Optimize Weight Rounding via Signed Gradient Descent for the Quantization of LLMs

Wenhua Cheng * **and Weiwei Zhang and Haihao Shen and Yiyang Cai**
Xin He and Kaokao Lv and Yi Liu
Intel

Abstract

Large Language Models (LLMs) have demonstrated exceptional proficiency in language-related tasks, but their deployment poses significant challenges due to substantial memory and storage requirements. Weight-only quantization has emerged as a promising solution, significantly reducing memory and storage needs without sacrificing too much performance. In this study, we introduceSignRound, a method that leverages signed gradient descent (SignSGD) to optimize rounding values and weight clipping in just 200 steps. SignRound integrates the advantages of Quantization-Aware Training (QAT) and Post-Training Quantization (PTQ), delivering exceptional results across 2 to 4 bits while minimizing tuning costs and avoiding additional inference overhead. For example, SignRound achieved absolute average accuracy improvements ranging from 6.91% to 33.22% at 2 bits, as measured by the average zero-shot accuracy across 11 tasks. It also demonstrates strong generalization in recent models, achieving near-lossless 4-bit quantization in most scenarios. The source code is publicly available at https://github.com/intel/auto-round.

1 Introduction

In recent years, there has been a significant surge in the adoption of Large Language Models (LLMs), leading to their widespread deployment demand even on devices with constrained resources. However, deploying LLMs on these devices poses significant challenges due to their extensive memory and storage requirements. Additionally, the computational demands of these models create obstacles for real-time applications. Therefore, studying techniques such as quantization is crucial for enabling the efficient deployment of LLMs. Quantization techniques can be broadly categorized into two main types: quantization-aware training (QAT) (Esser et al., 2020; Zhuang et al., 2021; Lee et al., 2021; Liu et al., 2023b) and post-training quantization (PTQ) (Nagel et al., 2019; Xiao et al., 2023; Frantar et al., 2022; Nagel et al., 2020).

QAT involves training the model with quantization in mind, using simulated lower-precision representations to allow the model to learn and adapt to the effects of quantization. This approach often results in better accuracy compared to PTQ. However, QAT has drawbacks, including increased training complexity, longer training times, and the need to tune hyperparameters. The application of QAT to LLMs can be particularly resource-intensive, despite recent efforts (Hu et al., 2021; Dettmers et al., 2023) to improve the efficiency of fine-tuning LLMs.

On the other hand, PTQ directly quantizes the model without any simulated training or fine-tuning. While PTQ is a more straightforward approach, it is susceptible to significant accuracy drops. This underscores the importance of further advancements in PTQ methods to enhance their accuracy preservation capabilities.

Quantization commonly applies to two types of tensors: activations and weights. Quantizing activations for LLMs can be challenging (Wei et al., 2023; Xiao et al., 2023; Bondarenko et al., 2024), making weight-only quantization a more practical option. Moreover, the main bottleneck in generating new tokens for LLMs often arises from memory bandwidth limitations (Kim et al., 2023a), emphasizing the advantage of weight-only quantization.

This study focuses on weight-only quantization. In quantizing weights, a critical step involves rounding, primarily achieved through rounding-to-nearest (RTN). RTN quantizes each weight independently by rounding it to the nearest integer, but it overlooks the relationships between weights and between weights and activations. Adaptive Rounding (Nagel et al., 2020) explored the potential for an enhanced rounding strategy to improve accuracy.

*Correspondence:wenhua.cheng@intel.com

Findings of the Association for Computational Linguistics: EACL 2024, pages 11332–11350

They approached the rounding task by formulating it as a quadratic unconstrained binary optimization problem and approximating the loss using a Taylor series expansion. However, relying solely on the second-order term may not yield accurate results, as rounding can significantly modify weights, making other order terms non-negligible.

We select SignSGD(Balles et al., 2020; Li et al., 2023a; Safaryan and Richtárik, 2021) as our optimization method to approach the optimal rounding solution within a limited number of steps. The motivation behind this choice, which is elaborated in Section 3, stems from the well-defined boundaries of the solution space and the inherent simplicity of the method that necessitates only minimal hyperparameter tuning. Figure 1 provides an overview of our method. Our contributions primarily lie in three aspects:

- We introduce a concise yet effective method for optimizing the weight only quantization, combining the strengths of both QAT and PTQ. Our approach leverages SignSGD to tune the rounding with the weight clipping, without introducing any additional overhead during inference.
- Our empirical results demonstrate a significant performance enhancement compared to recent works across various quantization configurations, ranging from 2-bit to 4-bit.
- We demonstrate that SignRound's performance can be further enhanced by fine-tuning model-specific hyperparameters within a constrained space. Moreover, our method demonstrates strong generalization across various models and delivers nearly lossless results across the majority of scenarios using 4-bit quantization.

2 Related Work

Quantization Aware Training. QAT methods have gained widespread popularity in model compression, as they enable the fine-tuning process (Esser et al., 2020; Zhuang et al., 2021; Lee et al., 2021), often leading to superior accuracy compared to the PTQ method.

Post-training Quantization (PTQ). PTQ methods simplify the quantization process without the need for additional training. (Nagel et al., 2019; Liu et al., 2021; Frantar and Alistarh, 2022; Hassibi et al., 1993; Yao et al., 2021). Given its low resource requirement, PTQ is particularly suitable for the quantization of Large Language Models.

Large Language Models Quantization. Significant strides have been made in addressing the pressing need for quantizing large language models (LLMs). GPT3.int8() (Dettmers et al., 2022) introduces a mixed-precision approach to preserve crucial channels in high precision. AQLM (Mao et al., 2024) builds upon Additive Quantization, a classic algorithm from the Multi-Codebook Quantization family, adapting it to LLM quantization. ZeroQuantV2 (Yao et al., 2024) employs low-rank matrices to enhance model quality recovery. RPTQ (Yuan et al., 2023) addresses range differences between channels by rearranging and quantizing them in clusters. LLM-QAT (Liu et al., 2023b) employs QAT to enhance performance. Some other methods, such as SPIQ (Yvinec et al., 2023b), SmoothQuant (Xiao et al., 2023), and Outlier Suppression+ (Wei et al., 2023), utilize handcrafted equivalent transformations to mitigate quantization errors. These methods rely on the model architecture to fuse the equivalent transformation operations. LRQ (Lee et al., 2024) only needs to learn significantly fewer parameters while enabling the individual scaling of weights, thus boosting the generalization capability of quantized LLMs.

Weight Only Quantization. Weight-only quantization reduces the memory footprint and bandwidth demands by quantizing only the weights while retaining activations in floating-point precision, offering a promising balance between accuracy and compression. GPTQ (Frantar et al., 2022) optimizes weights using the Optimal Brain Surgeon technique (Hassibi et al., 1993), achieving low-bit quantization on LLMs with minimal tuning overhead. AWQ (Lin et al., 2023) follows the equivalent transformation approach with additional tuning in a constrained space, sharing similar limitations with SmoothQuant (Xiao et al., 2023). TEQ (Cheng et al., 2023) and OmniQuant (Shao et al., 2023) both utilize a trainable equivalent transformation, while OmniQuant employs extra weight clip tuning. HQQ (Badri and Shaji, 2023) accelerates quantization for large models by eliminating the need for calibration data, making the quantization process extremely fast. Some other works have incorporated optimization methods with extra inference overhead to improve quantization ac-

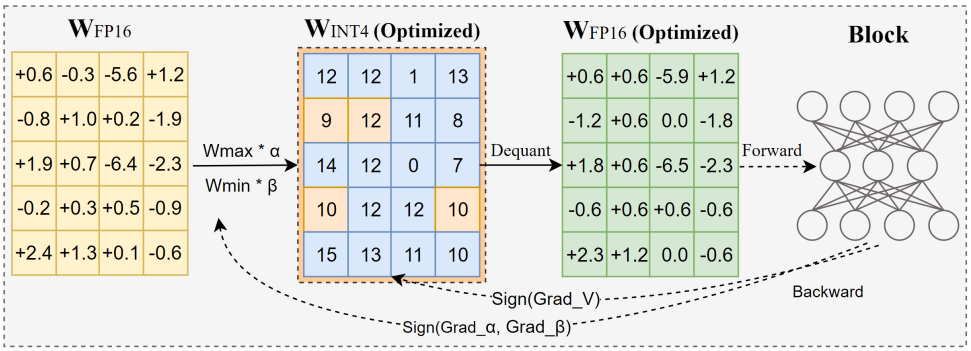


Figure 1: An illustration of SignRound. Unlike the direct rounding in RTN, SignRound performs signed gradient descent to fine-tune the rounding and weight clipping through block-wise output reconstruction. After lightweight forward and backward steps, $\mathbf{W}_{\text{INT4}}$ has been well optimized. Note that Quant and Dequant are two standard operations for quantization and dequantization respectively.

curacy, such as dense-and-sparse decomposition techniques in SqueezeLLM (Kim et al., 2023a) and EasyQuant (Tang et al., 2023), as well as nonuniform quantization methods in NUPES (Yvinec et al., 2023a), QuIP# (Tseng et al., 2024),(Gong et al., 2024), AQLM (Mao et al., 2024), etc. Additionally, FineQuant (Kim et al., 2023b) introduces a straightforward heuristic weight quantization approach that adaptively determines quantization granularity. In this work, we focus on approaches that do not introduce overhead during inference.

Rounding Methods. Adaptive Rounding (Nagel et al., 2020) has already showcased the potential of an advanced rounding strategy to enhance accuracy (Li et al., 2021; Wei et al., 2022). They used the rounding task as a quadratic unconstrained binary optimization problem by approximating the task loss through a Taylor series expansion. However, considering only the second-order term may not yield accurate results. This is because the rounding value gets multiplied by a scaling coefficient during de-quantization, potentially introducing significant weight changes that make other order terms non-negligible. FlexRound (Lee et al., 2023) introduces a more flexible approach to rounding by incorporating element-wise division. However, it's not easily scalable to apply to LLMs due to the needs of specialized hyperparameters for each specific model and task. Furthermore, Oscillation-free (Liu et al., 2023a) suggests that the introduction of learnable parameters might result in weight oscillation problems. AQuant (Li et al., 2022) introduced a dynamic approach where the border becomes a function dependent on the activation value to reduce the quantization error of activation.

