approaches like the GPTQ series have drawbacks such as long quantization times.

In this paper, we demonstrate that a good quantization method performs: (1) Transformation of the quantization difficulty of weights and activations; (2) Hierarchical change of some weights to fit the global optimum; (3) Data-free in the PTQ setting; (4) A small amount of quantization time consumption; (5) Inference efficiency (mixed-precision is not allowed). Based on the requirements mentioned above, we introduce a novel technique called *Learnable Singular value Increment* (LSI), which can effectively meet all the aforementioned requirements simultaneously. Unlike QAT, LSI exclusively focuses on training singular values of the weights, which constitute less than 0.1% of the total weights. Through the incorporation of a smoothing technique, LSI further simplifies the quantization problem. Moreover, existing methods primarily focus on aligning the performance of quantized models with unquantized models. However, with LSI, we have the capability to fine-tune quantized models without compromising the overall capabilities of these models themselves.

After conducting thorough experiments, we have achieved state-of-the-art results across a wide range of quantization settings, while marking a significant breakthrough in the field of quantized model fine-tuning. Our contributions encompass the following key points:

- Introduction of an innovative technique, LSI,

which promotes hierarchical organization of model weights, facilitating their adaptation to quantization settings without compromising the inherent capacity of the model.

- Integration of LSI with established smoothing techniques, effectively addressing the outlier issue and determining optimal transformation scales for quantization.
- Demonstrating the applicability of LSI in fine-tuning quantized models under few-shot conditions, with the fine-tuning results showing significant improvements.

## 2 Related Works

### 2.1 Quantization Methods

**Weight-Only Quantization.** Previously proposed methods have primarily focused on weight-only quantization, where the emphasis is on converting the weight matrices of the model into low-bit representations. This approach allows for significant reductions in computational resources when storing and distributing models. For instance, a model with 30 billion parameters can be stored using a memory of as little as 20GB. However, in this setup, a significant portion of quantization errors arises from high-magnitude activations, often referred to as outliers. Many works (Dettmers et al., 2023; Lee et al., 2023) have attempted to address this issue through mixed precision quantization while maintaining acceptable results. Nevertheless, this approach can introduce hardware inefficiencies and lead to increased inference time. Other methods, such as AWQ (Lin et al., 2023), employ more sophisticated scaling strategies, dividing weights into different channels, and exclusively incorporating quantization scales and zero points.

**Weight-Activation Quantization.** Weight-activation quantization involves the quantization of both model weights and activations. In the widely used self-attention setting, reducing the activation precision from 16 bits to a lower level results in a significant memory and time improvement, often of a squared magnitude.

In many cases, residual networks are interposed between layers, maintaining uniform and flat weights but leading to imbalanced and occasionally polarized activation. The quantization of activation is notably more challenging than that of weights, primarily because the outliers in activation

are often substantially different from typical weight values.

## 2.2 Quantization Techniques

**Uniform Quantization.** Although non-uniform quantization usually outperforms uniform quantization, it falls short in the dependency of some specialized devices (Guo et al., 2022). In contrast, uniform quantization is a more practical and feasible approach that can be efficiently executed on regular hardware devices. Uniform quantization discretizes a high-precision value into low-bit levels. Typically, we use the uniform quantization function  $Q$  to transform a float linear weight matrix  $W$  (commonly used in LLMs) to  $k$  bits integer matrix  $\tilde{W}$  as follows:

$$\tilde{W} = Q(W, k, s_h, z) = \text{Clamp}(\lfloor \frac{W}{s_h} \rfloor + z, 0, 2^k - 1), \quad (1)$$

where  $s_h$  and  $z$  are corresponding shift and zero-point, respectively.

**Smooth.** The smooth technique is a method that involves transforming weights or activations by transferring certain magnitudes between them while maintaining their mathematical equivalence. Given a magnitude factor represented as a scaling matrix  $\text{diag}(s_c)$ , activations  $\mathbf{X}$ , and the final output  $\mathbf{Y}$ , the transformation can be driven as:

$$\mathbf{Y} = \mathbf{X}\mathbf{W} + \mathbf{B} = \underbrace{[(\mathbf{X} - \delta) \oslash s_c]}_{\tilde{\mathbf{X}}} \cdot \underbrace{[s_c \odot \mathbf{W}]}_{\tilde{\mathbf{W}}} + \underbrace{[\mathbf{B} + \delta\mathbf{W}]}_{\tilde{\mathbf{B}}} \quad (2)$$

where  $\tilde{\mathbf{X}}$ ,  $\tilde{\mathbf{W}}$  and  $\tilde{\mathbf{B}}$  are equivalent activation, weight, and bias, respectively.  $\oslash$  and  $\odot$  are elementwise division and multiplication, respectively.

## 3 Our Methods

### 3.1 Inspiration

Our inspiration is drawn from the idea of compressing images into smaller sizes, where Singular Value Decomposition (SVD) is employed, and from the training method LoRA (Hu et al., 2021), which focuses on training sub-matrices rather than the entire weight matrix. In the context of quantization, we can perceive a quantized model as a training objective after quantization has been applied to the original model. The goal of the training process (quantization process) is to make the linear weights in the model discrete enough to meet the specified n-bit quantization setting. Building upon this concept, our goal is to enable the model parameters to autonomously adapt to the quantization

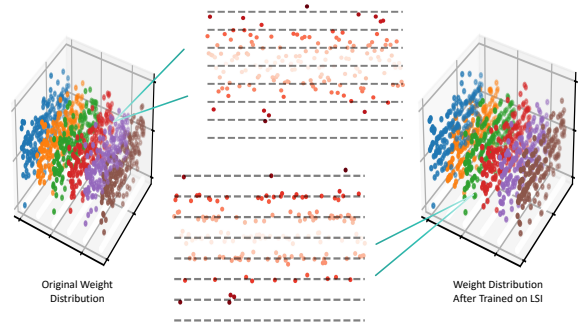


Figure 1: **Weight Distribution Comparison** between original weights and weights trained after LSI.

process in a data-independent condition. In other words, we aim to have the weights automatically adjust to an appropriate magnitude that aligns with the specified n-bit setting without training on large datasets. We believe that by achieving this level of self-adjustment, we can attain better results in the context of quantization.

However, irrespective of how meticulously we design the quantization strategy, quantization errors are an inherent part of the process. It is crucial to recognize that not all differences resulting from quantization errors are detrimental. As mentioned in GPTQ (Frantar et al., 2023), when group-wise updates are made to the Hessian matrix during quantization, performance is nearly equivalent to updating the Hessian matrix one element at a time, as errors tend to compensate for each other throughout the procedure.

