LR*Quant*: Learnable and Robust Post-Training Quantization for Large Language Models

Jiaqi Zhao, Miao Zhang*, Chao Zeng, Ming Wang, Xuebo Liu, Liqiang Nie
School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen)
{24B351012,23S151104,190110509}@stu.hit.edu.cn
{zhangmiao,liuxuebo,nieliqiang}@hit.edu.cn

Abstract

Post-training quantization (PTQ) for large language models (LLMs) significantly accelerates model inference and relieves memory constraints, without incurring model training. A "smoothing paradigm" is commonly used in LLM quantization, which transfers the quantization difficulty of activation to weight quantization using mathematically equivalent transformations. However, existing methods face two issues: 1) Most smoothing parameters are hand-crafted defined which leads to suboptimal results; 2) There are significant performance degradations when tested on unseen datasets. To address these challenges, this paper introduces a robust learnable smooth-based PTQ framework, called **LR*Quant***. Firstly, we consider a learnable paradigm to find optimal smoothing parameters which are initialized by logarithmic activation equivalent. In addition, we empirically found that only relying on MSE loss could hardly lead to optimal quantization results, and we then propose a novel loss function based on the negative logarithm of cosine similarity (NLC loss) between outputs of full-precision and quantized block. At last, we *pioneeringly* introduce Test-time adaptation (TTA) into LLM quantization, which allows for rapid model adaptation during testing to improve generalization performance. More surprisingly, we find that by using our TTA method, we can achieve better results on test sets than directly using test sets for calibration in some cases while avoiding catastrophic forgetting. Codes are available at https://github.com/zjq0455/RLQ.

1 Introduction

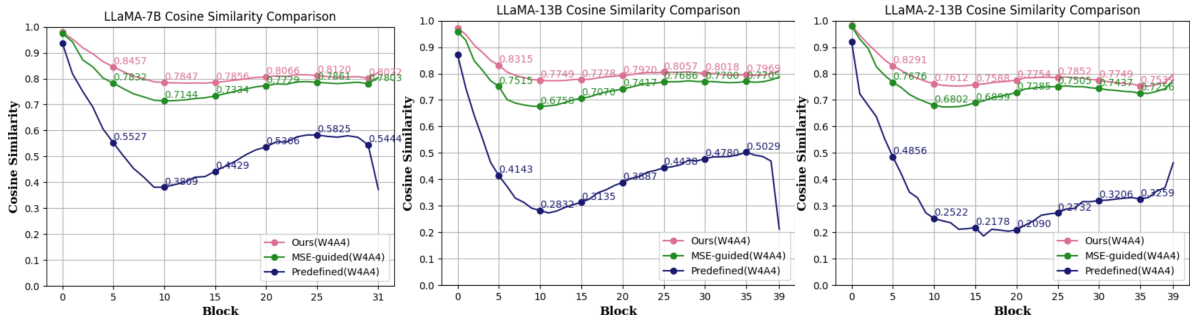
Quantization (Esser et al., 2019; Chee et al., 2023; Dettmers et al., 2023a,b) is a well-known model compression technique converting the weights of large language models (LLMs) and activations from full precision to lower-bit representations. Among various methods, Post-Training Quantization (PTQ) (Nagel et al., 2020; Hubara et al., 2021; Yao et al., 2022) is the most popular for LLMs, as it doesn't require retraining the model and offers fast quantization, saving computational resources. For instance, GPTQ (Frantar et al., 2022) uses only an A100-80G GPU to quantize BLOOM-175B (Workshop et al., 2022) within 4 hours.

Previous PTQ methods exhibit uncommendable performance when confronted with more challenging configurations, such as W4A4, due to larger activation distribution variances. A new strategy, known as "smooth quantization" (Xiao et al., 2023), has been recently introduced to address this by shifting the quantization difficulty from activations to weights. However, current smoothing strategies, such as hand-crafted scaling factors (Xiao et al., 2023; Frantar et al., 2022) and uniform zero points (Wei et al., 2023) are all predefined, which usually lead to suboptimal. The intuitive proposition of learning the smoothing parameters, coupled with the adoption of MSE loss which aims to optimize quantization output by evaluating the similarity in magnitude, is an easily conceived solution. We conducted an example experiment and summarized the cosine similarity between the outputs of each quantized block (MSE-guided learning(Shao et al., 2024) and predefined method (Xiao et al., 2023)) and full-precision block. As shown in Figure 1, there are large differences, both for the predefined and the MSE-guided learning method, demonstrating that except for MSE, another learning strategy for better optimizing is essential.

In addition, a prevalent issue across prior PTQ methods is that quantization on the calibration dataset leads to suboptimal performance when subsequently applied to unseen datasets. For instance, quantize a LLaMA based on WikiText2 (Merity et al., 2016) and evaluate on PTB (Marcus et al., 1994). Meanwhile, Test-time adaptation (TTA) is an increasingly promising technique to improve

*Corresponding author

Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2240–2255



(a) LLaMA-7B (b) LLaMA-13B (c) LLaMA-2-13B

Figure 1: Cosine similarity between outputs of full-precision and quantized block (by predefined method (Xiao et al., 2023), MSE-guided learning method (Shao et al., 2024), and our **LR*Quant***).

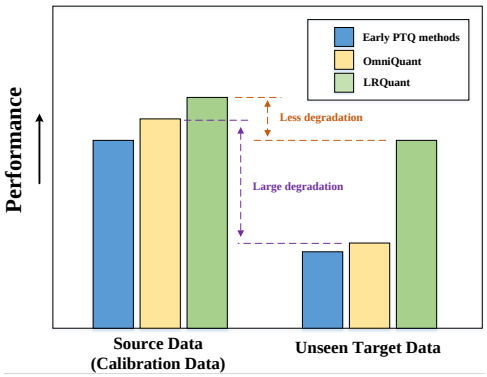


Figure 2: **LR*Quant*** not only outperforms previous methods under general setting with the help of NLC loss but also has a strong generalization capability.

model generalization capacity (Wang et al., 2020; Niu et al., 2022, 2023), which enhances the model's robustness during testing without requiring retraining. However, according to our investigation, there is currently no method that utilizes TTA to address the aforementioned issue in PTQ methods.

Considering all the aforementioned aspects, this paper introduces a novel PTQ framework named **LR*Quant***. Firstly, we define learnable parameters and devise a novel block-wise loss function, named negative logarithm of cosine similarity loss (NLC loss) considering directional differences of outputs, to guide the learning of all learnable parameters. As depicted in Figure 1, this modification significantly improves our quantization performance. Additionally, **LR*Quant*** is the *first* to incorporate TTA strategy into LLMs quantization aiming to fortify models' robustness on unseen test sets. Rather than adapting the learnable parameters of the whole model based on the unseen data where we observe catastrophic forgetting, **LR*Quant*** only updates the learnable parameters of the last block to effectively balance performance and forgetting. Figure 2 illustrates an approximate performance comparison between our **LR*Quant*** and previous PTQ methods. More excitedly, we observe that with our TTA method, the adapted model even outperforms the quantized model directly calibrated by the test set in some cases. This discovery implies that if a quantized model needs to be applied to different scenarios, there will be no need to re-calibrate and re-quantize the full-precision model from scratch.