Signed Gradient Descent. Signed gradient descent is not commonly utilized and is typically applied in specific scenarios, such as reducing communication costs. This is because signed gradient carries significantly less information compared to original gradient. Recent studies have shed light on the advantages of sign-based methods over gradient descent in certain conditions. Balles et al. (Balles et al., 2020) found that sign-based methods are preferable when the Hessian matrix is concentrated on its diagonal and the maximal eigenvalue is much larger than the average eigenvalue. Li et al. (Li et al., 2023a) investigated a variant of sign-based gradient descent that exhibits faster convergence. Safaryan et al. (Safaryan and Richtárik, 2021) proposed a stochastic sign descent with momentum, which converges under the standard bounded variance assumption with the optimal asymptotic rate. These findings contribute to a better understanding of the potential benefits and applications of signed gradient descent methods.

3 Methodology

We begin with an overview of quantization before delving into the specifics of our approach. The following operations can be utilized to quantize and dequantize the weights $\mathbf{W}$:

$$\widetilde{\mathbf{W}} = s * clip(\left\lfloor \frac{\mathbf{W}}{s} + zp \right\rceil, n, m), n, m \in \mathbb{N} \quad (1)$$

where the rounding operation $\lfloor \cdot \rceil$ is typically performed using the RTN method. Although RTN is a straightforward approach, it quantizes each element independently, which results in the loss of the ability to model the correlation among different weights or activations. The s represents the

Algorithm 1 SignRound

Input: Calibration Data $\mathcal{D}$, learning rate lr, total steps T, Model M, block module m_w with weights w, batch size bs

Output: $best\_V$, $best\_\alpha$, $best\_\beta$

$V \leftarrow 0$, $\alpha \leftarrow 1.0$, $\beta \leftarrow 1.0$, $best\_l \leftarrow maximum$
for $i \leftarrow 0$ to T **do**
  $d \leftarrow$ *draw bs samples*
  $x \leftarrow M(d)_m$ ▷ get the inputs of m
  $y_f \leftarrow m_w(x)$ ▷ get the output of original module
  $\widetilde{w} \leftarrow qdq(w, \alpha, \beta, V)$ ▷ quantize and dequantize w via Eq.3
  $y_q \leftarrow m_{\widetilde{w}}(x)$ ▷ get the output of quantized module
  $loss \leftarrow mse(y_q, y_f)$ ▷ get the loss via Eq.5
  $loss.backward()$
  if $loss < best\_l$ **then**
    $best\_V, best\_\alpha, best\_\beta \leftarrow V, \alpha, \beta$
    $best\_l \leftarrow loss$
  end if
  update α, β and V via SignSGD optimizer
end for

quantization scale, which can be obtained using the following equation, and zp is the zero point.

$$s = \frac{max(\mathbf{W}) - min(\mathbf{W})}{2^{bit} - 1} \tag{2}$$

In order to improve the efficacy of the rounding quantization operation, we build upon prior research (Nagel et al., 2020) by introducing a trainable parameter V to adjust the rounding values.

$$\widetilde{\mathbf{W}} = s * clip(\left\lfloor \frac{\mathbf{W}}{s} + zp + \mathbf{V} \right\rfloor, n, m), n, m \in \mathbb{N} \tag{3}$$

Additionally, following recent works (Lin et al., 2023; Shao et al., 2023), we introduce two additional trainable parameters, denoted as $\alpha \in [0, 1]$ and $\beta \in [0, 1]$, to fine-tune the scale of weight clipping. These parameters are incorporated into the equations as follows:

$$s = \frac{max(\mathbf{W}) * \alpha - min(\mathbf{W}) * \beta}{2^{bit} - 1} \tag{4}$$

These modifications enable a more adaptable quantization process. We utilize block-wise output reconstruction to train these parameters via optimizer, thus framing the optimization as follows.

$$\min_{\alpha,\beta,\mathbf{V}} \|\mathbf{WX} - \widetilde{\mathbf{W}}\mathbf{X}\|_F^2 \tag{5}$$

where $\mathbf{X}$ is the input of the block and $||\cdot||_F$ denotes the Frobenius norm.

Our method distinguishes itself primarily by leveraging SignSGD, which optimizes parameters based on the sign of the gradients as follows:

$$\mathbf{W}_{t+1} = \mathbf{W}_t - lr_t * sign(\mathbf{g}_t) \tag{6}$$

where t represents the step, lr is the learning rate and g denotes the gradient. The motivation is detailed below. ***Firstly***, the optimal values for up and down rounding typically reside in a large region rather than a single float, as only the threshold for altering the rounding value is significant. This eliminates the necessity for the gradient magnitude to converge precisely to a single point. ***Secondly***, due to the confined boundaries, i.e.$[-0.5, 0.5]$ for rounding and $[0, 1]$ for weight clipping, SignSGD allows efficient navigation of this space within a limited number of steps. In contrast, optimizers like Adam (Kingma and Ba, 2014) may struggle due to significant variations in gradient magnitude, making it challenging to converge to the optimal value within a restricted number of steps. ***Thirdly***, SignSGD is inherently intuitive, facilitating easy adjustment of the step size (learning rate). For example, we employed the same optimizer hyperparameters across all experiments unless explicitly stated, consisting of 200 steps and a learning rate of 5e-3 with linear weight decay. Based on Eq. 6, the maximum adjustment for each parameter is the sum of the learning rates over all steps, that is, $200 \times 0.005/2 = 0.5$. As a result,the adjustment can cover a range of [-0.5,0.5] when initialized at 0 for rounding, and a range of [0.5,1.0] when initialized at 1 and clipped to ≤ 1.0 for weight clipping, which works well in practice. ***Fourth***, SignSGD distinguishes itself by its lightweight design, demanding fewer memory and computational resources than optimizers like Adam(Kingma and Ba, 2014).

Figure 1 provides an illustration of our approach. And the Pseudocode 1 presents more details of SignRound.

4 Experiments

This section presents a comprehensive evaluation of SignRound from multiple perspectives. We begin with a brief overview of the LLM architectures

Config	Method	Mistral-7B	V2-7B	V2-13B	V2-70B
	16 bits	63.30	57.98	61.42	66.12
W4G-1	RTN	58.84	55.49	60.46	65.22
	GPTQ	61.37	56.76	59.79	65.75
	AWQ	61.36	57.25	60.58	**66.28**
	HQQ	58.40	46.05	46.82	57.47
	Omni	60.52	56.62	60.31	65.80
	Ours	**62.33**	**57.48**	**61.20**	66.27
	Ours*	**62.64**	**57.52**	**61.23**	66.27
W4G128	RTN	62.36	56.92	60.65	65.87
	GPTQ	62.32	56.85	**61.00**	66.22
	AWQ	62.16	57.35	60.91	66.23
	HQQ	**62.75**	57.41	60.65	66.06
	Omni	62.18	57.30	60.51	66.02
	Ours	62.62	**57.57**	60.85	**66.39**
	Ours*	**62.87**	**57.97**	60.90	**66.41**

Config	Method	Mistral-7B	V2-7B	V2-13B	V2-70B
	16 bits	63.30	57.98	61.42	66.12
W3G128	RTN	58.20	53.81	58.57	64.08
	GPTQ	59.91	54.14	**59.58**	65.08
	AWQ	59.96	55.21	58.86	65.12
	HQQ	59.33	54.31	58.10	64.80
	Omni	58.53	54.72	59.18	65.12
	Ours	**60.43**	**56.68**	59.44	**65.31**
	Ours*	**60.96**	**56.68**	**59.78**	**65.59**
W2G128	RTN	30.52	29.94	33.51	38.14
	GPTQ	39.61	35.37	42.46	28.47
	AWQ	30.06	30.10	32.16	32.23
	HQQ	31.41	29.87	35.28	37.42
	Omni	32.17	40.74	46.55	51.31
	Ours	**52.71**	**48.64**	**53.46**	**61.69**
	Ours*	**53.01**	**50.34**	**54.16**	**61.77**

Table 1: Average accuracies (↑) across 11 tasks, as detailed in Section 4.1, for LLaMA and Mistral models at W2-W4. 'Our*' denotes the highest accuracy achieved among the 8 hyperparameter choices, outlined in Section 4.2, whereas for the 70B model, we tested only a few options.