Incorporating both of these fundamental insights, the Learnable Singular Value Increment (LSI) method emerges as a solution to address both training and performance issues in the context of quantization. It is noteworthy that our approach differs fundamentally from previous methods in the insight into quantization. While other methods primarily aim to reduce quantization errors, our approach makes use of these errors as a constructive element in the quantization process. We introduce weight disturbance by LSI to directly cause some "errors" to help the linear weights change their original magnitude to get the global optimum while not disturbing the original distribution that much. During quantization, LSI organizes weights hierarchically, grouping them into sets that closely resemble the specified discrete weight values, as presented in Fig. 1.

### 3.2 Learnable Singular Value Increment (LSI)

We now drive LSI optimization beginning with SVD. For a single linear layer, a weight matrix  $\mathbf{W}^{a \times b}$  (we default  $a \times b$ ) can be decomposed into three sub-matrices,  $\mathbf{U}^{a \times a}$ ,  $\mathbf{S}^a$  (only contain diagonal weights),  $\mathbf{V}_h^{a \times b}$ . Combining with uniform quantization, we assume that when we fix  $\mathbf{U}$  and  $\mathbf{V}_h$ , there exists an optimal sub-matrix  $\mathbf{S}'$  (full matrix irrelevant to SVD) after QAT-like training. So we can get

$$\tilde{\mathbf{W}} = \mathbf{U} \odot \mathbf{S}' \odot \mathbf{V}_h, \quad (3)$$

which can largely satisfy the quantization setting and result in the least quantization error throughout the input. It means that given input  $\mathbf{X}$ , the output  $\mathbf{Y}$ , the minimum quantization error  $\mathbf{E}_{min}$  conditioned on  $\mathbf{S}'$  can be expressed as:

$$\mathbf{E}_{min} = \mathbf{Y} - \mathbf{X} \odot \tilde{\mathbf{W}} = \mathbf{Y} - \mathbf{X} \odot \mathbf{U} \odot \mathbf{S}' \odot \mathbf{V}_h. \quad (4)$$

In the equation provided above, once we identify the minimum error  $\mathbf{E}_{min}$  and fix the quantization method  $\mathbf{Q}$  along with variables  $s$  and  $z$ , we can make  $\mathbf{S}'$  solvable. However, It is important to note that uniform quantization alone does not yield the optimal solution. Our goal is to introduce slight changes to the weight distribution, making the weights hierarchical to better conform to the uniform quantization setting. The change should be aware of the input since not all weights share the same importance during inference.

Training the entire weight matrix is computationally intensive, and not all errors will have a negative impact, as mentioned previously. In our method, LSI introduces an additional variable called **Learnable Singular value Increment  $\mathbf{I}'$**  that is added on the original singular value to slightly change the weight distribution of original weight  $\mathbf{W}$ . In this setting, the quantized weight  $\tilde{\mathbf{W}}$  can be obtained by:

$$\begin{aligned} & Q(\mathbf{W}, k, s_h, z, \mathbf{I}') \\ &= \text{Clamp}(\lfloor \frac{\mathbf{U} \odot \text{diag}(\mathbf{S} + \mathbf{I}') \odot \mathbf{V}_h}{s_h} \rfloor + z, 0, 2^k - 1), \\ \tilde{\mathbf{W}} &= (Q(\mathbf{W}, k, s_h, z, \mathbf{I}') - z) s_h. \end{aligned} \quad (5)$$

And we optimize the corresponding equation to find the optimal  $\mathbf{I}'$  with linear function  $\mathcal{F}$ :

$$\arg \min_{\mathbf{I}'} \|\mathcal{F}(\mathbf{W}, \mathbf{X}) - \mathcal{F}(\tilde{\mathbf{W}}, \mathbf{X})\|. \quad (6)$$

Otherwise, if we shift our focuses on  $\mathbf{S}'$  (the optimal quantization sub-matrix),  $\mathbf{I}'$  is to make:

$$\arg \min_{\mathbf{I}'} \|\mathbf{U} \odot \mathbf{S}' \odot \mathbf{V}_h - Q(\mathbf{W}, k, s_h, z, \mathbf{I}') - z) s_h\|. \quad (7)$$

$\mathbf{I}'$  is typically represented as a 1-D matrix, we cannot solely rely on LSI to perfectly align and significantly reduce errors. Instead, our approach deliberately introduces  $\mathbf{I}'$  as "errors" to facilitate minor disturbances in original weight distribution to compensate for each other from a global perspective. It's noteworthy that even if we train the quantized model layer by layer, our method can still largely achieve globally optimal results in the end. Due to the introduction of weight perturbation, later layers can largely compensate for the errors missed by earlier layers, which can be seen in Table 7. This simplifies the training procedure while working towards the desired goal of error compensation.

In our experiments, however, we observed that for the group-wise setting, where different and delicate scaling scales are applied to various groups, LSI alone faces challenges in learning a set of optimal parameters that can balance all groups effectively. We will further discuss this issue in Sec 4.4. To address this issue, we introduced an additional small square matrix, around the dimensions of  $100 \times 100$  to  $600 \times 600$ , adding at the beginning of the diagonal matrix, specifically on the first  $n$  rows and the first  $n$  columns. This addition is made once the  $\text{diag}(\mathbf{S} + \mathbf{I}')$  component has been computed in Equation 5. We believe that the most prominent values are influenced by the most significant singular values, while the relatively smaller singular values have less impact on high-magnitude values. Therefore, the introduction of this additional square matrix helps achieve a better balance in the group-wise setting.

### 3.3 Smooth and Clipping

Techniques like *Smooth* and *Clipping* are powerful methods that find extensive use in various applications. *Smooth* is effective at transferring quantization challenges to make the quantization of weights more manageable, while *Clipping* is instrumental in addressing outlier issues. By incorporating these techniques with recent advancements, such as *Learnable Weight Clipping (LWC)* and *Learnable Equivalent Transformation (LET)* proposed in OmniQuant (Shao et al., 2023), LSI can significantly reduce quantization errors and achieve remarkable performance gains.