Our key contributions can be summarized as:

- We present a learnable PTQ framework for LLMs (called **LR*Quant***), by setting smoothing parameters into learnable with logarithmic activation equalization initialization.
- We propose a novel block-wise loss function called NLC loss, which is based on the negative logarithm of cosine similarity between outputs of full-precision and quantized blocks, to assist in correcting the output directions rather than only the magnitude by MSE loss.
- **LR*Quant*** is the *first* to introduce TTA into PTQ by updating the learnable parameters of the last block based on test data, which enhances the performance on unseen test data in some cases, while avoiding catastrophic forgetting on source data. Remarkably, our adapted quantized model exhibits superior performance compared to calibration using test sets from scratch.

2 Related Work

2.1 Smooth-Based Post-Training Quantization

"Smooth" is a solution proposed to address the challenge of activation quantization difficulty (Xiao

et al., 2023). It is based on mathematical equivalent transformations, where the activations with large variances across different channels are divided by scaling factors. Correspondingly, the weights which are generally easier to quantize are multiplied by scaling factors. Outlier Suppression (Wei et al., 2022) finds that the scaling parameter γ in LayerNorm (LN) layer of LLM is a key factor influencing the distribution of activation value outliers. Therefore, Gamma Migration is proposed, which involves removing γ from LN and moving it to the weights in the next layer and the short branch. Outlier Suppression+ (Wei et al., 2023) introduces channel-wise shifting, which eliminates the asymmetry in activation distribution and reduces the range of the tensor. SmoothQuant (Xiao et al., 2023) is the first method to introduce channel-wise scaling factors designed based on the maximum value. FPTQ (Li et al., 2023) improves the calculation method of scaling factors by utilizing non-linear offline logarithmic activation equalization (LAE) to adjust the distribution of activations, making the activation distribution moderate. OmniQuant (Shao et al., 2024) stands out as the latest smooth-based PTQ method which defines learnable smoothing parameters. However, OmniQuant uses the basic channel-wise maximum values as the initial scaling factors and only use MSE loss to update, means that its performance can not reach the optimal level. This paper also defines learnable parameters where smoothing parameters are initialized with LAE. Furthermore, we design a novel loss function namely NLC loss based on cosine similarity considering the directional differences of outputs to update the learnable parameters.

2.2 Test-Time Adaptation

Test-time adaptation is a technique to adapt models to unseen datasets to improve robustness during testing. Unlike fine-tuning (Wang et al., 2017; Howard and Ruder, 2018; Hu et al., 2021), it is source-free and doesn't require additional labels from test data. Tent (Wang et al., 2020) minimizes the prediction entropy generated by the model during testing to reduce generalization error. EATA (Niu et al., 2022) believes that excessive adaptation can lead to catastrophic forgetting of the source data and proposes a sample selection method to choose samples with high contribution to adaptation from the test sets for entropy minimization. On this basis, SAR (Niu et al., 2023) proposes removing noisy samples with large gradients during testing and enhancing robustness to noise through sharpness-aware learning on the remaining samples. TS (Park et al., 2023) changes the traditional approach of updating model parameters and instead performs style shifting on test samples to make them resemble the nearest source domain.

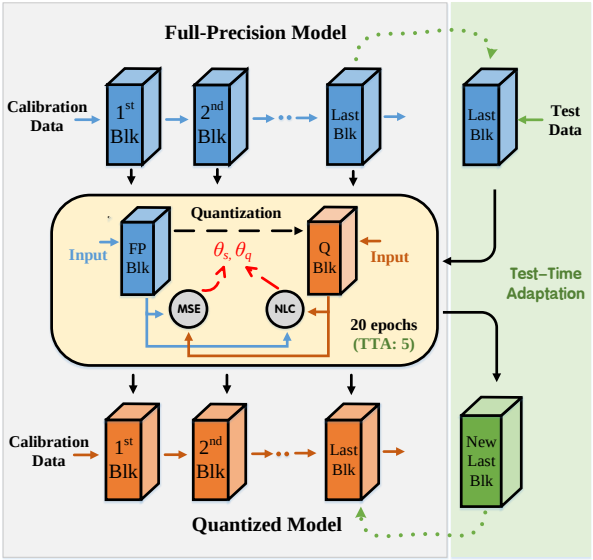


Figure 3: An overview of our **LR*Quant***. **LR*Quant*** uses our NLC loss and MSE loss to update learnable smoothing and quantization parameters. During testing, **LR*Quant*** utilizes test data to adapt the last block to improve performance on unseen data.

In this paper, we are the first to introduce TTA into LLMs quantization, aiming to improve the model's generalization capability on unseen test sets while avoiding catastrophic forgetting and outperforming re-calibration from scratch.

3 LR*Quant*

In this section, we provide a detailed introduction to our **LR*Quant*** as shown in Figure 3. We first describe the learnable parameters and NLC loss and then introduce our TTA approach for quantized models on unseen datasets.

3.1 Learnable Parameters

Learnable Weight Quantization. For weight quantization, the general function is as follows:

$$\mathrm{W_q} = \mathrm{clamp}(\lfloor \frac{\mathrm{W}}{S} \rceil + z, 0, 2^b - 1), \quad (1)$$

where $\lfloor . \rceil$ means round-to-nearest operation. b denotes the target bit-width. $\mathrm{W_q}$ is the quantized weight and W is the full-precision weight. S is the step size and z is called zero-point, and we follow

Lin et al. (2023) to learn these two parameters to improve performance, which can be elaborated as:

$$S = \frac{\alpha \text{max(W)} - \beta \text{min(W)}}{2^b - 1}, z = -\lfloor \frac{\beta \text{min(W)}}{S} \rceil, \quad (2)$$

where α and β are learnable quantization parameters (Shao et al., 2024) to control the upper and lower bound of clipping range respectively and they are constrained within the range of [0, 1].

Learnable Activation Smoothing Quantization. Smooth-based PTQ methods are supposed to smooth the distribution of activations to mitigate the impact of outliers under the condition of outputting exactly equal results in mathematical form. For linear layers, this can be expressed as:

$$\text{X'W'} + \text{B'} = \text{XW} + \text{B}, \quad (3)$$

where X, W, B and X', W', B' are activation, weight, bias before and after equivalent transformation respectively. In more detail:

$$\text{X'} = (\text{X} - z_s)\text{diag}(s)^{-1}, \quad (4)$$

$$\text{W'} = \text{diag}(s)\text{W}, \quad (5)$$

$$\text{B'} = \text{B} + z_s\text{W}, \quad (6)$$

where s and z_s are scaling factors and shifting factors (Wei et al., 2023) to narrow down the activation distribution range. We should notice that in order to calculate easily and implement smooth quantization in codes, for activation quantization, s and z_s are fused into the weights of previous LN or linear layers, and for weights, s can be absorbed into themselves.

As is well-known, the initial values of hyperparameters have a significant impact on the final results (Glorot and Bengio, 2010). Inspired by (Lin et al., 2023) and (Li et al., 2023), the impact of the smoothing process on weight quantization is consistently smaller compared to the benefits obtained by smoothing activation and the use of logarithmic equivalence proves to be more effective in suppressing activation outliers. In this way, we define the initial scaling factor s as:

$$s_i^0 = \max(|\text{x}_i|)/\log_a(a + \max(|\text{x}_i|)), \quad (7)$$

where i is the index of the input channel. We set a as 2 followed by (Li et al., 2023), while s_i^0 is just an initial value as our scaling factor is learnable.