Config	Method	Mmlu	Lamb.	Hella.	Wino.	Piqa	Truth.	Open.	Boolq	RTE	ARC-e	ARC-c.	Avg.
W4G-1	16 bits	61.35	75.68	61.27	74.03	80.79	28.03	32.80	83.67	67.51	80.81	50.34	63.30
	RTN	55.92	66.10	59.01	71.35	80.14	24.85	29.00	79.17	57.76	77.95	45.99	58.84
	GPTQ	58.22	73.45	59.47	**74.03**	**80.20**	26.93	31.00	81.50	64.98	78.24	47.01	61.37
	AWQ	57.20	71.45	59.21	73.64	79.43	25.34	30.40	**82.69**	**68.95**	79.25	47.44	61.36
	HQQ	52.65	66.58	59.09	70.56	79.60	23.13	27.80	80.03	59.57	77.02	46.33	58.40
	Omni	57.52	70.00	60.27	72.93	79.87	23.99	30.80	81.53	63.90	78.54	46.42	60.52
	Ours	**59.52**	**73.76**	**60.75**	73.32	80.09	**27.17**	**33.00**	82.02	66.07	**80.47**	**49.49**	**62.33**
	Ours*	**60.00**	73.30	60.57	**74.35**	80.09	**27.91**	32.20	**83.52**	67.51	79.92	**49.66**	**62.64**
W2G128	16 bits	61.35	75.68	61.27	74.03	80.79	28.03	32.80	83.67	67.51	80.81	50.34	63.30
	RTN	23.45	0.14	27.43	49.64	54.30	24.24	15.20	38.69	51.99	29.08	21.59	30.52
	GPTQ	25.23	30.47	38.28	53.83	64.91	24.11	17.40	58.29	50.90	47.77	24.57	39.61
	AWQ	25.38	0.00	25.71	52.01	51.58	23.99	17.60	37.83	47.29	26.98	22.27	30.06
	HQQ	23.35	0.85	27.77	51.62	56.69	**26.68**	15.80	40.55	53.43	28.62	20.14	31.41
	Omni	23.24	5.38	29.38	49.72	56.09	26.32	16.60	41.99	52.71	32.11	20.39	32.17
	Ours	**40.46**	**58.61**	**50.87**	**62.90**	**75.84**	24.85	**22.80**	**78.56**	**57.04**	**70.88**	**37.03**	**52.71**
	Ours*	**43.72**	**59.75**	**51.87**	**64.25**	75.14	24.72	**23.60**	75.78	55.23	**71.80**	**37.20**	**53.01**

Table 2: Detailed accuracies(↑) across 11 tasks(0-shot) of Mistral models at W4G-1 and W2G128. 'Our*' denotes the highest accuracy achieved among the 8 hyperparameter choices, outlined in Section 4.2. Appendix C provides more detailed data.

and tasks included in our assessment. Next, we provide a detailed comparison between our method and several existing approaches, emphasizing the unique advantages of SignRound. Furthermore, we conduct ablation studies to reinforce the efficacy of our choices and investigate the sensitivity of hyperparameters. Lastly, we evaluate the generation ability of our method across various recent models. The tuning cost comparisons are provided in Appendix A.

4.1 Experimental Settings

Evaluation and Tasks. We evaluate multiple language tasks to address the task-agnostic setting. Specifically, we present the average accuracy results for 11 zero-shot tasks, including HellaSwag (Zellers et al., 2019), WinoGrande (Sakaguchi et al., 2021), PIQA (Bisk et al., 2020), LAMBADA (Paperno et al., 2016), TruthfulQA (Lin et al., 2022), OpenBookQA (Mihaylov et al., 2018), BoolQ (Clark et al., 2019), RTE (Dagan et al., 2010), ARC-Easy, ARC-Challenge (Clark et al., 2018), and MMLU (Hendrycks et al., 2020). We use lm-eval-harness (Gao et al., 2023) for all the above tasks. Furthermore, we complement our evaluation with perplexity (PPL) analysis on Wikitext2 (Merity et al., 2016), PTB (Marcus et al., 1993), and C4 (Raffel et al., 2020), following the

Model	Method	Steps	Mistral-7B	V2-7B	V2-13B
W4G-1	Flex	200	58.93	56.10	60.06
		1000	60.62	56.98	60.29
		5000	60.94	57.49	60.69
	Ada	200	58.30	55.06	59.86
		1000	58.38	55.05	59.92
	Ours	200	62.33	57.48	61.20
		200*	**62.64**	**57.52**	**61.23**
W2G128	Flex	200	30.10	30.01	30.66
		1000	30.16	31.26	32.29
	Ada	200	30.74	30.21	30.36
		1000	30.84	30.30	30.02
	Ours	200	52.71	48.64	53.46
		200*	**53.01**	**50.34**	**54.16**

Table 3: Comparing with some other rounding methods, the average accuracies (↑) across 11 tasks (detailed in Section 4.1) for Mistral and LLaMA models at W4G-1 and W2G128.

implementation[1] of GPTQ and Wikitext2 (Merity et al., 2016) using lm-eval-harness (Gao et al., 2023). However, we argue that perplexity is notably influenced by outliers, as illustrated in Table 14 for different algorithms. This susceptibility likely arises from the mathematical expression $\mathrm{PPL}(X) = \exp\left(-\frac{1}{t}\sum_{i=1}^{t}\log p_\theta(x_i|x_{<i})\right)$, where assigning a low probability to even one token can significantly inflate the perplexity score. Consequently, we prioritize the accuracy of the 11 tasks mentioned above as the primary metric, with perplexity data serving as supplementary reference.

Quantization Configurations. In alignment with GPTQ (Frantar et al., 2022), our focus is specifically on weight-only quantization, targeting the linear layers within transformer blocks. Layers such as the embedding layer and typically the last linear layer like 'lm-head' are excluded from the quantization process. Our evaluation primarily centers on W4G-1, W4G128, W3G128 and W2G128 configurations, where W4 indicates quantizing weights with 4 bits and G represents finer-grained grouping as described in (Park et al., 2022; Frantar et al., 2022). We adopt asymmetric quantization. To mitigate overfitting on the WikiText and C4 datasets, for all the methods that need calibration, we randomly select 512 calibration samples with the same seed from the readily available pile-10k dataset [2] , which comprises the first 10k samples from pile (Gao et al., 2020). We used a sequence length of 2048 for calibration, while for other methods, we adhere to their official settings.

Large Language Models. We compare different algorithms on commonly used models such as LLaMA-V1 (Touvron et al., 2023a), LLaMA-V2 (Touvron et al., 2023b), and Mistral-7B-v0.1 (Jiang et al., 2023). Our comparison covers a wide range of LLM parameters, ranging from 7B to 70B, to ensure comprehensive coverage and analysis.

SignRound Hyperparameters. Unless explicitly stated, the tuning process involved adjusting each block for 200 steps with a learning rate of 5×10^{-3}, a batch size of 8, and linear learning rate decay. Additionally, we employed automatic mixed precision (AMP) to accelerate the tuning.

4.2 Comparing With Recent Methods

In this section, we compare our methods with those that have already demonstrated remarkable results and impose no additional overhead on our tested models in weight-only quantization for LLMs, including GPTQ (Frantar et al., 2022), AWQ (Lin et al., 2023), HQQ (Badri and Shaji, 2023), OmniQuant (Shao et al., 2023) with a naive method RTN.

To ensure fair comparison as much as possible, we enabled act-order and true-sequential in GPTQ and also activated static_group in scenarios with group_size. The notation GPTQ$^+$ indicates that we adjusted the random seed or data pre-processing to address issues related to the non-positive definite Hessian matrix or other issues. For OmniQuant(Shao et al., 2023), we adhere to the official settings, which include running for 20 epochs including W2G128 for saving time and disabling 'let'. We conducted calibration tests using sample sizes of 512 and 128, as well as a sample size of 512 with a batch size of 4. Our findings show that using a sample size of 512 typically results in comparable or slightly higher performance for models less than or equal to 13B. Therefore, we present the results based on the sample size of 512. For 70B models, due the the Not a Number (NAN) loss issue and to reduce the tuning cost of OmniQuant, we adopted 128 samples for calibration.

We present the summary results of Mistral-7B and LLaMAV2 in Table 1, detailed results of Mistral-7B in Table 2, and additional detailed results are provided in Appendix C due to space constraints. In summary, our approach demonstrated superior performance compared to GPTQ

[1] https://github.com/IST-DASLab/gptq

[2] https://huggingface.co/datasets/NeelNanda/pile-10k

Config	Model	2.5e-3	5e-3	7.5e-3	1e-2	1.25e-2	1.5e-2	1.75e-2	2e-2	SignSGD
W4G-1	Mistral-7B	61.82	61.16	61.30	60.69	60.80	61.07	61.53	61.23	**62.33**
	V2-7B	56.79	57.45	57.09	57.28	56.88	57.24	57.40	57.10	**57.48**
	V2-13B	60.58	60.73	60.76	60.86	61.02	60.79	61.06	60.85	**61.20**
W2G128	Mistral-7B	37.12	40.37	41.11	42.02	42.86	43.55	43.44	42.44	**52.71**
	V2-7B	42.26	44.64	45.08	45.04	45.15	43.13	38.71	35.73	**48.64**
	V2-13B	47.81	50.01	49.55	50.80	48.67	51.94	38.28	34.67	**53.46**

Table 4: Comparison of Adam optimizer with various learning rates against the SignSGD optimizer.. The average accuracies(↑) across 11 tasks (detailed in Section 4.1) for Mistral and LLaMA models at W4G-1 and W2G128.

Config	Mistral-7B	V2-7B	V2-13B	Mistral-7B	V2-7B	V2-13B
	W4G-1			W2G128		
RTN	58.84	55.49	60.46	30.52	29.94	33.51
Weight clip only	61.10	**57.41**	60.10	46.60	40.53	49.77
Rounding only	**61.62**	56.74	**60.64**	**52.32**	**49.14**	**54.41**
Default	**62.33**	**57.48**	**61.20**	**52.71**	48.64	53.46

Table 5: Ablation study of round tuning and weight clip tuning. The average accuracies(↑) across 11 tasks(detailed in Section 4.1) for Mistral and LLaMA models at W4G-1 and W2G128.

(Frantar et al., 2022), achieving scores of 30/32, AWQ (Lin et al., 2023) with 27/32, HQQ (Badri and Shaji, 2023) with 15/16, and OmniQuant (Shao et al., 2023) with a score of 29/32 across LLaMAV1/LLaMAV2/Mistral-7B on various quantization settings, including W4G-1, W4G128, W3G128, and W2G128. These evaluations were based on the average accuracies of 11 zero-shot tasks.

It's worth noting that as the bit depth decreases, the advantages of SignRound become more notable. For example, as shown in Table 2, SignRound could yield absolute average accuracy improvements ranging from 6.91% to 33.22% at W2G128.