In the linear layer, LET is to make  $\text{diag}(s_c)$  in Eq. 2 learnable. Additionally, in the attention operation, LET introduces learnable parameter  $\text{diag}(s_a)$  to smooth the query  $\mathbf{Q}$  and key  $\mathbf{K}$ , which



can be represented as:

$$\begin{aligned} \mathbf{P} &= \text{Softmax}(\mathbf{Q}\mathbf{K}^T) \\ &= \text{Softmax}\left(\underbrace{(\mathbf{Q} \oslash s_a)}_{\hat{\mathbf{Q}}}\underbrace{(s_a \odot \mathbf{K}^T)}_{\hat{\mathbf{K}}^T}\right). \end{aligned} \quad (8)$$

On the other hand, LWC is to make upper and lower boundaries in quantization function  $Q$  learnable in Eq. 1 instead of fixed  $2^k$ .

## 4 Experiments

### 4.1 Settings

**Quantization.** In line with the methodology outlined in (Shao et al., 2023), our experiments cover both weight-only and weight-activation quantization settings. For the weight-only component, we employ channel-wise weight quantization at INT4/INT3/INT2 bit levels. In the weight-activation setting, we utilize quantization settings of w6a6/w4a4, where ‘w’ and ‘a’ signify *weight* and *activation*, respectively. In cases where groups are divided, with each group having a distinct set of quantization parameters, we use ‘g’ to represent the group name. Furthermore, we adhere to the original setup, keeping the Softmax part in float32, as this helps mitigate excessive disturbance caused by self-attention layers during the inference process. We also inherit the enhanced acceleration nature of OmniQuant in INT3/INT2 settings on CUDA. Please see (Shao et al., 2023) for more details.

**Training.** Given that LSI effectively satisfies both smoothing and shifting techniques, we initialize the scaling and shifting parameters using well-trained parameters from (Shao et al., 2023). We then train a set of LSI parameters based on this initial setup. Singular values can introduce significant variations in the distribution of weights, so we maintain a low learning rate at  $2e-4$ . We employ the AdamW (Loshchilov and Hutter, 2019) optimizer with a weight decay of 0 to optimize our parameters. All the data used in our training was collected from WikiText2 (Merity et al., 2017). Notably, the training process is quite fast, with larger models requiring fewer epochs. For instance, in the ‘w4a16g128’ setting, the *OPT-30B* model only needs to be trained for 2 epochs on a dataset with 32 samples. All techniques proposed before were included during the whole quantization procedures. Additionally, without groups, we all set square matrix dimension  $n = 200$ . But with group-wise scaling, we test several dimensions to validate the effectiveness of the increment square matrix, which

we will discuss in Sec 4.4. For the finetuning, we select PTB (Marcus et al., 1994), where the perplexity of our baselines on it is significantly higher than others. We only train the last two layers of models with epochs around 10 to 40 on 128 PTB samples, which is very fast to implement.

**Models.** We conduct evaluation on two popular baselines for generalization, LLaMA(7-30B) (Touvron et al., 2023), OPT(125M-66B) (Zhang et al., 2022). For more details about our test results, please see the supplementary materials.

**Evaluation.** Our evaluation for perplexity is mainly focused on WikiText2 (Merity et al., 2017), PTB (Marcus et al., 1994), C4 (Raffel et al., 2020). Furthermore, following previous works, we also evaluate several zero-shot tasks in weight-activation quantization setting, including PIQA (Bisk et al., 2020), ARC (Clark et al., 2018), and HellaSwag (Clark et al., 2018). Samples of datasets we evaluated obey the GPTQ (Frantar et al., 2023) settings. For accuracy tasks, lm-eval-harness (Gao et al., 2021) is employed for all zero-shot tasks.

**Baselines.** For weight-only quantization, we choose previously state-of-the-art works, GPTQ (Frantar et al., 2023), AWQ (Lin et al., 2023) and recently advanced work OmniQuant (Shao et al., 2023) as baselines. For weight-activation quantization, both QAT and PTQ methods are included, containing SmoothQuant (Xiao et al., 2023), RPTQ (Yuan et al., 2023), QAT (Liu et al., 2023) and Omniquant (Shao et al., 2023). Following SmoothQuant (Xiao et al., 2023), we do not change the per-channel quantization strategy for weights and the per-tensor quantization strategy for activation.

### 4.2 Weight-only Quantization Results

In this section, we mainly demonstrate the results of the OPT series without group-wise scaling in w3a16 and w4a16 settings on WikiText2, as shown in Table 1, while w3a16 and w4a16 group-wise results are shown in Sec 4.4. Our full results can be found in the supplementary material. As exhibited in these tables, prominent progress can be seen in various settings. LSI provides effective and powerful solutions for the quantization of smaller LLMs and w2 settings while helping further improve the performance in more sophisticated settings.

Table 1: WikiText2 perplexity of Weight-only quantization results in OPT models.

OPT / PPL↓		125M	1.3B	2.7B	6.7B	13B	30B	66B
FP16	-	27.65	14.63	12.47	10.86	10.12	9.56	9.34
W2A16-g128	GPTQ (Frantar et al., 2023)	597.66	115.16	61.59	20.18	21.36	12.71	82.10
	AWQ (Lin et al., 2023)	251.84	47.97	28.50	16.20	14.32	12.31	14.54
	OmniQuant (Shao et al., 2023)	75.43	23.95	18.13	14.43	12.94	11.39	30.84
	<b>Ours</b>	<b>56.17</b>	<b>22.59</b>	<b>17.65</b>	<b>14.23</b>	<b>12.75</b>	<b>11.30</b>	<b>29.66</b>
W2A16-g64	GPTQ (Frantar et al., 2023)	204.40	49.58	29.37	16.81	16.65	11.87	356.01
	AWQ (Lin et al., 2023)	124.18	29.78	20.64	14.63	13.28	11.59	12.74
	OmniQuant (Shao et al., 2023)	62.56	21.40	16.76	13.57	12.33	11.00	10.59
	<b>Ours</b>	<b>50.94</b>	<b>21.09</b>	<b>16.69</b>	<b>13.51</b>	<b>12.25</b>	<b>10.95</b>	<b>10.56</b>
W3A16	GPTQ (Frantar et al., 2023)	53.05	21.17	16.83	15.09	11.73	10.30	14.42
	AWQ (Lin et al., 2023)	69.43	28.01	263.10	15.13	20.09	35.74	4.5e3
	OmniQuant (Shao et al., 2023)	35.66	16.68	13.80	11.65	10.87	10.00	9.83
	<b>Ours</b>	<b>32.19</b>	<b>16.24</b>	<b>13.44</b>	<b>11.46</b>	<b>10.66</b>	<b>9.96</b>	<b>9.79</b>
W4A16	GPTQ (Frantar et al., 2023)	31.43	15.56	12.82	11.41	10.31	9.63	9.55
	AWQ (Lin et al., 2023)	32.28	15.49	12.93	11.30	10.39	9.77	9.61
	OmniQuant (Shao et al., 2023)	29.45	15.04	12.76	11.03	10.30	9.65	9.65
	<b>Ours</b>	<b>28.86</b>	<b>15.00</b>	<b>12.71</b>	<b>11.00</b>	<b>10.24</b>	<b>9.63</b>	<b>9.50</b>