3.2 NLC Loss

Block-wise quantization is commonly used in PTQ for LLMs, which quantizes the whole model block by block, to avoid excessive GPU memory usage. The MSE is usually adopted to guide the quantization process:

$$\mathcal{L}_{MSE} = \|F(\text{W}, \text{X}) - F(Q_w(\text{W}), Q_a(\text{X}))\|_2^2, \quad (8)$$

where $F(\cdot)$ is the embedding function of a block. $Q_w(\cdot)$ represents the weight quantizer with both learnable quantization parameters Θ_q (short for α and β) and smoothing parameters Θ_s (short for s and z_s) while $Q_a(\cdot)$ denotes the activation quantizer with only smoothing parameters Θ_s.

However, MSE loss only measures the magnitude similarity of vectors without considering the direction similarity. Hence, this paper is motivated by maximizing the cosine similarity between the output of each full-precision and quantized block, to improve directional similarity. We leverage the negative logarithm of cosine similarity for the outputs as the loss function called NLC loss $\mathcal{L}_{NLC}$:

$$\mathcal{L}_{NLC} = -\log(\mathcal{C}(F(\text{W}, \text{X}), F(Q_w(\text{W}), Q_a(\text{X})))), \quad (9)$$

where $\mathcal{C}(\cdot)$ is the cosine similarity operator:

$$\mathcal{C}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}. \quad (10)$$

In the end, we combined MSE loss and NLC loss, and our final optimization objective is:

$$\Theta_q^*, \Theta_s^* = \underset{\Theta_q, \Theta_s}{\text{argmin}}(\mathcal{L}_{MSE} + \mathcal{L}_{NLC}), \quad (11)$$

where Θ_q^* and Θ_s^* are the optimal learnable parameters. Our NLC loss approaches 0 when the quantized and the full-precision outputs are close to be the same, so as guiding the quantization process. As illustrated in Figure 1, with NLC loss, **LR*Quant*** has significantly outperformed MSE-only method.

3.3 TTA for Quantized Models

In most existing PTQ methods for LLMs, the pretrained large model is first quantized based on the calibration dataset, which is then directly evaluated on unseen test datasets. However, when applying the quantized model to an unseen dataset with domain shift, it may cause performance degradation. A naive solution is to re-calibrate the whole quantized model based on the test dataset, while which is time-consuming. In addition, we empirically

found that, adapting the entire model using test data will inevitably lead to catastrophic forgetting of source calibration knowledge, leading to performance degradation. Therefore, balancing model adaptation performance and catastrophic forgetting is crucial. To address this issue, we innovatively introduce the idea of TTA into LLM quantization.

Considering that the most task-relevant block in the model is the last one, we choose to only adapt the last block of the quantized model based on the unseen test data. As illustrated in Figure 3, in the TTA phase of our **LR*Quant***, we only extract the last block from the full-precision model for quantization based on the test data, and we still utilize a combination of MSE loss and NLC loss to update the parameters with only a few epochs. The adaptation goal is as follows:

$$\Theta_q^{l*}, \Theta_s^{l*} = \underset{\Theta_q^l, \Theta_s^l}{\operatorname{argmin}} (\mathcal{L}_{MSE} + \mathcal{L}_{NLC}), \quad (12)$$

where Θ_q^{l*} and Θ_s^{l*} are the learnable parameters of the last block after adaptation, where we set 5 as the number of TTA epochs. This entire process is source-free, and the time required for each adaptation process on test sets is within one minute.

An intuitive question is why only update the learnable parameters of the last block rather than the entire model. Here we describe the two advantages of our TTA scheme over re-calibration for PTQ. First, it is much more time-consuming and challenging to update the whole model than only the last block. Second, re-calibrating the whole model based on the test set can lead to catastrophic forgetting which significantly alters the suitable quantization parameters for source calibration data. Differently, our TTA scheme only re-calibrates the last block, which can somehow retain source knowledge. Moreover, our experimental results in Table 6 show that, compared to calibrating the whole full-precision model using test sets from scratch, our TTA scheme can achieve much better performance on unstable target sets such as PTB, which maintains its performance on the source calibration set even after adapting to the new test dataset. In the end, we present the entire **LR*Quant*** pseudocode as Algorithm 1 at Appendix A.

4 Experiments

In this section, we conduct extensive experiments to validate our **LR*Quant*** achieves the following objectives: (1) Demonstrates outstanding performance in challenging quantization tasks (W4A4 and W6A6) through the presented learnable parameters and NLC loss. (2) Enhances model performance on unseen test sets using our TTA scheme.

4.1 Experimental Setup

Baseline. Because **LR*Quant*** is a smooth-based PTQ method, we select the same type but predefined methods SmoothQuant (Xiao et al., 2023), LAE (Li et al., 2023) and learnable approach OmniQuant (Shao et al., 2024) as baselines. It is important to note that, due to the lack of data from original papers and to ensure consistency with our **LR*Quant*** experimental environment, all results of other methods are reproduced as their settings.

Models. We mainly choose LLaMA (7B,13B and 30B) (Touvron et al., 2023a) and LLaMA-2 (7B,13B) (Touvron et al., 2023b) to evaluate our **LR*Quant***, since LLaMA and LLaMA-2 families are currently the most popular and widely applied LLMs. Additionally, we also conduct experiments on OPT (1.3B, 2.7B, 6.7B) (Zhang et al., 2022), which can be found in Appendix D.

Training. For learnable smoothing parameters, we initialize the channel-wise shifting factors as Wei et al. (2023) and the scaling factors with Eq.(7), where we set the base a as 2. We utilize the AdamW optimizer (Loshchilov and Hutter, 2017) with zero weight decay to optimize the learnable smoothing parameters and quantization parameters with learning rate 1e-3 and 1e-2 respectively. Our calibration data consists of 128 random 2048 token-segments from WikiText2 (Merity et al., 2016). Using a batch size of 1, all processes are performed on one NVIDIA A100-40G GPU. The calibration training process consists of 20 epochs, while the TTA process involves 5 epochs.

Datasets. Following most smooth-based methods (Shao et al., 2024; Xiao et al., 2023), test data comes from WikiText2, Penn Treebank (PTB) (Marcus et al., 1994), and C4 (Raffel et al., 2020) for language generation tasks. For evaluating the performance on zero-shot tasks, we select several popular tasks including PIQA (Bisk et al., 2020), ARC (Clark et al., 2018), BoolQ (Clark et al., 2019), HellaSwag (Zellers et al., 2019) and Winogrande (Sakaguchi et al., 2021) using lm-evaluation-harness (Gao et al., 2023).