Moreover, we can enhance the performance by tuning the model's hyperparameters from a selection of eight choices, denoted as ours*. These choices include steps (200, 1000), weight clip learning rate (1.0/steps, 2.0/steps), and the option to either enable or disable quantized inputs, which refers to utilizing the output from the previous quantized block or the previous original block.

4.3 Comparing with Rounding Methods

In this section, we conduct a comparative analysis between SignRound, FlexRound(Lee et al., 2023), and AdaRound(Nagel et al., 2020). Notably, during the experiment, there is no formal official implementation available for FlexRound and AdaRound for LLMs. Hence, we reference the implementations [3] [4] for further details. However, it's important to highlight that due to the lack of AMP support and other optimizations, the implementation is notably slow, especially when adhering to the official settings, which involve tuning 5000 steps, as presented in Table 9. Therefore, our comparison is limited to models of size 13B or smaller. We set the learning rate to 2e-4 for LLaMA-v2-7b and Mistral-7B, and 1e-4 for LLaMA-v2-13b to align with the official settings as closely as possible. As shown in Table 3, SignRound achieves better results in just 200 steps compared to the 5000 steps required by other rounding methods.

4.4 Ablation Studies

SignSGD versus Adam. To validate the effectiveness of SignSGD, Table 4 compares it with the Adam optimizer (Kingma and Ba, 2014). SignSGD employs a fixed learning rate of 5e-3 throughout all experiments, comprising 200 steps, with linear weight decay. For Adam, we explored learning rates ranging from 2.5e-3 to 2e-2. We choose to quantize models of 13B or less with W4G-1 due to the experiment's cost. SignSGD demonstrated a distinct advantage in average accuracy metrics across 11 tasks, which demonstrate the unique advantage of signed gradient descent in this scenario.

Round and Weigh Clip Tuning. To validate the contributions of rounding tuning and weight clip tuning, we conducted ablation studies on three mod-

[3] https://openreview.net/forum?id=-tYCaP0phY_
[4] https://github.com/quic/aimet

Model	Seqlen_512	Samples_128	Batch_4	Steps_100	Steps_1000	LR_1e-2	Default
Mistral-7B	60.32	61.82	61.78	61.06	**62.58**	61.27	62.33
V2-7B	**57.91**	56.41	57.21	57.10	57.19	55.89	57.48
V2-13B	60.88	60.87	**61.21**	60.80	61.01	61.03	61.20

Table 6: Ablation study of hyperparameter sensitivity. The average accuracies(↑) across 11 tasks(detailed in Section 4.1) for LLaMA models at W4G-1.

Model	Method	Average Acc	Variation %
Gemma-2b	BF16	53.69	-
	Ours	53.40	-0.54%
Llama-2-7b-chat-hf	FP16	59.01	-
	Ours	58.97	-0.07%
Llama-3-8B-Instruct	BF16	63.52	-
	Ours	63.12	-0.63%
Mistral-7B-Instruct-v0.2	BF16	66.47	-
	Ours	66.21	-0.39%
Mixtral-8x7B	BF16	66.98	-
	Ours	66.33	-0.97%
Mixtral-8x7B-Instruct	BF16	70.00	-
	Ours	69.77	-0.33%
Phi-3-mini-4k-instruct	BF16	66.62	-
	Ours	66.33	-0.44%

Table 7: The average accuracies(↑) across 11 tasks(detailed in Section 4.1)) with 1000 steps for LLMs at W4G128. Table 15 provides the detailed data.

els with two quantization configurations. As shown in Table 5, each component provides benefits over RTN, with rounding tuning offering greater advantages. However, when combined, weight clip tuning can sometimes result in lower accuracy in certain cases at W2G128.

Hyperparameters Sensitivity. To validate the sensitivity of hyperparameters in SignRound, we conducted ablation studies on sequence length for calibration, the number of samples for calibration, tuning batch size, tuning steps, and tuning learning rate. The results are presented in Table 6. Overall, our default hyperparameters achieved balanced results.

4.5 Generalization to Other Models

To assess the generalization of our method on LLMs, we evaluate SignRound on various mainstream LLMs such as Gemma (Team et al., 2024), Phi (Li et al., 2023b), Mistral (Jiang et al., 2023), Mixtral (Jiang et al., 2024) and Llama3 (Touvron et al., 2024). Table 7 demonstrated that all int4 models maintained an accuracy drop within 1% of FP16 or BF16 accuracy by employing 1000 tuning steps and model wise hyperparameters among 4 choices detailed in Section 4.1. The detailed results are provided in Appendix C. Notably, the generalization experiments utilized an updated version (0.4.0+) of lm-eval-harness (Gao et al., 2023) and real quantized models, which may result in minor discrepancies compared to other benchmark data.

5 Conclusions

In this paper, we introduce SignRound, an efficient and concise approach for optimizing weight rounding in the quantization of large language models. SignRound employs signed gradient descent for tuning rounding value and weight clipping in 200 steps, completing the quantization of LLAMA-V2-70B in approximately 2.5 hours. Our extensive experiments show that SignRound outperforms other quantization methods across various models and weight bits in the majority of scenarios. Additionally, SignRound shows promising generation capabilities in recent models and achieves enhanced performance through model-specific hyperparameter tuning.

6 Limitations

Despite the advantages, we observed a noticeable gap in accuracy performance for ultra-low bit quantization, particularly with 2-bit quantization, compared to the original model. This challenge could potentially be addressed by exploring non-uniform quantization and mixed-precision quantization, which we leave for future work.

7 Ethics Statement

Our research aims to advance knowledge in LLM quantization. SignRound utilizes open-source models and publicly available datasets, and is not tied to particular applications, requiring only minimal fine-tuning steps on the original models. This ensures that the technical details of our method carry no potential ethical implications. We acknowledge the contributions of the creators and maintainers of these resources and provide citations to the original sources.

References

Hicham Badri and Appu Shaji. 2023. Half-quadratic quantization of large machine learning models.

Lukas Balles, Fabian Pedregosa, and Nicolas Le Roux. 2020. The geometry of sign gradient descent. *arXiv preprint arXiv:2002.08056*.

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.

Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. 2024. Quantizable transformers: Removing outliers by helping attention heads do nothing. *Advances in Neural Information Processing Systems*, 36.

Wenhua Cheng, Yiyang Cai, Kaokao Lv, and Haihao Shen. 2023. Teq: Trainable equivalent transformation for quantization of llms. *arXiv preprint arXiv:2310.10944*.

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. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936.

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*.

Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. 2010. Recognizing textual entailment: Rational, evaluation and approaches–erratum. *Natural Language Engineering*, 16(1):105–105.

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

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

Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. 2020. Learned step size quantization. In *International Conference on Learning Representations*.

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

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

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.

Zhuocheng Gong, Jiahao Liu, Jingang Wang, Xunliang Cai, Dongyan Zhao, and Rui Yan. 2024. What makes quantization for large language model hard? an empirical study from the lens of perturbation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18082–18089.

Babak Hassibi, David G Stork, and Gregory J Wolff. 1993. Optimal brain surgeon and general network pruning. In *IEEE international conference on neural networks*, pages 293–299. IEEE.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt.

2020. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

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

Albert Q Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, et al. 2024. Mixtral of experts. *arXiv preprint arXiv:2401.04088*.

Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W Mahoney, and Kurt Keutzer. 2023a. Squeezellm: Dense-and-sparse quantization. *arXiv preprint arXiv:2306.07629*.

Young Jin Kim, Rawn Henry, Raffy Fahim, and Hany Hassan Awadalla. 2023b. Finequant: Unlocking efficiency with fine-grained weight-only quantization for llms. *arXiv preprint arXiv:2308.09723*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Jung Hyun Lee, Jeonghoon Kim, Se Jung Kwon, and Dongsoo Lee. 2023. Flexround: Learnable rounding based on element-wise division for post-training quantization. *arXiv preprint arXiv:2306.00317*.

Jung Hyun Lee, Jeonghoon Kim, June Yong Yang, Se Jung Kwon, Eunho Yang, Kang Min Yoo, and Dongsoo Lee. 2024. Lrq: Optimizing post-training quantization for large language models by learning low-rank weight-scaling matrices. *arXiv preprint arXiv:2407.11534*.

Jung Hyun Lee, Jihun Yun, Sung Ju Hwang, and Eunho Yang. 2021. Cluster-promoting quantization with bit-drop for minimizing network quantization loss. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5370–5379.

Xiuxian Li, Kuo-Yi Lin, Li Li, Yiguang Hong, and Jie Chen. 2023a. On faster convergence of scaled sign gradient descent. *IEEE Transactions on Industrial Informatics*.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. 2023b. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*.

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

Zhengyi Li, Cong Guo, Zhanda Zhu, Yangjie Zhou, Yuxian Qiu, Xiaotian Gao, Jingwen Leng, and Minyi Guo. 2022. Efficient activation quantization via adaptive rounding border for post-training quantization. *arXiv preprint arXiv:2208.11945*.

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*.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252.

Shih-Yang Liu, Zechun Liu, and Kwang-Ting Cheng. 2023a. Oscillation-free quantization for low-bit vision transformers. In *International Conference on Machine Learning*, pages 21813–21824. PMLR.

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

Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. 2021. Post-training quantization for vision transformer. *Advances in Neural Information Processing Systems*, 34:28092–28103.

Yu Mao, Weilan Wang, Hongchao Du, Nan Guan, and Chun Jason Xue. 2024. On the compressibility of quantized large language models. *arXiv preprint arXiv:2403.01384*.

Mitch Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. In *International Conference on Learning Representations*.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2381–2391.

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.

Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. 2019. Data-free quantization through weight equalization and bias correction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1325–1334.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc-Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. 2016. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1525–1534.

Gunho Park, Baeseong Park, Se Jung Kwon, Byeongwook Kim, Youngjoo Lee, and Dongsoo Lee. 2022. nuqmm: Quantized matmul for efficient inference of large-scale generative language models. *arXiv preprint arXiv:2206.09557*.