Table 2: Weight-activation quantization results of OPT Models. We report perplexity on three datasets: WikiText2 (WIKI), Pen Treebank (PT), and C4. RPTQ indicates the data from RPTQ ((Yuan et al., 2023)) paper, which keeps the output of LN and SoftMax as 8-bit. RPTQ\* represents reproducing RPTQ with our setting that quantizes all activation into low-bit except keeping the softmax output at full precision. OPT-66B results can be found in supplementary material.

OPT / PPL↓		OPT-6.7b			OPT-13b			OPT-30b		
Task		WIKI	PT	C4	WIKI	PT	C4	WIKI	PT	C4
FP16	-	10.86	13.09	11.74	10.13	12.34	11.20	9.56	11.84	10.69
W6A6	SmoothQuant (Xiao et al., 2023)	11.34	13.82	12.14	10.56	12.76	11.40	9.67	12.01	10.81
	RPTQ (Yuan et al., 2023)	11.19	13.98	12.08	11.00	15.23	11.68	10.22	14.95	11.73
	RPTQ*	10.96	13.24	11.86	10.25	12.60	11.31	9.60	12.23	10.83
	OmniQuant (Shao et al., 2023)	10.96	13.20	11.81	10.21	12.47	11.27	9.62	<b>11.92</b>	10.76
	<b>Ours</b>	<b>10.91</b>	<b>13.19</b>	<b>11.80</b>	<b>10.19</b>	<b>12.45</b>	<b>11.27</b>	<b>9.60</b>	<b>11.93</b>	<b>10.75</b>
W4A4	SmoothQuant (Xiao et al., 2023)	1.8e4	1.4e4	1.5e4	7.4e3	6.5e3	5.6e3	1.2e4	7.8e3	8.3e3
	RPTQ (Yuan et al., 2023)	12.00	15.17	12.85	12.74	15.76	14.71	11.15	14.11	13.48
	RPTQ*	17.83	25.10	19.91	16.45	23.01	16.80	11.50	14.87	12.81
	OmniQuant (Shao et al., 2023)	12.24	15.54	13.56	11.65	15.89	13.46	10.60	13.75	11.89
	<b>Ours</b>	<b>11.82</b>	<b>14.86</b>	<b>13.10</b>	<b>11.10</b>	<b>15.16</b>	<b>12.81</b>	<b>10.29</b>	<b>13.32</b>	<b>11.64</b>

### 4.3 Weight-Activation Quantization Results

In the context of weight-activation quantization, we have successfully improved several metrics over the original OmniQuant (Shao et al., 2023), as presented in Table 2, and achieved enhanced performance across various tasks with the LLaMA families, as shown in Table 3. Specifically, in W6A6 settings, we generally observe slightly better results, and in W4A4 settings, we significantly outperform existing methods.

### 4.4 Ablation Study

In our extensive ablation studies, we investigated the effectiveness of LSI and the impact of the additional increment square matrix. Table 4 indicates that LSI is remarkably beneficial when the model size is relatively small, and adjusting the value of  $k$  indeed provides certain advantages. However, as the volume of the models increases, particularly

with the influence of group-wise scaling, the benefits brought by LSI and its corresponding matrix diminish, and can even facilitate the overfitting problem as observed in Table 5. Meanwhile, the introduction of the adding matrix can result in some affinity to some specific datasets, which may be a kind of overfitting problem.

We posit that this diminishing impact might be attributed to group-wise scaling, which discretizes the entire weight matrix into different parts, resulting in incoherence within the weight matrix. This implies that the compensation for quantization error is restricted to individual groups, disregarding the integrated nature of the entire matrix. Considering the coherence of the weight matrix, singular values influence the distribution of the entire weight matrix rather than a singular part.

On the other hand, LSI alone can achieve competitive performance, however, it suffers severely

Table 3: **Weight-activation quantization results of LLaMA Models.** We report our perplexity results on WikiText2 and C4, along with accuracy (for multi-choices tasks, we report our accuracy norm) of 6 zero-shot tasks compared with other baselines.(In W6A6 settings, layers in LLaMA can not train on LSI somehow)

LLaMA / Acc $\uparrow$	#Bits	Method	Wiki	C4	PIQA	ARC-e	Arc-c	HS <sup>1</sup>	WG <sup>2</sup>	Avg.
LLaMA-7B	FP16	-	-	-	77.47	52.48	41.46	73.00	67.07	62.30
	W4A4	SmoothQuant (Xiao et al., 2023)	-	-	49.80	30.40	25.80	27.40	48.00	36.28
	W4A4	LLM-QAT	-	-	51.50	27.90	23.90	31.10	51.90	37.26
	W4A4	LLM-QAT+SQ	-	-	55.90	35.50	26.40	47.80	50.60	43.26
	W4A4	OmniQuant (Shao et al., 2023)	11.26	14.51	66.15	45.20	31.14	56.44	53.43	50.47
	<b>W4A4</b>	<b>Ours</b>	<b>11.02</b>	<b>13.77</b>	<b>67.90</b>	<b>47.43</b>	<b>31.91</b>	<b>57.51</b>	<b>56.27</b>	<b>52.20</b>
LLaMA-13B	FP16	-	-	-	79.10	59.89	44.45	76.21	70.31	65.99
	W4A4	SmoothQuant (Xiao et al., 2023)	-	-	61.04	39.18	30.80	52.29	51.06	46.87
	W4A4	OmniQuant (Shao et al., 2023)	10.87	13.78	69.69	47.39	33.10	58.96	55.80	53.05
	W4A4	<b>Ours</b>	<b>10.68</b>	<b>12.84</b>	<b>69.69</b>	<b>47.43</b>	<b>33.61</b>	<b>62.37</b>	<b>59.59</b>	<b>54.54</b>
LLaMA-30B	FP16	-	-	-	80.08	58.92	45.47	79.21	72.53	67.24
	W4A4	SmoothQuant (Xiao et al., 2023)	-	-	58.65	35.53	27.73	35.56	48.06	41.11
	W4A4	OmniQuant (Shao et al., 2023)	10.33	12.49	71.21	49.45	34.47	64.65	59.19	55.79
	W4A4	<b>Ours</b>	<b>10.20</b>	<b>12.12</b>	<b>72.90</b>	<b>49.45</b>	<b>36.43</b>	<b>65.98</b>	<b>60.22</b>	<b>57.00</b>

<sup>1</sup> ‘HS’ stands for HellaSwag.