4.2 Experiments on Language Generation Tasks

The core competency requisite for LLMs is to comprehend and generate language. Hence, to vali-

Bits	Models	Methods	WikiText2	PTB	C4	PTB-new	C4-new
W4A4	LLaMA-7B	FP16	5.67	27.34	7.07	41.15	7.34
		LAE	135.72	500.23	140.69	648.07	166.53
		SmoothQuant	38.09	203.76	58.50	320.09	74.12
		OmniQuant	12.46	107.31	15.83	201.06	17.78
		LR*Quant*	**11.25**	**52.05**	**14.14**	**99.28**	**15.41**
	LLaMA-13B	FP16	5.09	19.22	6.61	28.09	6.79
		LAE	151.02	283.48	132.17	397.80	148.29
		SmoothQuant	79.01	267.09	99.58	297.53	121.95
		OmniQuant	12.39	72.26	15.47	105.07	17.38
		LR*Quant*	**11.26**	**42.76**	**13.19**	**63.78**	**14.53**
	LLaMA-30B	FP16	4.10	16.29	5.98	23.51	6.13
		LAE	388.53	855.89	210.49	1194.53	233.55
		SmoothQuant	449.28	1320.15	193.36	2291.59	233.35
		OmniQuant	15.09	68.11	18.05	90.31	19.68
		LR*Quant*	**12.00**	**36.55**	**12.83**	**53.64**	**14.12**
	LLaMA-2-7B	FP16	5.47	22.51	6.97	37.91	7.26
		LAE	180.94	2114.33	155.20	6149.77	174.06
		SmoothQuant	106.18	1477.48	98.42	2961.55	113.35
		OmniQuant	16.66	717.61	21.16	1572.5	24.05
		LR*Quant*	**12.75**	**87.63**	**15.82**	**281.41**	**17.57**
	LLaMA-2-13B	FP16	4.88	28.87	6.47	50.93	6.73
		LAE	338.27	1231.87	516.26	2102.05	541.60
		SmoothQuant	184.11	1500.14	165.09	2681.07	158.29
		OmniQuant	12.93	147.25	15.47	**253.71**	17.25
		LR*Quant*	**12.23**	**132.83**	**14.02**	262.99	**15.57**

Table 1: **W4A4** perplexities (**lower is better**) comparison of quantized LLaMA and LLaMA-2 models. **W6A6** results are in Table 11 at Appendix D. All models are quantized based on WikiText2 and evaluated on all datasets.

date the first objective—achieving superior performance in challenging quantization tasks—we initially compare a crucial metric for language generation tasks, perplexity, with the baselines.

The outcomes presented in Table 1 highlight a significant superiority of our learnable method over predefined methods in all W4A4 experiments. When compared to another learnable method, OmniQuant, across WikiText2, C4, and C4-new tasks, **LR*Quant*** exhibits a noteworthy reduction in perplexities by an average of 2.00, 3.20, and 3.79, respectively. Notably, on PTB and PTB-new tasks where OmniQuant faces substantial challenges, **LR*Quant*** demonstrates heightened stability. A more visually intuitive representation of the cosine similarity comparison with OmniQuant is depicted in Figure 1. W6A6 results can be found in Table 11 at Appendix D and OPT results can be found in Table 13 at Appendix E. These findings underscore the efficacy of learnable parameters and NLC loss.

4.3 Experiments on Zero-Shot Tasks

In addition to language generation, zero-shot tasks reflect the model's ability to handle unseen problems, also serving as a crucial metric for characterizing model performance. To further validation, we compare zero-shot accuracies with baselines.

As indicated in Table 2, similar to the results observed in language generation tasks, our **LR*Quant*** consistently outperforms all predefined methods. Moreover, our **LR*Quant*** also exhibits superior performance compared to OmniQuant in most cases, showcasing an average performance increase range of 3.22%∼8.95%. Results and analysis for W6A6 can be found in Table 12. Combining the experimental results from the two subsections, we deduce that our **LR*Quant*** attains the state-of-the-art level in challenging quantization tasks. Additionally, in order to further demonstrate the superiority of our method on weight-only tasks, we conduct corresponding experiments in Appendix G.

4.4 Ablation Experiments

Having showcased the superior performance of **LR*Quant*** in challenging quantization tasks, our objective extends to providing additional evidence for the effectiveness of our two innovations: learnable parameters and NLC loss. To substantiate this, we conduct ablation studies on the LLaMA-7B at W4A4, and the results are presented in Table 3, where *LAE* and *NLC* represent the initialization methods for learnable smoothing parameters and

Models	Methods	PIQA	ARC-e	ARC-c	BoolQ	HellaS	WinoG	Avg.
LLaMA-7B	FP16	78.40	67.34	38.13	73.11	56.42	66.85	63.38
	LAE	56.58	28.74	21.84	56.39	27.32	52.72	40.60
	SmoothQuant	61.86	42.42	22.86	58.37	33.30	50.51	44.89
	OmniQuant	63.49	46.17	24.74	62.53	39.58	53.51	48.34
	LR*Quant*	**66.64**	**52.98**	**28.92**	**63.30**	**43.99**	**53.51**	**51.56**
LLaMA-13B	FP16	78.78	74.53	43.94	68.53	59.09	70.08	65.82
	LAE	58.10	35.31	22.09	61.31	30.60	50.98	43.07
	SmoothQuant	62.45	44.31	24.48	61.07	35.63	50.11	46.34
	OmniQuant	63.87	47.72	26.27	62.17	40.78	52.64	48.91
	LR*Quant*	**72.41**	**57.91**	**31.65**	**64.92**	**47.43**	**56.43**	**55.13**
LLaMA-30B	FP16	80.08	58.92	45.47	68.44	79.21	72.53	67.44
	LAE	56.03	28.45	19.45	50.48	26.69	48.85	38.33
	SmoothQuant	54.57	28.82	19.45	55.90	26.93	49.88	39.26
	OmniQuant	65.29	50.25	24.06	62.17	41.87	52.41	49.34
	LR*Quant*	**73.12**	**61.53**	**33.28**	**70.64**	**50.71**	**60.46**	**58.29**
LLaMA-2-7B	FP16	78.45	69.32	40.02	71.07	56.69	67.25	63.80
	LAE	56.20	28.87	21.75	56.26	28.13	52.32	40.59
	SmoothQuant	58.59	32.65	21.67	59.90	29.23	50.19	42.04
	OmniQuant	60.39	43.47	22.26	61.43	37.25	49.64	45.74
	LR*Quant*	**63.71**	**46.96**	**26.27**	**63.39**	**44.42**	**53.19**	**49.66**
LLaMA-2-13B	FP16	78.73	73.27	45.56	69.02	59.72	69.61	65.99
	LAE	54.62	29.50	20.90	58.25	28.56	50.35	40.36
	SmoothQuant	54.18	28.78	19.28	60.55	27.82	50.19	40.13
	OmniQuant	**67.36**	**53.96**	**29.61**	63.33	44.93	51.54	**51.79**
	LR*Quant*	61.70	43.06	27.56	**64.19**	**46.32**	**54.85**	49.61

Table 2: Zero-shot accuracies (**higher is better**) comparison of quantized LLaMA and LLaMA-2 models at **W4A4**. **W6A6** results can be found in Table 12 at Appendix D.

our NLC loss, respectively.

LAE	MSE	NLC	Wiki-2	PTB	C4
	✓		12.46	107.31	15.83
✓	✓		12.34	91.58	15.69
✓		✓	12.22	**50.57**	14.98
✓	✓	✓	**11.25**	52.05	**14.14**

Table 3: Ablation results of LLaMA-7B quantized by **LR*Quant*** at W4A4 quantization.

The first row in the table is same as OmniQuant which only uses the MSE loss. It is evident from the second and third row that each innovation in our **LR*Quant*** is effective. Although *LAE* + *NLC* group achieves optimal performance on PTB, which shows mediocre results on WikiText2 and C4. Conversely, *LAE* + *MSE* + *NLC* demonstrates the overall best performance, which is selected as our default setting.

We also conduct experiments to demonstrate the effectiveness of smoothing strategy. As illustrated in Figure 4, it can be observed that after smoothing the magnitude of the activation values has significantly decreased and shows much better performance, especially for those channels with more prominent outliers.