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.

Mher Safaryan and Peter Richtárik. 2021. Stochastic sign descent methods: New algorithms and better theory. In *International Conference on Machine Learning*, pages 9224–9234. PMLR.

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. 2023. Omniquant: Omnidirectionally calibrated quantization for large language models. In *The Twelfth International Conference on Learning Representations*.

Hanlin Tang, Yifu Sun, Decheng Wu, Kai Liu, Jianchen Zhu, and Zhanhui Kang. 2023. Easyquant: An efficient data-free quantization algorithm for llms. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

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

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*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2024. Meta llama 3: The most capable openly available llm to date.

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

Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. 2022. QDrop: Randomly dropping quantization for extremely low-bit post-training quantization. In *International Conference on Learning Representations*.

Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. 2023. Outlier suppression+: Accurate quantization of large language models by equivalent and effective shifting and scaling. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

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, Zhen Dong, Zhangcheng Zheng, Amir Gholami, Jiali Yu, Eric Tan, Leyuan Wang, Qijing Huang, Yida Wang, Michael Mahoney, et al. 2021. Hawq-v3: Dyadic neural network quantization. In *International Conference on Machine Learning*, pages 11875–11886. PMLR.

Zhewei Yao, Xiaoxia Wu, Cheng Li, Stephen Youn, and Yuxiong He. 2024. Exploring post-training quantization in llms from comprehensive study to low rank compensation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19377–19385.

Zhihang Yuan, Lin Niu, Jiawei Liu, Wenyu Liu, Xinggang Wang, Yuzhang Shang, Guangyu Sun, Qiang Wu, Jiaxiang Wu, and Bingzhe Wu. 2023. Rptq: Reorder-based post-training quantization for large language models. *arXiv preprint arXiv:2304.01089*.

Edouard Yvinec, Arnaud Dapogny, and Kevin Bailly. 2023a. Nupes: Non-uniform post-training quantization via power exponent search. *arXiv preprint arXiv:2308.05600*.

Edouard Yvinec, Arnaud Dapogny, Matthieu Cord, and Kevin Bailly. 2023b. Spiq: Data-free per-channel static input quantization. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3869–3878.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. Hellaswag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Bohan Zhuang, Mingkui Tan, Jing Liu, Lingqiao Liu, Ian Reid, and Chunhua Shen. 2021. Effective training of convolutional neural networks with low-bitwidth weights and activations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):6140–6152.

A Quantization Cost

Table 8 compares the quantization costs of different methods, with all measurements conducted on a single NVIDIA A100 GPU with 80GB of memory. We ensure each evaluation process exclusively occupies one GPU, but CPU and other resources may be shared among different processes due to limited resources. For SignRound, we disabled low_gpu_mem_usage in our implementation to achieve faster tuning, albeit with higher memory usage. Despite this, LlaMA-2-70B was still able to run on an A100 GPU with 80GB of memory. Although HQQ is exceptionally fast, our methods outperform others in terms of speed. Table 9 also compares the costs between FlexRound, Adaptive Round, and our method.

Model	GPTQ	AWQ	HQQ	Omni.	Ours
Llama-2-7B	1821	1328	19	10255	1041
Llama-2-13B	3266	2630	30	18186	1918
Llama-2-70B	18517	13586	119	35694	9116

Table 8: Quantization cost in seconds at W4G-1 for LLaMAV2. Align with the accuracy experiments, OmniQuant 70b is tested with 128 calibration samples, while all the others are tested with 512 samples.

B View of distribution of tuned parameters

Figure 2 illustrates the distribution of the magnitudes of $\mathbf{V}$ in Eq.3 and α, β in Eq. 4 for Mistral-7B-v0.1 and Llama-2-7B at W4G-1. The results indicate that the distribution is flat for most layers, except for a few layers at the beginning and the end.

Model	FlexRound	AdaRound	Ours
Mistral-7B-V0.1	9369	9332	1045
Llama-2-7B	9628	9701	1041
Llama-2-13B	17583	17865	1918

Table 9: Quantization Time (seconds) of Rounding Methods at W4G-1 with 200 steps for LLaMAV2 Models and Mistral-7B.

C More results

We present the detailed LLMs generalization results in Table 15, the accuracy is within 1% of the 16 bit benchmark after simple fine tuning on different types of models. The detailed accuracy results for 11 tasks using the LLaMA and Mistral models, ranging in size from 7B to 70B, at W2-W4 are provided in Tables 10, 11, 12 and 13. The detailed perplexity (PPL) results are shown in Table 14. Overall, SignRound demonstrates a clear advantage in accuracy tasks, particularly in ultra-low bit quantization, achieving state-of-the-art performance compared to several popular weight quantization methods. In terms of perplexity (PPL), SignRound outperformed all other methods in 83 out of 124 scenarios, demonstrating its advantages. However, we observed that several quantization algorithms, including SignRound, exhibit sensitivity across different models and tasks. The reason for this sensitivity is detailed in Section 4.1.

Model	Method	Mmlu	Lamb.	Hella.	Wino.	Piqa	Truth.	Open.	Boolq	RTE	ARC-e	ARC-c.	Avg.
Mistral-7B	16 bits	61.35	75.68	61.27	74.03	80.79	28.03	32.80	83.67	67.51	80.81	50.34	63.30
	RTN	55.92	66.10	59.01	71.35	80.14	24.85	29.00	79.17	57.76	77.95	45.99	58.84
	GPTQ	58.22	73.45	59.47	**74.03**	**80.20**	26.93	31.00	81.50	64.98	78.24	47.01	61.37
	AWQ	57.20	71.45	59.21	73.64	79.43	25.34	30.40	**82.69**	**68.95**	79.25	47.44	61.36
	HQQ	52.65	66.58	59.09	70.56	79.60	23.13	27.80	80.03	59.57	77.02	46.33	58.40
	Omni	57.52	70.00	60.27	72.93	79.87	23.99	30.80	81.53	63.90	78.54	46.42	60.52
	Ours	**59.52**	**73.76**	**60.75**	73.32	80.09	**27.17**	**33.00**	82.02	66.07	**80.47**	**49.49**	**62.33**
	Ours*	**60.00**	73.30	60.57	**74.35**	80.09	**27.91**	32.20	**83.52**	67.51	79.92	**49.66**	**62.64**
V2-7B	16 bits	42.69	73.90	57.15	68.90	78.07	25.21	31.40	77.74	62.82	76.35	43.52	57.98
	RTN	36.87	67.96	55.63	68.51	76.82	**26.19**	30.60	73.64	58.84	74.07	41.30	55.49
	GPTQ	39.66	**71.92**	55.89	68.03	77.58	25.09	30.20	76.67	62.09	75.55	41.72	56.76
	AWQ	**40.24**	71.20	56.26	**69.61**	76.93	26.07	**32.60**	**77.31**	63.18	75.00	41.30	57.25
	HQQ	28.94	43.96	48.43	59.43	71.82	23.62	24.80	52.11	53.79	64.90	34.73	46.05
	Omni	39.82	71.45	55.76	67.56	76.88	25.09	30.80	76.15	64.98	74.12	40.19	56.62
	Ours	39.97	71.63	**56.52**	68.43	**77.91**	25.70	31.60	76.18	**65.70**	**76.01**	**42.58**	**57.48**
	Ours*	**40.85**	**72.75**	56.01	67.88	77.86	25.34	31.80	76.39	**66.43**	75.88	41.55	**57.52**
V2-13B	16 bits	52.86	76.77	60.04	72.14	79.05	25.95	35.20	80.55	65.34	79.38	48.38	61.42
	RTN	50.37	74.35	59.12	71.98	**79.00**	24.85	33.00	**81.77**	64.98	79.08	46.59	60.46
	GPTQ	51.14	75.37	59.14	72.06	78.02	25.34	32.20	80.46	62.09	77.36	44.54	59.79
	AWQ	51.16	**75.98**	59.51	70.80	78.40	25.21	**34.60**	78.26	**66.79**	79.12	46.59	60.58
	HQQ	35.92	49.54	46.27	58.01	72.47	23.99	19.80	61.77	51.26	62.84	33.19	46.82
	Omni	51.01	75.45	59.48	71.74	78.94	24.60	33.20	77.37	66.07	78.75	46.76	60.31
	Ours	**52.30**	75.96	**59.79**	**72.30**	78.84	**25.58**	34.00	80.15	**66.79**	**79.38**	**48.12**	**61.20**
	Ours*	52.29	**76.15**	59.73	71.90	78.51	25.21	34.40	80.24	**67.51**	79.34	**48.21**	**61.23**
V2-70B	16 bits	66.23	79.64	64.77	77.98	82.15	30.60	37.20	83.70	67.87	82.70	54.44	66.12
	RTN	63.85	77.62	63.38	76.72	81.50	28.89	**37.80**	83.39	68.23	81.99	54.10	65.22
	GPTQ	64.81	79.27	63.86	76.87	81.61	31.46	36.40	82.23	70.04	**82.53**	54.18	65.75
	AWQ	65.08	78.77	64.14	77.11	81.45	30.48	37.20	83.64	**72.92**	82.49	**55.80**	**66.28**
	HQQ	56.45	66.74	53.67	73.32	76.50	25.58	33.40	67.95	61.73	72.90	43.94	57.47
	Omni	64.40	79.20	63.91	76.95	81.94	**31.70**	37.60	82.35	69.31	82.24	54.18	65.80
	Ours	**65.43**	**79.55**	**64.47**	**78.06**	**82.10**	30.60	36.40	**83.91**	71.12	**82.53**	54.78	66.27
V1-7B	16 bits	32.74	73.53	56.94	70.01	78.67	22.03	34.60	75.08	66.43	75.25	41.81	57.01
	RTN	31.34	70.02	55.35	69.77	77.69	20.32	32.60	73.43	59.57	74.45	41.30	55.08
	GPTQ	29.06	71.08	55.11	70.01	77.37	20.93	32.20	72.69	63.90	74.66	41.64	55.33
	AWQ	**33.33**	70.81	55.98	68.27	78.07	21.18	31.40	74.37	64.62	74.03	41.21	55.75
	Omni	32.52	**72.13**	55.87	**70.17**	78.35	**22.77**	32.80	75.05	66.07	**75.13**	40.19	56.46
	Ours	31.80	71.96	**56.57**	69.53	**79.00**	21.91	**33.20**	**75.72**	**66.79**	74.83	**43.09**	**56.76**
V1-13B	16 bits	44.21	76.21	59.92	72.77	79.16	25.70	33.20	77.89	70.76	77.40	46.42	60.33
	RTN	39.57	70.93	58.82	71.98	78.02	24.85	32.00	**78.20**	66.43	75.67	44.62	58.28
	GPTQ$^+$	40.01	74.67	58.92	71.03	78.45	**26.44**	**33.60**	77.09	68.23	76.85	44.97	59.12
	AWQ	**44.56**	74.13	59.13	71.27	**78.94**	25.83	33.20	76.42	66.06	76.89	**46.67**	59.37
	Omni	43.66	75.59	59.36	**72.38**	78.89	25.34	32.20	75.99	**69.68**	**77.10**	45.65	59.62
	Ours	43.94	**75.82**	**59.51**	72.22	78.78	25.70	32.80	77.34	67.51	76.47	**46.67**	**59.71**
V1-30B	16 bits	55.14	77.55	63.33	75.85	81.12	28.27	36.00	82.78	66.79	80.39	52.90	63.65
	RTN	53.05	75.65	62.08	74.82	80.09	25.95	35.80	81.87	63.54	79.76	50.26	62.08
	GPTQ	53.04	77.22	61.95	73.80	80.69	27.29	34.60	81.07	66.06	78.79	49.15	62.15
	AWQ	54.13	76.77	62.78	74.11	**81.07**	**27.78**	35.00	**82.66**	67.15	79.97	51.71	63.01
	Omni	53.43	77.64	62.73	**75.30**	80.58	26.56	35.40	82.51	**67.87**	79.76	50.51	62.93
	Ours	**54.72**	**77.84**	**62.91**	75.06	80.69	26.68	**36.40**	82.60	66.79	**80.13**	**52.13**	**63.27**
V1-65B	16 bits	59.79	79.12	64.53	77.35	81.23	27.91	38.00	84.86	69.68	81.36	52.82	65.15
	RTN	58.74	76.42	64.12	**76.72**	81.01	**29.25**	**38.60**	84.13	70.40	80.72	**51.88**	64.73
	GPTQ$^+$	59.10	78.17	63.78	75.69	**81.34**	28.27	38.40	83.76	68.59	**80.98**	51.62	64.52
	AWQ	58.86	77.37	63.86	76.56	80.85	28.27	35.20	83.94	**71.48**	78.75	50.94	64.19
	Omni	**59.59**	**79.16**	64.03	75.93	**81.99**	27.05	36.80	**84.65**	**71.48**	**80.98**	51.79	**64.86**
	Ours	59.21	**79.16**	**64.37**	**76.64**	81.34	26.81	37.80	84.40	69.68	**80.98**	51.79	64.74