<sup>2</sup> ‘WG’ stands for WinoGrande.

Table 4: **Weight-only** quantization results of OPT Models (125m-2.7b) in W3A16g128 and W4A16g128 settings with different  $k$ .

OPT / PPL $\downarrow$		OPT-125m			OPT-1.3b			OPT-2.7b		
Task		WIKI	PT	C4	WIKI	PT	C4	WIKI	PT	C4
FP16	-	27.65	32.54	24.60	14.63	16.96	14.72	12.47	15.11	13.16
W3A 16g128	GPTQ (Frantar et al., 2023)	39.24	45.17	30.08	16.47	19.90	16.47	13.69	17.06	14.54
	AWQ (Lin et al., 2023)	36.74	44.07	30.39	16.32	19.59	16.27	13.58	16.52	14.19
	OmniQuant (Shao et al., 2023)	32.25	40.76	29.34	15.72	19.06	16.11	13.18	16.29	<b>14.15</b>
	<b>Ours k100</b>	<b>31.63</b>	<b>40.74</b>	<b>29.21</b>	<b>15.68</b>	<b>18.99</b>	<b>16.11</b>	<b>13.17</b>	<b>16.27</b>	14.17
	<b>Ours k200</b>	<b>31.06</b>	<b>39.84</b>	<b>28.78</b>	<b>15.64</b>	<b>18.95</b>	<b>16.09</b>	<b>13.15</b>	<b>16.27</b>	14.16
W4A 16g128	GPTQ (Frantar et al., 2023)	29.81	35.48	25.96	14.89	17.41	15.05	<b>12.52</b>	15.42	13.40
	AWQ (Lin et al., 2023)	29.15	34.95	25.90	14.94	17.46	15.04	12.74	15.33	13.39
	OmniQuant (Shao et al., 2023)	28.86	34.28	25.63	14.88	<b>17.40</b>	15.03	12.65	15.28	13.38
	<b>Ours k100</b>	<b>28.57</b>	<b>33.68</b>	<b>25.51</b>	<b>14.87</b>	17.42	<b>15.02</b>	12.64	<b>15.26</b>	<b>13.38</b>
	<b>Ours k200</b>	<b>28.40</b>	34.21	<b>25.45</b>	<b>14.85</b>	17.44	<b>15.02</b>	<b>12.62</b>	15.28	<b>13.38</b>

from bias. Without the transformation of quantization difficulty, LSI obtains relatively good performance through significant overfitting on a specific dataset, as seen in Table 6.

#### 4.5 Finetuning of LSI

As illustrated in Sec 4.4, LSI alone has a grave problem of overfitting. After investigations, we find that LSI can help bridge the gaps caused by previous layers, which means that even only employing LSI in the last several layers, there are still some overfitting problems. However, if we use this property to quickly finetune a model on a specific dataset, this weakness turns into an advantage. In finetuning part, we first employ LSI to change the original weight distribution and then transfer the quantization difficulty using smooth techniques. Through our experiments, in nearly all settings, employing LSI only on the last several layers of a LLM can also result in improved performance on a specific dataset without largely

compromising other abilities, as shown in Table 7. To test the generalization of LSI finetuning, we prepare *OmniQuant* and *Round To Nearest (RTN)* as baseline quantization strategies and only employ LSI in the last two layers. LSI can also satisfy other baselines, as we have tried to use GPTQ baselines and replace the last several layers quantized with our techniques, it also works well.

#### 4.6 Other Issues

**Inference Speed.** Overall, our method does not introduce additional inference time. Because after training, we integrate LSI into the original weights to alter them, and then quantize them to the specified bit. For the introduced smooth technology, only LWC will cause a minor delay in inference. However, for LET, the transfer of weight scaling in LET is integrated into the norm function of each layer. After training, it directly scales the gain in the Layernorm function of the original model, so there will be no impact during inference. For de-

Table 5: **Weight-only** quantization results of OPT Models (6.7m-30b) in W3A16g128 and W4A16g128 settings. Here, we adopt  $k = 320$  in OPT-6.7b,  $k = 450$  in OPT-13b and  $k = 600$  in OPT-30b

OPT / PPL↓		OPT-6.7m			OPT-13b			OPT-30b		
Task		WIKI	PT	C4	WIKI	PT	C4	WIKI	PT	C4
FP16	-	10.86	13.08	11.74	10.12	12.33	11.19	9.56	11.84	10.69
W3A 16g128	GPTQ (Frantar et al., 2023)	11.65	14.24	12.48	10.35	12.84	11.58	9.73	12.54	10.91
	AWQ (Lin et al., 2023)	11.41	13.98	12.30	10.68	12.87	11.61	9.85	66.68	10.96
	OmniQuant (Shao et al., 2023)	11.27	13.77	12.31	10.47	12.96	11.63	9.79	12.19	10.98
	<b>Ours k200</b>	11.26	13.77	12.31	10.45	12.94	11.63	9.79	12.17	10.98
	<b>Ours k320-600</b>	11.26	13.76	12.31	10.45	12.95	11.62	9.76	12.19	10.98
W4A 16g128	GPTQ (Frantar et al., 2023)	10.93	13.21	11.87	11.26	12.42	12.46	9.58	11.89	10.74
	AWQ (Lin et al., 2023)	10.93	13.28	11.87	10.21	12.46	11.28	9.59	11.90	10.75
	OmniQuant (Shao et al., 2023)	10.96	13.25	11.85	10.20	12.46	11.29	9.62	11.94	10.75
	<b>Ours k200</b>	10.95	13.25	11.85	10.19	12.47	11.29	9.61	11.95	10.74
	<b>Ours k320-600</b>	10.94	13.24	11.85	10.19	12.46	11.29	9.61	11.93	10.75