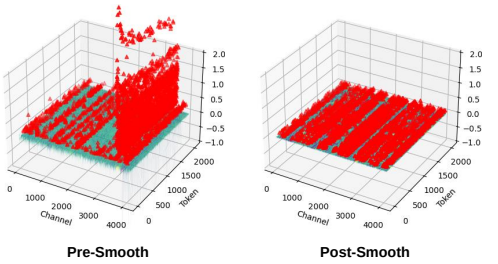


Figure 4: The magnitude comparison of activations before and after smoothing by **LR*Quant*** in a layer of LLaMA-7B.

Model	Methods	Wiki-2	PTB	C4	PTB-n	C4-n
LLaMA-7B	w/o smooth	2.9e3	2.4e3	3.1e3	2.8e3	3.5e3
	smooth	11.25	52.05	14.14	99.28	15.41
LLaMA-2-7B	w/o smooth	462.03	661.47	472.60	1044.91	514.84
	smooth	12.75	87.63	15.82	281.41	17.57

Table 4: Perplexity comparison of **LR*Quant*** with and w/o smooth at W4A4.

4.5 Experiments on Test-Time Adaptation Performance

After achieving the first validation objective, the subsequent goal is to verify whether our TTA method can enhance the performance on unseen datasets. We compare the perplexities of **LR*Quant***

Bits	**Models**	**Methods**	WikiText2	PTB	C4	PTB-new	C4-new
W4A4	LLaMA-7B	pre-TTA	11.25	52.05	14.14	99.28	15.41
		post-TTA	11.25	54.22	14.83	**98.10**	16.32
	LLaMA-13B	pre-TTA	11.26	42.76	13.19	63.78	14.53
		post-TTA	11.26	**40.58**	13.44	**60.34**	15.08
	LLaMA-2-7B	pre-TTA	12.75	87.63	15.82	281.41	17.57
		post-TTA	12.75	**80.69**	17.17	**227.27**	19.69
	LLaMA-2-13B	pre-TTA	12.23	132.83	14.02	262.99	15.57
		post-TTA	12.23	**97.42**	14.46	**196.16**	19.50
W6A6	LLaMA-7B	pre-TTA	5.88	32.56	7.35	49.14	7.67
		post-TTA	5.88	32.64	7.63	**49.09**	7.97
	LLaMA-13B	pre-TTA	5.27	20.13	6.84	28.39	7.07
		post-TTA	5.27	**19.98**	7.15	28.48	7.41

Table 5: Perplexities (**lower is better**) comparison of quantized models by **LR*Quant*** before and after TTA at W4A4 and W6A6. In this experiment, models are quantized on WikiText2 and then directly evaluated (pre-TTA) or adapted (post-TTA) on the remaining datasets.

Bits	**Testsets**	**Methods**	LLaMA-13B	LLaMA-2-13B
W4A4	PTB	Calibration	40.86	59.35
		TTA	**40.58**	97.42
W6A6	PTB	Calibration	20.07	30.61
		TTA	**19.98**	**29.99**

Table 6: Perplexities on target dataset after direct calibration and our TTA scheme. In this experiment, “Calibration" directly calibrates and quantizes models on the target dataset (PTB), while “TTA" quantizes models based on WikiText2, and adapts to the target datasets. This table reports their evaluation performance on the target datasets.

before and after adaptation, where the model is quantized based on WikiText2 while adapted and evaluated on remaining datasets. As illustrated in Table 5, although a little worse on C4, perplexities of almost all adapted models with our TTA scheme have shown a certain degree of reduction on PTB, which intuitively proves the effectiveness of our method.

Bits	**Methods**	LLaMA-7B	LLaMA-13B	LLaMA-2-7B
W4A4	Calibrate	137.2	240.9	137.8
	TTA	**0.4**	**0.6**	**0.4**
W6A6	Calibrate	138.6	243.9	137.7
	TTA	**0.5**	**0.6**	**0.4**

Table 7: Time (***miniute***) comparison of directly calibrating and TTA on PTB.

Models	**Methods**	W4A4	
		C4	PTB
LLaMA-7B	reCalib	11.77	14.09
	TTA	**11.25**	**11.25**
LLaMA-13B	reCalib	11.80	13.22
	TTA	**11.26**	**11.26**
LLaMA-2-7B	reCalib	13.64	15.12
	TTA	**12.75**	**12.75**
LLaMA-2-13B	reCalib	**12.10**	14.94
	TTA	12.23	**12.23**

Table 8: Perplexities on original WikiText2 at W4A4 after re-calibrating and *TTA* on test datasets (C4 and PTB). In this experiment, “reCalib" first quantizes models based on WikiText2 and then re-quantizes the model based on test datasets. This table reports their evaluation performance on the original WikiText2 dataset.

In addition, we compare our TTA scheme with directly calibrating the whole model using PTB. In Table 6, *Calibrate* means the model is directly calibrated and quantized using PTB and then immediately evaluated on it, while *TTA* means our TTA scheme where WikiText2 is the calibration set and adapt on PTB. The calibration of the whole model requires 300 $\times$ more time than TTA adaptation. More surprisingly, our TTA scheme can outperform directly calibrating the whole model with the test sets in some cases. For other cases with slight decreases, our method remains comparable performance but shows hundreds of times faster than recalibration (see Table 7). In summary, we consider that our TTA scheme can significantly promote the deployment efficiency of quantized models in new scenarios while ensuring effectiveness.

Last, we conduct experiments to evaluate our TTA scheme in relieving catastrophic forgetting. As indicated in Table 8, we evaluate the W4A4 performance on WikiText2 to compare re-calibrating the model from scratch (*reCalib*) with our TTA method (*TTA*) after adapting on new datasets. As shown, the performance on the original WikiText2 deteriorates after re-calibrating on new test sets (C4 and PTB). Differently, our *TTA* can retain its performance on the original dataset, verifying its effectiveness in mitigating catastrophic forgetting. W6A6 results are in Table 14 at Appendix F.

5 Conclusion

In this paper, we propose a robust smooth-based PTQ framework namely **LR*Quant***. Firstly, **LR*Quant*** defines learnable smoothing parameters and initializes them using LAE. Subsequently, we introduce a novel block-wise loss function namely NLC loss which further considers the directional similarity of outputs. The experiments substantiate that our **LR*Quant*** attains state-of-the-art levels on challenging weight-activation quantization tasks. Moreover, we pioneeringly proposed a TTA scheme for LLM quantization to improve generalization performance on unseen datasets by adapting the learnable parameters of the last block based on the test data. The experiments indicate that our method not only relieves catastrophic forgetting but also surpasses re-calibration using the target set, thereby significantly promoting the deployment efficiency of quantized models in some cases.

Limitations

Our **LR*Quant*** defines learnable smoothing parameters and introduces a novel loss function based on cosine similarity so it has become state-of-the-art weight-activation PTQ method. However, due to hardware limitation we have not applied our method to quantize larger LLMs with over 100 billion parameters. This will be added to our list of future work.

Additionally, our **LR*Quant*** is the first to introduce TTA into LLMs quantization which has successfully improved the quantized models' performance on unseen datasets. However, we do not draw inspirations from state-of-the-art TTA methods which makes our performance may not achieve optimal, such as on C4 dataset our TTA scheme still stands behind original results. In future work, we aim to explore ideas from other methods and propose a more effective TTA scheme for LLMs quantization.