Table 10: Accuracies(↑) across 11 tasks(0-shot) of LLaMA and Mistral models at W4G-1. The notation GPTQ$^+$ indicates that we adjusted the random seed or data pre-processing to address issues related to the non-positive definite Hessian matrix or other issues.

Model	Method	Mmlu	Lamb.	Hella.	Wino.	Piqa	Truth.	Open.	Boolq	RTE	ARC-e	ARC-c.	Avg.
Mistral-7B	16 bits	61.35	75.68	61.27	74.03	80.79	28.03	32.80	83.67	67.51	80.81	50.34	63.30
	RTN	59.72	74.44	**61.06**	73.40	80.36	27.17	**32.60**	83.67	64.62	79.63	49.32	62.36
	GPTQ	59.17	74.52	60.37	**74.90**	**80.58**	26.68	31.00	83.33	**67.15**	79.67	48.12	62.32
	AWQ	60.20	75.14	60.43	73.80	80.03	27.05	30.40	**84.01**	62.09	**80.39**	**50.26**	62.16
	HQQ	60.02	75.41	60.79	74.11	81.01	27.29	32.60	82.97	66.79	79.92	49.32	62.75
	Omni	59.71	73.94	60.62	73.56	80.36	26.68	30.80	83.58	65.70	80.01	49.06	62.18
	Ours	**60.47**	**75.59**	61.03	73.88	80.09	**27.54**	31.60	83.09	66.07	79.97	49.49	**62.62**
V2-7B	16 bits	42.69	73.90	57.15	68.90	78.07	25.21	31.40	77.74	62.82	76.35	43.52	57.98
	RTN	40.91	72.44	**56.91**	68.35	77.58	24.97	31.20	77.61	56.32	**76.26**	43.52	56.92
	GPTQ	**42.57**	**73.28**	56.36	**69.06**	78.02	25.34	30.20	75.72	57.04	75.63	42.15	56.85
	AWQ	41.00	72.60	56.40	68.98	77.31	**25.70**	**31.60**	**78.75**	58.48	76.14	**43.86**	57.35
	HQQ	41.79	73.20	56.21	68.43	77.58	25.83	31.60	76.09	62.82	75.84	42.15	57.41
	Omni	41.72	73.04	56.59	68.98	77.91	24.97	30.80	75.81	61.37	75.76	43.34	57.30
	Ours	41.82	72.75	56.79	68.67	**78.13**	25.58	30.20	77.49	**63.54**	75.76	42.58	**57.57**
V2-13B	16 bits	52.86	76.77	60.04	72.14	79.05	25.95	35.20	80.55	65.34	79.38	48.38	61.42
	RTN	52.10	76.27	59.77	72.14	78.62	24.72	34.20	80.24	62.09	79.00	**47.95**	60.65
	GPTQ	**52.66**	76.54	59.76	72.14	78.35	25.70	34.00	79.33	**66.43**	78.58	47.53	**61.00**
	AWQ	52.39	**76.89**	**59.97**	**73.24**	**79.00**	25.21	32.60	**80.40**	63.54	79.04	47.70	60.91
	HQQ	52.09	75.74	59.46	72.14	78.45	24.36	33.60	79.17	66.06	79.00	47.01	60.65
	Omni	52.01	76.17	59.53	72.06	78.35	23.87	33.40	80.80	66.07	78.37	47.18	60.51
	Ours	51.92	76.46	59.87	71.67	**79.00**	**25.83**	**35.20**	79.60	63.54	**79.25**	47.01	60.85
V2-70B	16 bits	66.23	79.64	64.77	77.98	82.15	30.60	37.20	83.70	67.87	82.70	54.44	66.12
	RTN	64.91	79.06	63.93	78.14	81.66	30.11	37.00	83.61	68.59	82.79	54.78	65.87
	GPTQ	65.63	79.22	64.45	78.22	81.88	31.09	37.00	84.19	**69.31**	82.79	54.61	66.22
	AWQ	**65.79**	**79.76**	64.48	77.58	**82.32**	30.72	**38.00**	83.06	68.95	82.70	**55.12**	66.23
	HQQ	65.34	79.14	64.56	77.35	81.56	30.48	37.20	83.67	69.31	82.83	55.20	66.06
	Omni	65.30	79.39	64.52	77.51	81.88	30.60	37.40	83.39	68.23	82.91	55.12	66.02
	Ours	65.65	79.49	**64.60**	**78.30**	82.05	**31.58**	37.40	**84.83**	68.95	**82.87**	54.52	**66.39**
V1-7B	16 bits	32.74	73.53	56.94	70.01	78.67	22.03	34.60	75.08	66.43	75.25	41.81	57.01
	RTN	32.63	72.31	56.26	70.01	78.45	20.93	**33.60**	74.74	64.26	74.71	42.75	56.42
	GPTQ	31.16	72.40	55.85	70.09	78.13	**22.28**	30.40	74.65	64.26	74.20	40.19	55.78
	AWQ	**33.42**	**72.95**	56.30	68.75	77.97	21.42	32.80	74.89	62.09	75.00	41.21	56.07
	Omni	31.15	72.35	56.25	69.22	78.35	21.42	33.80	74.74	65.70	74.87	**42.06**	56.36
	Ours	32.15	72.85	**56.45**	**70.17**	**78.51**	**22.28**	32.80	**75.14**	**67.87**	**75.13**	41.89	**56.84**
V1-13B	16 bits	44.21	76.21	59.92	72.77	79.16	25.70	33.20	77.89	70.76	77.40	46.42	60.33
	RTN	42.71	75.26	59.30	72.53	**79.54**	25.95	32.60	76.76	65.34	**76.98**	45.82	59.34
	GPTQ$^+$	42.65	75.41	59.51	72.93	79.33	24.97	32.40	77.49	68.23	76.89	45.56	59.58
	AWQ	42.66	75.76	59.50	72.77	78.89	**26.56**	**33.60**	77.46	68.59	76.94	45.48	59.84
	Omni	**43.99**	**76.29**	**59.53**	**73.56**	79.43	25.83	33.20	77.58	67.15	76.64	45.48	59.88
	Ours	42.27	76.17	**59.53**	**73.56**	79.33	25.70	32.80	**78.20**	**70.04**	76.94	**46.25**	**60.07**
V1-30B	16 bits	55.14	77.55	63.33	75.85	81.12	28.27	36.00	82.78	66.79	80.39	52.90	63.65
	RTN	54.24	77.02	62.90	74.35	**80.52**	27.29	34.20	81.96	67.15	80.89	52.05	62.96
	GPTQ	54.20	77.41	62.79	75.14	80.41	27.54	34.60	81.93	67.51	80.05	50.51	62.92
	AWQ	55.14	77.49	63.08	**75.77**	**80.52**	27.29	34.20	**82.87**	67.15	80.43	**52.90**	63.35
	Omni	**55.22**	77.80	**63.09**	75.14	80.30	**28.52**	**36.00**	82.20	**69.31**	**80.81**	52.82	**63.75**
	Ours	54.68	**77.90**	62.93	74.82	80.47	28.15	35.80	82.39	66.79	80.13	51.11	63.20
V1-65B	16 bits	59.79	79.12	64.53	77.35	81.23	27.91	38.00	84.86	69.68	81.36	52.82	65.15
	RTN	59.53	**79.51**	64.63	**77.35**	80.96	27.91	38.40	**84.43**	**71.48**	81.48	52.22	**65.26**
	GPTQ$^+$	**60.47**	78.79	64.45	76.24	81.18	28.03	37.40	83.85	68.95	**81.57**	**53.07**	64.91
	AWQ	59.45	79.31	**64.67**	76.72	**81.56**	**28.15**	38.00	**84.43**	71.12	81.10	52.13	65.15
	Omni	59.27	78.65	64.48	76.87	81.23	27.78	**39.00**	84.13	70.76	**81.57**	**53.07**	65.17
	Ours	58.93	79.22	64.48	77.03	81.28	27.91	38.60	84.31	70.76	81.19	52.22	65.08