Table 6: **LSI-only** quantization results

PPL		LLaMA-7b		OPT-2.7b			OPT-6.7b		
Task		WIKI	C4	WIKI	PT	C4	WIKI	PT	C4
W3A16g128	GPTQ (Frantar et al., 2023)	6.55	7.85	13.69	17.06	14.54	11.65	14.24	12.48
	<b>Ours</b>	6.25	7.91	13.70	17.35	14.82	11.75	14.87	13.05
W4A16	GPTQ (Frantar et al., 2023)	6.13	7.43	12.82	15.94	13.75	11.41	13.75	12.15
	<b>Ours</b>	5.95	7.47	12.76	16.15	14.06	11.27	13.93	12.33

Table 7: **Finetuning** by LSI

PPL		LLaMA-7b		
Task		PT	WIKI	C4
W3A 16g128	RTN	37.37	7.01	8.62
	<b>RTN w/ LSI</b>	35.58	6.91	8.52
	OmniQuant (Shao et al., 2023)	33.45	6.15	7.75
	<b>Omni w/ LSI</b>	30.69	6.16	7.77
PPL		LLaMA-30b		
W4A16	RTN	17.15	4.57	6.34
	<b>RTN w/ LSI</b>	17.06	4.55	6.32
	OmniQuant (Shao et al., 2023)	16.48	4.25	6.11
	<b>Omni w/ LSI</b>	16.46	4.26	6.12

tailed inference speeds, one can refer to (Shao et al., 2023). Our inference speed is essentially identical to theirs.

**Best Results.** Our best results were not obtained by using OmniQuant’s parameters as the initialization. During the training process, we found that random initialization followed by a longer training period could potentially yield better results. However, due to the instability of random initialization and the loss incurred by the extended quantization time, we did not use random initialization in our experiments.

## 5 Limitations

### 5.1 Overfitting Problem

Despite the introduction of additional parameters constituting an extremely small portion compared to the overall parameter volume, their significance is substantial. Aligned with our philosophy, the re-distribution of model weights should be cognizant

of activations, which leads to the overfitting problem. As demonstrated in our supplementary material, there is a trade-off between achieving improvements in perplexity on one dataset and a cost associated with the other.

In our experiments, we observed that LSI can indeed significantly reduce the loss caused by quantization. However, the elimination of loss does not always guarantee a corresponding performance improvement. This phenomenon is further substantiated through the training procedure, as discussed in our supplementary material.

Generally speaking, LSI tends to achieve superior results by largely aligning with one dataset. In the early stages, when the overall error is substantial, conforming to one dataset can substantially enhance coherence and restore the original capacity of the model. However, as errors gradually diminish, a boundary is encountered. Given that errors are unavoidable in quantization settings, beyond this boundary, LSI exhibits overfitting problems.

### 5.2 Hard to Train

As discussed in Sec 5.1, there is a boundary in training. But for different models with different volumes, these boundaries do not display a stable paradigm, so it is needed to train with different epochs to gradually get closer to the optimal performance. But in general, with model volume growing, less training is needed. For example, in the W4A16 setting, 5 epochs are needed to train



the *OPT-13B* on a 128-sample dataset, while only 1 epoch is needed for *OPT-30B* on a 32-sample dataset, which takes less than 1.5 hours. For more details, please see our supplementary material.

On the other hand, when implementing only LSI, the entire process becomes somewhat precarious. As our goal is to align the quantized weight distribution, LSI can assist weights in stepping over their original magnitude span. In this process, significant fluctuations are quite common, and even with the incorporation of smoothing techniques, this phenomenon cannot be entirely avoided.

## 6 Conclusion

We introduce LSI to adjust model weights to conform to quantization settings. Through integration with established techniques, our approach attains state-of-the-art performance across diverse quantization settings. LSI imparts a hierarchical structure to model weights, enhancing adaptability to quantization parameters without compromising training efficiency. Leveraging attributes of LSI allows effective finetuning of a quantized model on various datasets. Notably, during inference, LSI introduces no additional parameters and preserves the hardware efficiency inherited from OmniQuant.

## Acknowledgements

This work was supported in part by the Guangdong Major Project of Basic and Applied Basic Research (2023B0303000016), Shenzhen Technology Project (JCYJ20220531095810023), National Natural Science Foundation of China (U21A20487, 61976143), CAS Key Technology Talent Program, Colleges and Universities Key Laboratory of Intelligent Integrated Automation (No.201502), CAS Key Laboratory of Human-Machine Intelligence-Synergy Systems, Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences (2014DP173025), Guangdong-Hong Kong-Macao Joint Laboratory of Human-Machine Intelligence-Synergy Systems (2019B121205007).

## References

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar,

Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. 2023. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefer, and Dan Alistarh. 2023. Spqr: A sparse-quantized representation for near-lossless llm weight compression. *arXiv preprint arXiv:2306.03078*.

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, Saleh Ashkboos, Torsten Hoefer, and Dan Alistarh. 2023. Gptq: Accurate post-training quantization for generative pre-trained transformers.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, et al. 2021. A framework for few-shot language model evaluation. *Version v0. 0.1. Sept*.

Cong Guo, Chen Zhang, Jingwen Leng, Zihan Liu, Fan Yang, Yunxin Liu, Minyi Guo, and Yuhao Zhu. 2022. Ant: Exploiting adaptive numerical data type for low-bit deep neural network quantization. In *2022 55th IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 1414–1433. IEEE.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

Changhun Lee, Jungyu Jin, Taesu Kim, Hyungjun Kim, and Eunhyeok Park. 2023. Owq: Lessons learned from activation outliers for weight quantization in large language models. *arXiv preprint arXiv:2306.02272*.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. 2023. Awq: Activation-aware weight quantization for llm compression and acceleration. *arXiv preprint arXiv:2306.00978*.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2023. Llm-qat: Data-free quantization aware training for large language models. *arXiv preprint arXiv:2305.17888*.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *The International Conference on Learning Representations (ICLR)*.