Ethics Statement

This paper introduces solutions to the challenges associated with Large Language Models (LLMs) quantization, with the overarching goal of facilitating the widespread adoption and application of LLMs. In the current landscape, ethical concerns tied to LLMs, including the presence of hidden biases encoded in the models, are garnering heightened attention. Following our investigation, we assert that our proposed method does not further amplify the biases and contravene any ethical standards.

Acknowledgement

Miao Zhang was partially sponsored by the National Natural Science Foundation of China under Grant 62306084 and Shenzhen College Stability Support Plan under Grant GXWD20231128102243003. Xuebo Liu was sponsored by CCF-Tencent Rhino-Bird Open Research Fund.

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.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher De Sa. 2023. Quip: 2-bit quantization of large language models with guarantees. *arXiv preprint arXiv:2307.13304*.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

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, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023a. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos,

Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. 2023b. Spqr: A sparse-quantized representation for near-lossless llm weight compression. *arXiv preprint arXiv:2306.03078*.

Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. 2019. Learned step size quantization. *arXiv preprint arXiv:1902.08153*.

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*.

Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac'h, Haonan Li, Kyle McDonell, Niklas Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron, Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. 2023. A framework for few-shot language model evaluation.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*.

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*.

Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. 2021. Accurate post training quantization with small calibration sets. In *International Conference on Machine Learning*, pages 4466–4475. PMLR.

Qingyuan Li, Yifan Zhang, Liang Li, Peng Yao, Bo Zhang, Xiangxiang Chu, Yerui Sun, Li Du, and Yuchen Xie. 2023. Fptq: Fine-grained post-training quantization for large language models. *arXiv preprint arXiv:2308.15987*.

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*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

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. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. 2020. Up or down? adaptive rounding for post-training quantization. In *International Conference on Machine Learning*, pages 7197–7206. PMLR.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Yaofo Chen, Shijian Zheng, Peilin Zhao, and Mingkui Tan. 2022. Efficient test-time model adaptation without forgetting. In *International conference on machine learning*, pages 16888–16905. PMLR.

Shuaicheng Niu, Jiaxiang Wu, Yifan Zhang, Zhiquan Wen, Yaofo Chen, Peilin Zhao, and Mingkui Tan. 2023. Towards stable test-time adaptation in dynamic wild world. *arXiv preprint arXiv:2302.12400*.

Jungwuk Park, Dong-Jun Han, Soyeong Kim, and Jaekyun Moon. 2023. Test-time style shifting: Handling arbitrary styles in domain generalization. *arXiv preprint arXiv:2306.04911*.

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.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106.

Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, and Ping Luo. 2024. Omniquant: Omnidirectionally calibrated quantization for large language models. In *International Conference on Learning Representations*.

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*.

Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno Olshausen, and Trevor Darrell. 2020. Tent: Fully test-time adaptation by entropy minimization. *arXiv preprint arXiv:2006.10726*.

Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. 2017. Growing a brain: Fine-tuning by increasing model capacity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2471–2480.

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.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

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.

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.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

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*.

Appendix

A The Full Algorithm

Algorithm 1 Overall algorithm of **LR*Quant***

Input: full-precision LLM model $\mathcal{M}$; calibration set X; test set X^t
Output: quantized LLM model after TTA

$\mathrm{X}_{fp} = \mathrm{X}_q = \mathrm{X}$
for $\mathcal{B}_i$ in $\mathcal{M}$ **do** ▷ Block-wise quantization
  $\mathrm{X}_{fp} = \mathcal{B}_i(\mathrm{X}_{fp})$
  Init learnable parameters Θ_q and Θ_s ▷ Θ_s with Eq. (7)
  for k in epochs **do**
    for $(\mathrm{x}_q, \mathrm{x}_{fp})$ in $(\mathrm{X}_q, \mathrm{X}_{fp})$ **do**
      $\mathcal{B}_i^{'}$ = Quantize (Smooth$(\mathcal{B}_i, \Theta_s), \Theta_q)$
      $\mathrm{x}_q^{'} = \mathcal{B}_i^{'}(\mathrm{x}_q)$
      loss = MSE$(\mathrm{x}_q^{'}, \mathrm{x}_{fp})$+NLC$(\mathrm{x}_q^{'}, \mathrm{x}_{fp})$ ▷ NLC$(\cdot)$ with Eq. (9)
      loss.backward() ▷ Update Θ_q and Θ_s
    end for
  end for
  $\mathcal{B}_i^q$ = Quantize (Smooth$(\mathcal{B}_i, \Theta_s), \Theta_q)$
  $\mathrm{X}_q = \mathcal{B}_i^q(\mathrm{X}_q)$
end for
$\mathrm{X}_{fp}^t = \mathrm{X}_q^t = \mathrm{X}^t$ ▷ TTA phase
for $(\mathcal{B}_i, \mathcal{B}_i^q)$ in $(\mathcal{M}, \mathcal{M}_q)$ **do**
  if i == len$(\mathcal{M})$ **then** ▷ The last quantized block
    $\mathrm{X}_{fp}^t = \mathcal{B}_i(\mathrm{X}_{fp}^t)$
    Init Θ_q^l and Θ_s^l
    for k in 5 **do**
      for $(\mathrm{x}_q^t, \mathrm{x}_{fp}^t)$ in $(\mathrm{X}_q^t, \mathrm{X}_{fp}^t)$ **do**
        $\mathcal{B}_i^{'}$ = Quantize (Smooth$(\mathcal{B}_i, \Theta_s^l), \Theta_q^l)$
        $\mathrm{x}_q^{t'} = \mathcal{B}_i^{'}(\mathrm{x}_q^t)$
        loss = loss$(\mathrm{x}_q^{t'}, \mathrm{x}_{fp}^t)$.backward()
        loss.backward()
      end for
    end for
    $\mathcal{B}_i^q$ = Quantize (Smooth$(\mathcal{B}_i, \Theta_s^l), \Theta_q^l)$
  else
    continue
  end if
end for
return quantized model $\mathcal{M}_q$

B Hyperparameter Analysis on the Base of LAE

In this paper, we set 2 as the base of our learnable scaling factors initialization function, which is the same with (Li et al., 2023). In addition, we also compare several other bases (e, 5, and 10) to quantize LLaMA-7B at W4A4, and the experimental results (perplexity) are shown in the Table 9.

Base	Wiki-2	PTB	C4	PTB-n	C4-n
2	**11.25**	52.05	**14.14**	99.28	**15.41**
e	11.28	50.13	14.15	**82.95**	15.53
5	11.58	96.46	14.53	157.09	15.87
10	11.37	**47.38**	14.40	85.09	15.72

Table 9: Perplexity comparison of LAE using different bases to calculate initial learnable scaling factors to quantize LLaMA-7B at W4A4.

From the table, it can be observed that using different bases yields similar results. Even setting the base as a learnable parameter does not lead to a significant improvement in performance. Therefore, to save computational resources, we choose 2 as the base which shows relatively better performance.

C Weighted Combination of MSE and NLC Loss

To validate whether the weighted combination of MSE and NLC loss in **LR*Quant*** achieves better performance, we further select different combination ratio for comparison as indicated in Table 10.