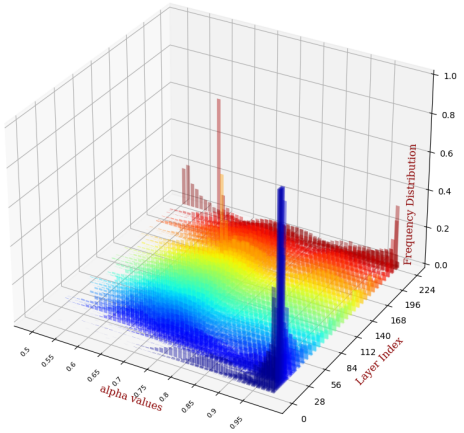
Table 11: Accuracies(↑) across 11 tasks(0-shot) of LLaMA and Mistral models at W4G128. The notation GPTQ$^+$ indicates that we adjusted the random seed or data pre-processing to address issues related to the non-positive definite Hessian matrix or other issues.

Model	Method	Mmlu	Lamb.	Hella.	Wino.	Piqa	Truth.	Open.	Boolq	RTE	ARC-e	ARC-c.	Avg.
Mistral-7B	16 bits	61.35	75.68	61.27	74.03	80.79	28.03	32.80	83.67	67.51	80.81	50.34	63.30
	RTN	53.49	68.74	58.12	68.27	79.33	24.60	29.60	79.97	57.40	76.89	43.77	58.20
	GPTQ	55.84	73.04	57.61	70.24	78.67	24.85	30.80	81.44	**63.54**	77.27	45.65	59.91
	AWQ	55.61	**73.69**	57.86	71.27	**79.82**	**26.07**	29.00	81.10	59.21	**79.00**	**46.93**	59.96
	HQQ	53.97	68.66	58.59	72.22	78.73	25.70	30.00	80.24	63.90	76.81	43.86	59.33
	Omni	54.79	69.34	58.42	68.51	79.38	24.85	28.80	80.15	56.68	77.74	45.14	58.53
	Ours	**57.54**	73.01	**59.60**	**72.85**	79.54	25.70	**31.60**	**81.74**	58.12	78.70	46.33	**60.43**
V2-7B	16 bits	42.69	73.90	57.15	68.90	78.07	25.21	31.40	77.74	62.82	76.35	43.52	57.98
	RTN	34.22	65.96	54.90	67.56	76.28	24.48	30.80	71.68	54.51	72.98	38.57	53.81
	GPTQ	36.11	69.61	53.66	**68.59**	76.01	21.91	27.80	73.43	54.51	73.74	40.19	54.14
	AWQ	35.82	69.90	54.98	67.40	76.01	25.21	29.80	74.68	57.76	74.07	41.64	55.21
	HQQ	34.40	66.64	53.27	67.01	75.46	25.46	28.80	73.58	61.37	72.94	38.48	54.31
	Omni	34.51	69.75	54.42	66.69	76.77	24.24	31.40	73.21	56.68	74.37	39.85	54.72
	Ours	**40.13**	**71.01**	**55.33**	68.27	**76.82**	**25.34**	**32.80**	**75.32**	**60.29**	**75.25**	**42.92**	**56.68**
V2-13B	16 bits	52.86	76.77	60.04	72.14	79.05	25.95	35.20	80.55	65.34	79.38	48.38	61.42
	RTN	48.01	72.33	57.74	70.72	78.07	**25.21**	32.00	77.28	60.65	77.69	44.62	58.57
	GPTQ	49.56	**75.24**	57.83	70.88	**78.56**	24.97	33.40	78.44	**62.82**	77.99	45.65	**59.58**
	AWQ	**49.77**	75.22	58.58	**71.82**	77.75	24.11	**34.20**	**79.97**	53.43	77.95	44.62	58.86
	HQQ	48.40	73.22	57.66	69.77	77.31	24.11	30.60	76.97	60.29	77.15	43.60	58.10
	Omni	47.25	73.67	58.46	70.01	78.40	24.36	33.60	79.79	64.62	77.86	46.16	59.18
	Ours	49.64	75.20	**59.11**	71.59	78.29	24.85	**34.20**	78.47	58.12	**78.58**	**45.82**	59.44
V2-70B	16 bits	66.23	79.64	64.77	77.98	82.15	30.60	37.20	83.70	67.87	82.70	54.44	66.12
	RTN	61.15	77.95	61.98	**77.90**	80.79	29.74	36.00	81.28	64.62	81.10	52.39	64.08
	GPTQ	63.15	79.06	62.94	77.66	81.45	30.72	36.20	81.53	67.87	81.65	**53.67**	65.08
	AWQ	64.09	**79.47**	63.75	76.48	**81.77**	29.74	**37.20**	**82.69**	66.06	81.40	**53.67**	65.12
	HQQ	63.45	78.05	63.12	77.03	81.01	29.38	36.60	82.23	66.43	81.78	53.67	64.80
	Omni	63.18	78.63	63.54	76.48	81.50	30.35	35.80	82.57	70.40	81.02	52.82	65.12
	Ours	**64.94**	78.89	**63.83**	76.56	81.50	**31.21**	**37.20**	81.41	**68.59**	**81.73**	52.56	**65.31**
V1-7B	16 bits	32.74	73.53	56.94	70.01	78.67	22.03	34.60	75.08	66.43	75.25	41.81	57.01
	RTN	28.00	67.67	53.43	66.38	76.50	21.42	31.20	72.72	59.21	70.92	38.31	53.25
	GPTQ	30.16	66.31	53.92	67.48	76.82	21.42	29.60	71.31	59.21	72.22	38.74	53.38
	AWQ	**30.33**	70.19	54.53	68.98	76.71	20.81	31.60	**74.68**	64.62	73.23	38.91	**54.96**
	Omni	28.35	70.54	54.48	68.27	77.48	21.05	29.40	72.29	**66.07**	72.73	37.12	54.34
	Ours	25.85	**70.95**	**55.45**	**69.69**	**77.37**	**21.66**	**32.00**	73.88	60.29	**73.48**	**39.33**	54.54
V1-13B	16 bits	44.21	76.21	59.92	72.77	79.16	25.70	33.20	77.89	70.76	77.40	46.42	60.33
	RTN	34.87	69.65	57.25	70.48	77.31	**26.93**	32.00	71.44	62.82	75.63	43.94	56.57
	GPTQ	35.51	73.08	57.89	70.80	77.37	24.48	31.40	**77.52**	62.82	74.41	43.26	57.14
	AWQ	**40.53**	73.94	57.89	69.53	**78.94**	26.68	33.40	74.83	**65.34**	75.93	45.05	58.37
	Omni	38.35	74.42	57.79	70.80	78.07	26.68	33.20	75.81	**65.34**	75.88	43.69	58.18
	Ours	39.16	**75.22**	**58.64**	**71.59**	**78.94**	25.95	**35.20**	76.30	**65.34**	**76.52**	**45.39**	**58.93**
V1-30B	16 bits	55.14	77.55	63.33	75.85	81.12	28.27	36.00	82.78	66.79	80.39	52.90	63.65
	RTN	52.41	75.08	61.45	74.27	79.87	25.95	33.00	81.38	65.34	79.12	48.89	61.52
	GPTQ	51.39	74.97	60.35	**75.30**	79.60	26.93	34.80	**82.75**	64.62	78.11	48.46	61.57
	AWQ	53.84	76.71	61.94	75.14	80.03	25.34	34.40	81.90	67.15	79.59	**50.77**	62.44
	Omni	53.67	76.95	61.82	74.51	80.14	25.95	34.40	81.10	66.07	79.76	48.21	62.05
	Ours	**54.39**	**77.49**	**62.13**	74.03	**80.47**	**27.30**	**35.00**	**79.76**	**68.59**	79.46	48.98	**62.51**
V1-65B	16 bits	59.79	79.12	64.53	77.35	81.23	27.91	38.00	84.86	69.68	81.36	52.82	65.15
	RTN	57.47	77.43	63.23	75.93	80.41	28.64	**38.40**	82.69	66.43	80.22	**51.19**	63.82
	GPTQ$^+$	57.92	**78.69**	62.98	**76.87**	80.63	27.66	37.60	84.16	68.95	80.89	**51.19**	64.32
	AWQ	**58.87**	77.94	**63.77**	75.37	**80.96**	27.66	36.80	**85.02**	**71.12**	**81.10**	50.34	64.45
	Omni	57.19	77.00	63.15	75.53	80.90	28.15	37.60	83.18	69.68	80.18	50.51	63.92
	Ours	58.30	78.11	63.60	76.56	80.85	**29.50**	37.80	84.80	70.04	80.22	50.68	**64.59**

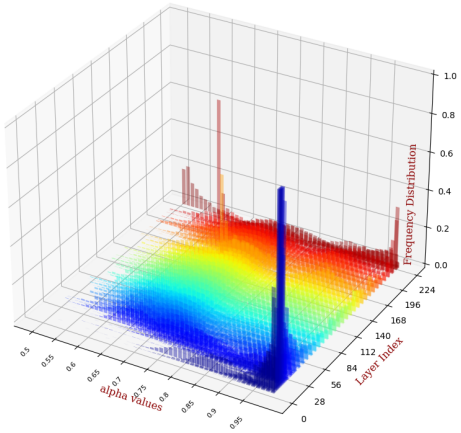
Table 12: Accuracies(↑) across 11 tasks(0-shot) of LLaMA and Mistral models at W3G128. The notation GPTQ$^+$ indicates that we adjusted the random seed or data pre-processing to address issues related to the non-positive definite Hessian matrix or other issues.