- Mitch Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. 1994. The penn treebank: Annotating predicate argument structure. In *Human Language Technology: Proceedings of a Workshop held at Plainsboro, New Jersey, March 8-11, 1994*.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2017. Pointer sentinel mixture models.
- Yao Mu, Qinglong Zhang, Mengkang Hu, Wenhai Wang, Mingyu Ding, Jun Jin, Bin Wang, Jifeng Dai, Yu Qiao, and Ping Luo. 2023. Embodiedgpt: Vision-language pre-training via embodied chain of thought. *arXiv preprint arXiv:2305.15021*.
- 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. *The Journal of Machine Learning Research*, 21(1):5485–5551.
- 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*.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.
- 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*.
- 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.
- 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.
- Peng Xu, Wenqi Shao, Kaipeng Zhang, Peng Gao, Shuo Liu, Meng Lei, Fanqing Meng, Siyuan Huang, Yu Qiao, and Ping Luo. 2023. Lvlm-ehub: A comprehensive evaluation benchmark for large vision-language models. *arXiv preprint arXiv:2306.09265*.
- Zhihang Yuan, Lin Niu, Jiawei Liu, Wenyu Liu, Xinggang Wang, Yuzhang Shang, Guangyu Sun, Qiang Wu, Jiayang Wu, and Bingzhe Wu. 2023. Rptq: Reorder-based post-training quantization for large language models. *arXiv preprint arXiv:2304.01089*.
- Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.
- Yiyuan Zhang, Kaixiong Gong, Kaipeng Zhang, Hongsheng Li, Yu Qiao, Wanli Ouyang, and Xiangyu Yue. 2023. Meta-transformer: A unified framework for multimodal learning. *arXiv preprint arXiv:2307.10802*.

## A Appendix

In this appendix, we provide further details as follows:

- Sec. A.1: Training procedure analysis of the local sensitivity information, and when optimal results can be achieved using **Learnable Single value Increment (LSI)**.
- Sec. A.2: Showcases the complete results for OPT, LLaMA-1 models.

### A.1 Training Procedure Analysis

In this section, our focus centers on elucidating the training procedure for our techniques. Fig 2 illustrates that exhaustive training does not consistently yield the optimal outcome. Notably, when trained on OPT-30B, a mere 32 samples suffice; however, even slight deviations within this range can induce noteworthy disturbances in performance. This poses difficulties in getting the optimal parameter.

LSI proves highly effective in mitigating quantization loss. Generally, LSI can achieve a reduction ranging from 20% to 40%, surpassing the performance of OmniQuant. Nevertheless, it’s crucial to note, as emphasized in our paper, that the reduction in loss does not necessarily translate to improved overall performance.

### A.2 Full Results

In this section, we provide a comprehensive presentation of our results across various datasets to complement the main paper. Specifically, the results include:

- OPT-66B results on W4A4 setting (Table 8).
- Wiki perplexity with weight-only quantization in the LLaMA families (Table 9).

Table 8: **Weight-activation quantization results** of OPT-66B. We test the results that only use LSI in the last several layers, and  $L$  refers to the layer number implemented LSI. RPTQ\* represents reproducing RPTQ with our setting that quantizes all activation into low-bit except keeping the softmax output at full precision.

OPT / PPL ↓		OPT-66b		
Task		WIKI	PT	C4
FP16	-	9.34	11.36	10.28
W4A4	SmoothQuant (Xiao et al., 2023)	2.2e5	1.e5	1.8e5
	RPTQ (Yuan et al., 2023)	12.23	18.87	15.93
	RPTQ*	11.16	13.73	11.78
	OmniQuant (Shao et al., 2023)	10.29	13.19	11.35
	<b>Ours L4</b>	<b>10.26</b>	<b>13.30</b>	<b>11.31</b>
	<b>Ours</b>	<b>10.21</b>	<b>13.08</b>	<b>11.26</b>

- PTB perplexity with weight-only quantization in OPT families (Table 10).
- C4 perplexity with weight-only quantization in OPT families (Table 11).

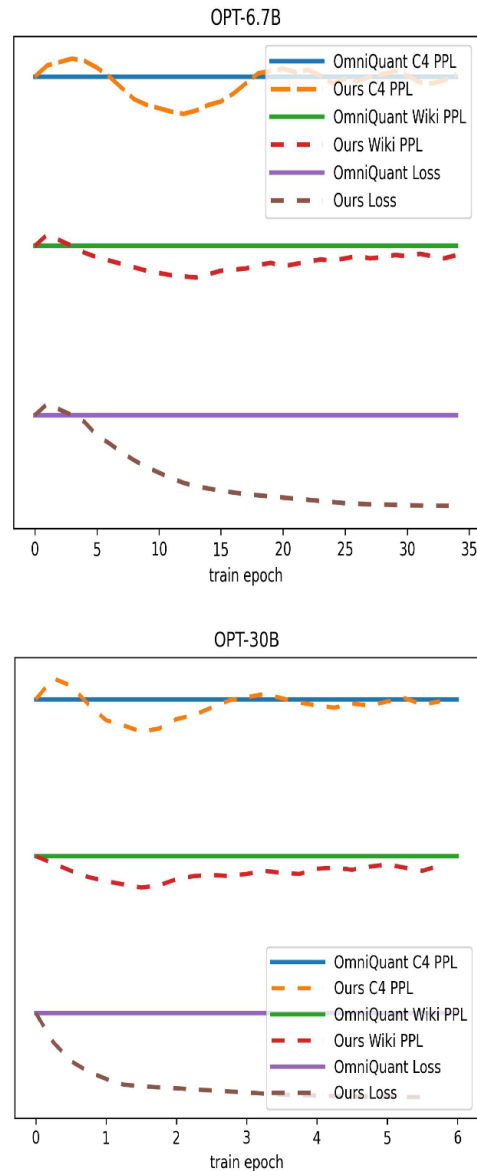


Figure 2: **Training details about LSI** on OPT-6.7B and OPT-30B in the setting of W4A16g128. The magnitude is not drawn in scale. Since OPT-30B is really sensitive to train epochs, in the OPT-30B part, we make one train epoch containing 24 samples, and we train it from 8 samples to 144 samples at the interval of 8 samples.

Table 9: **Weight-only** quantization results of LLaMA-1 Models (7B-30B) in settings without group-wise scaling. The LLaMA families exhibit insensitivity to scaling and weight distribution; only handling outliers can make a noticeable difference. In OmniQuant, they solely employ LWC instead of LWC+LET, meaning all progress is achieved through outlier elimination. In this context, we observed that LSI does not bring about significant improvements in group-wise settings. For LLaMA-7B and LLaMA-13b, there is a slight increment with changes around 0.01 degree of perplexity on average. However, for LLaMA-30B, the increment is nearly negligible. Therefore, the implementation of LSI is not considered necessary in group-wise scaling on LLaMA families. But we also release our checkpoint of LLaMA-7b and LLaMA-13b on those settings, so anyone can check for examination.