Methods	Wiki-2	PTB	C4
0.9MSE+0.1NLC	11.35	73.06	14.23
0.8MSE+0.2NLC	11.42	64.24	14.11
0.6MSE+0.4NLC	11.69	57.17	15.42
MSE+NLC	**11.26**	**42.76**	**13.19**

Table 10: Perplexity comparison of weighted combination on LLaMA-7B at W4A4.

From the results, it can be observed that even with the weighted combination method, the performance does not improve and is even inferior to evenly combination. Therefore, we select an evenly combination of two loss functions for our **LR*Quant***.

D LLaMA and LLaMA-2 Experiments at W6A6

Table 11 and Table 12 shows W6A6 experimental results corresponding to Section 4.2 and Section 4.3 respectively. As illustrated in Table 11, **LR*Quant*** achieves the best performance in most experiments. Additionally, as demonstrated in Table 12, **LR*Quant*** achieves optimal results in some experiments, while in the remaining ones, it obtains near-optimal performance, also demonstrating an advanced level. Combining the experiments for W4A4 in the content, we can conclude that **LR*Quant*** has become the current state-of-the-art method for weight-activation quantization.

E Perplexities on OPT Families

We also evaluate perplexities of our **LR*Quant*** on OPT families (1.3B, 2.7B, 6.7B). The results are shown in Table 13. From the table, we know that our method outperforms all the predefined methods and for another learnable method OmniQuant there can be clearly seen a moderate decrease on each

Bits	Models	Methods	WikiText2	PTB	C4	PTB-new	C4-new
W6A6	LLaMA-7B	LAE	6.04	**30.61**	7.47	**44.44**	7.80
		SmoothQuant	6.15	30.73	7.63	46.17	7.99
		OmniQuant	6.05	30.89	7.54	46.45	7.89
		LR*Quant*	**5.88**	32.56	**7.35**	49.14	**7.67**
	LLaMA-13B	LAE	5.43	22.31	6.98	31.57	7.22
		SmoothQuant	5.49	25.22	7.03	33.99	7.28
		OmniQuant	5.48	21.39	7.03	29.42	7.28
		LR*Quant*	**5.27**	**20.13**	**6.84**	**28.39**	**7.07**
	LLaMA-30B	LAE	4.57	17.36	6.34	25.00	6.54
		SmoothQuant	4.77	18.03	6.48	25.34	6.70
		OmniQuant	4.38	17.56	6.23	25.54	6.41
		LR*Quant*	**4.31**	**16.94**	**6.19**	**24.53**	**6.36**
	LLaMA-2-7B	LAE	5.78	26.04	7.35	**55.95**	7.72
		SmoothQuant	6.21	35.50	7.77	142.53	8.07
		OmniQuant	6.20	37.01	7.76	155.70	8.06
		LR*Quant*	**5.67**	**25.77**	**7.24**	67.09	**7.61**
	LLaMA-2-13B	LAE	5.13	31.90	6.71	59.36	7.03
		SmoothQuant	5.18	31.78	6.76	59.28	7.09
		OmniQuant	5.41	31.32	7.01	57.12	7.41
		LR*Quant*	**5.07**	**29.79**	**6.68**	**54.71**	**6.98**

Table 11: **W6A6** perplexities (**lower is better**) comparison of quantized LLaMA and LLaMA-2 models.

comparison so it proves that our **LR*Quant*** is able to applied for accurately and effectively quantizing OPT models.

F W6A6 Catastrophic Forgetting Experiments for Test-Time Adaptation

Be similar with W4A4 results in Table 8 at Section 4.5, the performance on WikiText2 of models re-calibrated by C4 or PTB is all surpassed by our method on W6A6, where we use C4 or PTB to adapt the last block. The comparison results can be found in Table 14. Thereby, we can assert further that our method relieves catastrophic forgetting through the implementation of our TTA scheme.

G Weight-Only Experiments

Our **LR*Quant*** is mainly focusing on the more challenging weight-activation quantization while quantization methods such as AWQ and GPTQ belong to weight-only quantization. To demonstrate the superiority of our proposed method on hardware-friendly quantization tasks, we also add additional weight-only experiments as shown in Table 15 and Table 16. From results, it can be seen that compared to the two well-known weight-only PTQ methods AWQ and GPTQ, our method still demonstrates significant advantages. Especially under more extreme weight quantization settings like W2A16, where GPTQ and AWQ collapse, RLQuant still maintains a good performance. For zero-shot tasks, our method still outperforms them in most results.

Models	Methods	PIQA	ARC-e	ARC-c	BoolQ	HellaS	WinoG	Avg.
LLaMA-7B	LAE	77.74	64.73	37.79	71.62	55.59	**66.29**	62.29
	SmoothQuant	**77.96**	66.62	**39.07**	70.70	**55.93**	64.16	62.41
	OmniQuant	77.04	**67.67**	38.31	**72.66**	55.43	63.14	62.38
	LR*Quant*	77.36	64.81	38.13	72.50	55.81	65.82	**62.41**
LLaMA-13B	LAE	78.29	71.67	**43.34**	65.74	58.00	**69.13**	64.36
	SmoothQuant	**78.72**	70.74	42.66	65.25	58.27	68.27	63.99
	OmniQuant	78.30	71.60	42.80	65.40	58.10	67.70	63.98
	LR*Quant*	78.29	**71.96**	42.90	**66.88**	**58.64**	68.98	**64.61**
LLaMA-30B	LAE	77.04	70.41	43.25	71.55	61.08	71.34	65.78
	SmoothQuant	76.98	69.61	45.73	71.43	60.28	71.03	65.84
	OmniQuant	79.98	73.74	**46.67**	67.80	**61.88**	71.90	67.00
	LR*Quant*	**80.74**	**74.54**	45.31	**68.44**	61.85	**72.22**	**67.18**
LLaMA-2-7B	LAE	**77.63**	66.83	38.82	**70.55**	55.84	65.35	62.50
	SmoothQuant	77.20	68.39	38.73	70.51	55.85	64.95	62.61
	OmniQuant	77.14	**68.77**	39.93	70.15	55.72	**65.74**	**62.91**
	LR*Quant*	77.36	68.30	**40.27**	69.14	**56.15**	65.43	62.78
LLaMA-2-13B	LAE	78.40	71.75	43.08	68.59	58.61	**71.19**	**65.27**
	SmoothQuant	**78.45**	**73.14**	43.77	**69.96**	58.15	65.19	64.78
	OmniQuant	76.44	70.75	41.72	67.71	57.12	64.88	63.10
	LR*Quant*	78.40	72.18	**43.94**	67.92	**59.16**	68.59	65.03

Table 12: Zero-shot accuracies (**higher is better**) comparison of quantized LLaMA and LLaMA-2 models at **W6A6**.