LLaMA / PPL↓		LLaMA-7B		LLaMA-13B		LLaMA-30B	
Task		WIKI	C4	WIKI	C4	WIKI	C4
FP16	-	10.86	11.74	10.12	11.19	9.56	10.69
W2A16	GPTQ (Frantar et al., 2023)	2.1e3	689.13	5.5e3	2.5e3	499.75	169.80
	OmniQuant (Shao et al., 2023)	15.47	24.89	13.21	18.31	8.71	13.89
	<b>Ours</b>	<b>12.91</b>	<b>17.90</b>	<b>9.08</b>	<b>12.36</b>	<b>8.45</b>	<b>11.96</b>
W3A16	GPTQ (Frantar et al., 2023)	8.06	9.49	6.76	8.16	5.84	7.29
	AWQ (Lin et al., 2023)	11.88	13.26	7.45	9.13	10.07	12.67
	OmniQuant (Shao et al., 2023)	6.49	8.19	5.68	7.32	4.74	6.57
	<b>Ours</b>	<b>6.38</b>	<b>8.17</b>	<b>5.65</b>	<b>7.33</b>	<b>4.69</b>	<b>6.58</b>
W4A16	GPTQ (Frantar et al., 2023)	6.13	7.43	5.40	6.84	4.48	6.20
	AWQ (Lin et al., 2023)	6.08	7.52	5.34	6.86	4.39	6.17
	OmniQuant (Shao et al., 2023)	5.86	7.34	5.21	6.76	4.25	6.11
	<b>Ours</b>	<b>5.84</b>	<b>7.32</b>	<b>5.20</b>	<b>6.75</b>	<b>4.24</b>	<b>6.11</b>

Table 10: **PTB perplexity of Weight-only quantization results in OPT models.**

OPT / PPL↓		125M	1.3B	2.7B	6.7B	13B	30B	66B
FP16	-	32.54	16.96	15.11	13.08	12.33	11.84	11.36
W2A16 g128	GPTQ (Frantar et al., 2023)	655.17	130.88	61.36	25.24	20.46	15.15	323.23
	AWQ (Lin et al., 2023)	263.88	71.87	43.15	19.49	17.61	14.92	19.33
	OmniQuant (Shao et al., 2023)	126.49	34.33	25.28	18.92	16.74	14.51	139.17
	<b>Ours</b>	<b>92.60</b>	<b>32.26</b>	<b>24.39</b>	<b>18.71</b>	<b>16.44</b>	<b>14.27</b>	<b>116.21</b>
W2A16 g64	GPTQ (Frantar et al., 2023)	245.28	55.61	36.12	19.45	17.02	14.05	88.92
	AWQ (Lin et al., 2023)	143.18	41.19	25.08	18.00	15.83	14.92	15.72
	OmniQuant (Shao et al., 2023)	112.10	30.36	22.63	17.58	15.70	13.98	13.51
	<b>Ours</b>	<b>81.40</b>	<b>29.17</b>	<b>22.51</b>	<b>17.55</b>	<b>15.55</b>	<b>13.90</b>	<b>13.47</b>
W3A16	GPTQ (Frantar et al., 2023)	34.05	27.39	15.94	13.75	13.71	12.54	21.16
	AWQ (Lin et al., 2023)	80.73	33.20	224.11	18.46	35.45	66.68	3.4e3
	OmniQuant (Shao et al., 2023)	45.29	20.42	17.08	14.23	13.49	12.54	11.71
	<b>Ours</b>	<b>40.56</b>	<b>19.85</b>	<b>16.65</b>	<b>14.02</b>	<b>13.42</b>	<b>12.48</b>	<b>11.69</b>
W4A16	GPTQ (Frantar et al., 2023)	37.75	18.23	15.94	13.75	12.58	11.98	11.58
	AWQ (Lin et al., 2023)	38.74	18.35	15.70	13.59	12.72	12.06	11.58
	OmniQuant (Shao et al., 2023)	34.94	17.80	15.52	13.41	12.62	11.95	11.86
	<b>Ours</b>	<b>34.83</b>	<b>17.74</b>	<b>15.43</b>	<b>13.37</b>	<b>12.55</b>	<b>11.95</b>	<b>11.73</b>

Table 11: **C4 perplexity of Weight-only quantization results in OPT models.**

OPT / PPL↓		125M	1.3B	2.7B	6.7B	13B	30B	66B
FP16	-	24.60	14.72	13.16	11.74	11.19	10.69	10.69
W2A16 g128	GPTQ (Frantar et al., 2023)	597.66	60.88	33.83	18.55	16.34	12.89	598.81
	AWQ (Lin et al., 2023)	168.35	38.38	26.41	16.48	14.73	12.98	15.42
	OmniQuant (Shao et al., 2023)	80.10	27.33	21.11	16.67	14.92	13.12	73.83
	<b>Ours</b>	<b>64.17</b>	<b>25.76</b>	<b>20.61</b>	<b>16.28</b>	<b>14.66</b>	<b>13.00</b>	<b>66.25</b>
W2A16 g64	GPTQ (Frantar et al., 2023)	133.51	31.31	23.23	16.24	14.48	12.24	58.60
	AWQ (Lin et al., 2023)	90.19	27.34	20.01	15.20	13.90	12.43	13.31
	OmniQuant (Shao et al., 2023)	64.01	23.71	19.16	15.44	14.16	12.80	12.13
	<b>Ours</b>	<b>56.22</b>	<b>23.53</b>	<b>19.03</b>	<b>15.31</b>	<b>13.97</b>	<b>12.75</b>	<b>12.10</b>
W3A16	GPTQ (Frantar et al., 2023)	37.75	19.45	13.75	15.67	12.28	11.34	13.68
	AWQ (Lin et al., 2023)	55.73	24.56	154.49	15.84	23.71	55.01	3.8e3
	OmniQuant (Shao et al., 2023)	32.17	17.10	14.93	12.78	12.13	11.37	10.82
	<b>Ours</b>	<b>30.20</b>	<b>16.71</b>	<b>14.59</b>	<b>12.56</b>	<b>12.14</b>	<b>11.38</b>	<b>10.79</b>
W4A16	GPTQ (Frantar et al., 2023)	27.12	15.57	13.75	12.15	11.36	10.80	10.50
	AWQ (Lin et al., 2023)	27.64	15.65	13.71	12.04	11.42	10.83	10.41
	OmniQuant (Shao et al., 2023)	26.36	15.28	13.58	11.97	11.41	10.80	10.63
	<b>Ours</b>	<b>26.02</b>	<b>15.26</b>	<b>13.52</b>	<b>11.94</b>	<b>11.37</b>	<b>10.79</b>	<b>10.47</b>