Bits	Models	Methods	WikiText2	PTB	C4	PTB-new	C4-new
W4A4	OPT-1.3B	FP16	14.62	16.96	14.72	20.29	16.07
		SmoothQuant	126.05	120.43	107.16	128.68	119.11
		LAE	102.12	106.38	76.40	112.53	86.95
		OmniQuant	19.19	23.62	19.45	28.92	21.49
		LR*Quant*	**19.11**	**23.29**	**19.39**	**28.75**	**21.45**
	OPT-2.7B	FP16	12.47	15.11	13.16	17.97	14.34
		SmoothQuant	252.04	207.65	151.90	220.49	169.49
		LAE	423.20	304.95	263.87	246.71	309.16
		OmniQuant	15.19	19.34	16.51	**23.24**	18.14
		LR*Quant*	**14.94**	**19.24**	**16.15**	23.33	**17.83**
	OPT-6.7B	FP16	10.86	13.09	11.74	15.77	12.71
		SmoothQuant	491.34	317.70	238.19	293.54	277.48
		LAE	305.84	235.22	180.49	202.07	214.96
		OmniQuant	12.34	15.44	13.56	18.50	14.88
		LR*Quant*	**12.33**	**15.31**	**13.50**	**18.31**	**14.76**
W6A6	OPT-1.3B	FP16	14.62	16.96	14.72	20.29	16.07
		SmoothQuant	15.42	17.65	15.32	21.30	16.70
		LAE	15.34	17.67	15.20	21.30	16.57
		OmniQuant	14.99	17.42	15.06	20.94	16.42
		LR*Quant*	**14.94**	**17.38**	**15.04**	**20.80**	**16.41**
	OPT-2.7B	FP16	12.47	15.11	13.16	17.97	14.34
		SmoothQuant	12.68	15.35	13.32	18.28	14.51
		LAE	12.96	15.65	13.53	18.67	14.68
		OmniQuant	12.56	15.30	13.27	18.23	14.47
		LR*Quant*	**12.55**	**15.23**	**13.27**	**18.14**	**14.47**
	OPT-6.7B	FP16	10.86	13.09	11.74	15.77	12.71
		SmoothQuant	10.98	13.23	11.85	15.89	12.82
		LAE	11.39	13.69	12.15	16.33	13.08
		OmniQuant	10.94	13.17	11.81	15.81	12.79
		LR*Quant*	**10.93**	**13.16**	**11.81**	**15.79**	**12.78**

Table 13: Perplexities (**lower is better**) comparison of quantized OPT models at **W4A4** and **W6A6**.

Models	Methods	W6A6	
		C4	PTB
LLaMA-7B	reCalib	5.91	5.95
	TTA	**5.88**	**5.88**
LLaMA-13B	reCalib	5.28	5.34
	TTA	**5.27**	**5.27**
LLaMA-2-7B	reCalib	5.71	5.76
	TTA	**5.67**	**5.67**
LLaMA-2-13B	reCalib	5.09	5.09
	TTA	**5.07**	**5.07**

Table 14: Perplexities on original WikiText2 at W6A6 after re-calibrating and *TTA* on test datasets (C4 and PTB). In this experiment, “reCalib" first quantizes models based on WikiText2 and then re-quantizes the model based on test datasets. This table reports their evaluation performance on the original WikiText2 dataset.

Bits	Models	Methods	WikiText2	PTB	C4	PTB-new	C4-new
W4A16	LLaMA-7B	AWQ	5.98	30.74	7.44	47.20	7.75
		GPTQ	5.93	**29.62**	7.47	46.17	7.84
		LR*Quant*	**5.84**	30.93	**7.32**	**45.72**	**7.64**
	LLaMA-13B	AWQ	5.25	22.73	6.80	31.01	7.02
		GPTQ	5.28	20.36	6.86	29.75	7.12
		LR*Quant*	**5.21**	**19.39**	**6.76**	**28.32**	**6.98**
	OPT-1.3B	AWQ	15.22	18.40	15.68	22.06	17.12
		GPTQ	15.44	18.47	15.79	22.04	17.21
		LR*Quant*	**15.05**	**17.65**	**15.27**	**21.23**	**16.70**
	OPT-2.7B	AWQ	13.17	16.17	13.92	19.45	15.13
		GPTQ	13.09	16.09	13.96	19.32	15.21
		LR*Quant*	**12.87**	**15.85**	**13.71**	**18.97**	**14.95**
W2A16	LLaMA-7B	AWQ	9.1e4	7.1e4	9.1e4	7.4e4	8.9e4
		GPTQ	491.62	9.7e3	5.1e3	1.2e4	4.7e3
		LR*Quant*	**16.20**	**121.41**	**24.65**	**389.78**	**26.18**
	LLaMA-13B	AWQ	2.1e5	1.6e5	1.5e5	1.7e5	1.7e5
		GPTQ	364.21	1.4e4	1.3e4	1.7e4	1.3e4
		LR*Quant*	**12.24**	**77.42**	**17.44**	**120.78**	**18.35**
	OPT-1.3B	AWQ	9.5e3	5.9e3	6.4e3	8.2e3	6.7e3
		GPTQ	5.5e3	6.6e3	5.5e3	7.8e3	5.9e3
		LR*Quant*	**48.43**	**91.86**	**69.34**	**100.78**	**76.07**
	OPT-2.7B	AWQ	2.3e4	9.0e3	1.2e4	1.5e4	1.2e4
		GPTQ	6.4e3	7.1e3	6.4e3	7.9e3	6.7e3
		LR*Quant*	**30.59**	**47.57**	**42.49**	**62.19**	**45.05**

Table 15: Perplexities comparison of PTQ methods on weight-only quantization tasks.

Bits	**Models**	**Methods**	PIQA	ARC-e	ARC-c	BoolQ	HellaS	WinoG
W4A16	LLaMA-7B	AWQ	78.12	66.58	37.71	**73.08**	55.20	66.53
		GPTQ	77.69	**67.88**	37.54	69.93	55.38	**66.85**
		LR*Quant*	**78.18**	65.02	**37.79**	72.96	**55.42**	65.35
	LLaMA-13B	AWQ	78.61	**73.82**	43.00	**68.47**	58.41	69.13
		GPTQ	78.61	73.19	**43.43**	66.88	57.83	69.13
		LR*Quant*	**78.99**	73.73	42.66	68.28	**58.42**	**69.29**
	OPT-1.3B	AWQ	71.10	56.81	**25.68**	56.69	**40.81**	57.93
		GPTQ	**71.44**	55.64	24.15	57.98	40.56	57.22
		LR*Quant*	70.95	**56.90**	23.46	**61.07**	40.65	**58.41**
	OPT-2.7B	AWQ	73.01	60.04	**26.96**	59.81	44.88	**61.56**
		GPTQ	**73.39**	59.43	26.88	53.30	44.38	60.85
		LR*Quant*	73.23	**60.82**	26.02	**65.75**	**45.04**	60.22
W2A16	LLaMA-7B	AWQ	53.69	26.30	21.16	46.02	25.38	49.01
		GPTQ	53.64	26.09	21.92	46.42	25.87	49.80
		LR*Quant*	**62.89**	**46.75**	**25.76**	**62.01**	**36.19**	**54.61**
	LLaMA-13B	AWQ	53.21	26.51	23.37	57.15	25.54	49.72
		GPTQ	51.90	25.84	22.52	39.54	26.08	49.17
		LR*Quant*	**68.22**	**57.49**	**27.81**	**64.34**	**43.08**	**58.56**
	OPT-1.3B	AWQ	51.63	24.83	**20.05**	37.82	25.71	48.93
		GPTQ	52.77	25.59	19.28	40.06	25.74	49.80
		LR*Quant*	**60.23**	**43.22**	19.80	**57.98**	**30.71**	**51.78**
	OPT-2.7B	AWQ	53.15	25.04	**21.67**	40.09	25.89	51.06
		GPTQ	51.85	25.38	21.16	42.29	26.05	51.22
		LR*Quant*	**64.04**	**47.81**	20.65	**54.10**	**33.59**	**53.51**

Table 16: Zero-shot accuracies comparison of PTQ methods on weight-only quantization tasks.