Model	Method	Mmlu	Lamb.	Hella.	Wino.	Piqa	Truth.	Open.	Boolq	RTE	ARC-e	ARC-c.	Avg.
Mistral-7B	16 bits	61.35	75.68	61.27	74.03	80.79	28.03	32.80	83.67	67.51	80.81	50.34	63.30
	RTN	23.45	0.14	27.43	49.64	54.30	24.24	15.20	38.69	51.99	29.08	21.59	30.52
	GPTQ	25.23	30.47	38.28	53.83	64.91	24.11	17.40	58.29	50.90	47.77	24.57	39.61
	AWQ	25.38	0.00	25.71	52.01	51.58	23.99	17.60	37.83	47.29	26.98	22.27	30.06
	HQQ	23.35	0.85	27.77	51.62	56.69	**26.68**	15.80	40.55	53.43	28.62	20.14	31.41
	Omni	23.24	5.38	29.38	49.72	56.09	26.32	16.60	41.99	52.71	32.11	20.39	32.17
	Ours	**40.46**	**58.61**	**50.87**	**62.90**	**75.84**	24.85	**22.80**	**78.56**	**57.04**	**70.88**	**37.03**	**52.71**
V2-7B	16 bits	42.69	73.90	57.15	68.90	78.07	25.21	31.40	77.74	62.82	76.35	43.52	57.98
	RTN	23.98	0.02	26.04	49.49	52.50	24.85	15.20	41.01	49.10	27.48	19.71	29.94
	GPTQ	23.65	11.72	32.59	55.17	58.32	**25.95**	15.80	52.14	51.99	40.45	21.25	35.37
	AWQ	25.38	0.00	25.69	49.96	52.34	23.75	17.80	37.83	52.71	24.62	21.08	30.10
	HQQ	24.51	0.02	26.06	49.49	53.26	24.72	13.80	37.92	50.90	26.52	21.33	29.87
	Omni	22.97	35.53	40.28	55.88	65.13	22.89	15.60	63.24	53.07	50.13	23.46	40.74
	Ours	**27.20**	**55.25**	**47.35**	**61.01**	**72.96**	24.85	**25.60**	**68.07**	**54.51**	**65.99**	**32.25**	**48.64**
V2-13B	16 bits	52.86	76.77	60.04	72.14	79.05	25.95	35.20	80.55	65.34	79.38	48.38	61.42
	RTN	23.77	7.47	33.08	49.01	57.94	**26.19**	16.00	47.74	53.43	32.03	21.93	33.51
	GPTQ	24.69	45.20	41.06	55.80	67.08	23.26	19.80	54.40	52.35	55.60	27.82	42.46
	AWQ	27.04	0.00	25.80	51.85	52.99	23.62	13.60	62.17	47.29	26.22	23.12	32.16
	HQQ	23.48	8.17	31.27	52.17	61.86	24.85	17.20	50.46	54.51	42.85	21.25	35.28
	Omni	25.53	49.84	46.23	57.93	70.13	24.60	21.80	66.85	55.60	63.22	30.29	46.55
	Ours	**34.33**	**63.92**	**53.35**	**64.33**	**76.17**	25.70	**26.00**	**72.75**	**61.73**	**71.17**	**38.57**	**53.46**
V2-70B	16 bits	66.23	79.64	64.77	77.98	82.15	30.60	37.20	83.70	67.87	82.70	54.44	66.12
	RTN	24.20	20.18	40.88	54.85	63.87	24.11	17.60	43.06	53.07	50.51	27.22	38.14
	GPTQ	23.12	0.00	25.04	49.57	49.51	0.00	27.60	37.83	52.71	25.08	22.70	28.47
	AWQ	24.46	0.00	25.46	51.38	52.50	23.50	14.20	62.17	52.71	25.76	22.35	32.23
	HQQ	23.16	19.46	35.45	56.67	66.00	22.52	20.00	40.46	52.71	52.06	23.12	37.42
	Omni	33.84	61.83	52.44	64.33	74.10	24.48	28.20	71.68	53.07	67.21	33.28	51.31
	Ours	**54.04**	**72.97**	**59.65**	**74.90**	**79.00**	**29.01**	**34.80**	**79.63**	**69.68**	**78.37**	**46.59**	**61.69**
V1-7B	16 bits	32.74	73.53	56.94	70.01	78.67	22.03	34.60	75.08	66.43	75.25	41.81	57.01
	RTN	24.36	0.52	27.24	49.25	54.24	24.24	15.20	39.63	**57.40**	27.86	21.84	31.07
	GPTQ	22.95	12.75	33.36	51.70	60.07	23.99	13.40	48.62	53.07	40.82	21.50	34.75
	AWQ	23.12	0.00	25.37	53.28	52.56	**25.21**	13.80	37.83	52.71	25.63	22.53	30.18
	Omni	23.58	**44.23**	**42.39**	**58.48**	68.82	21.54	20.40	60.80	53.07	59.55	27.56	**43.68**
	Ours	**24.46**	13.53	42.16	56.99	**70.02**	24.60	**25.20**	**62.91**	47.29	**60.90**	**31.74**	41.80
V1-13B	16 bits	44.21	76.21	59.92	72.77	79.16	25.70	33.20	77.89	70.76	77.40	46.42	60.33
	RTN	24.66	4.97	29.67	49.33	57.24	**25.58**	12.40	44.10	53.79	32.07	22.01	32.35
	GPTQ$^+$	26.43	40.48	39.47	58.25	66.97	23.50	18.60	52.78	50.54	51.52	25.00	41.23
	AWQ	27.04	0.00	25.59	50.36	53.05	24.11	15.60	62.17	47.29	25.97	23.21	32.22
	Omni	26.93	56.41	47.67	61.17	73.23	23.38	24.60	68.75	53.07	67.00	33.79	48.73
	Ours	**31.87**	**59.65**	**51.25**	**67.64**	**76.28**	**25.58**	**27.80**	**69.11**	**58.48**	**70.71**	**37.12**	**52.32**
V1-30B	16 bits	55.14	77.55	63.33	75.85	81.12	28.27	36.00	82.78	66.79	80.39	52.90	63.65
	RTN	23.24	5.55	27.22	53.99	56.80	21.79	18.20	51.65	53.07	36.74	21.33	33.60
	GPTQ	30.47	49.93	45.05	61.88	68.88	23.26	22.60	68.29	51.99	60.69	30.72	46.70
	AWQ	27.04	0.00	25.41	50.20	52.94	**24.48**	16.60	62.17	47.29	24.71	23.38	32.20
	Omni	26.89	63.03	52.23	64.64	74.27	23.87	29.20	70.86	54.51	70.45	36.18	51.47
	Ours	**40.83**	**67.92**	**56.73**	**68.90**	**76.17**	24.36	**31.60**	**75.54**	**62.45**	**74.92**	**42.41**	**56.53**
V1-65B	16 bits	59.79	79.12	64.53	77.35	81.23	27.91	38.00	84.86	69.68	81.36	52.82	65.15
	RTN	24.48	32.78	43.59	57.85	67.52	22.89	22.80	61.53	50.54	52.10	28.24	42.21
	GPTQ$^+$	37.06	67.44	53.97	69.46	76.44	24.36	28.00	73.64	60.29	71.34	38.57	54.60
	AWQ	25.38	0.00	25.58	49.96	53.10	24.24	11.00	37.83	52.71	24.96	22.44	29.75
	Omni	27.36	65.94	55.53	68.11	76.99	25.21	29.60	75.69	59.21	69.82	35.07	53.50
	Ours	**47.21**	**72.07**	**60.06**	**73.24**	**78.62**	**25.46**	**34.20**	**80.64**	**62.82**	**77.48**	**46.76**	**59.87**

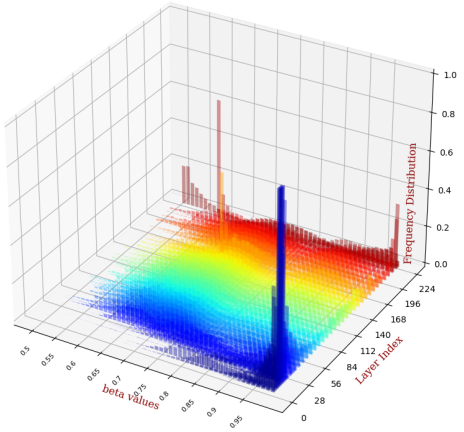
Table 13: Accuracies(↑) across 11 tasks(0-shot) of LLaMA and Mistral models at W2G128. The notation GPTQ$^+$ indicates that we adjusted the random seed or data pre-processing to address issues related to the non-positive definite Hessian matrix or other issues.



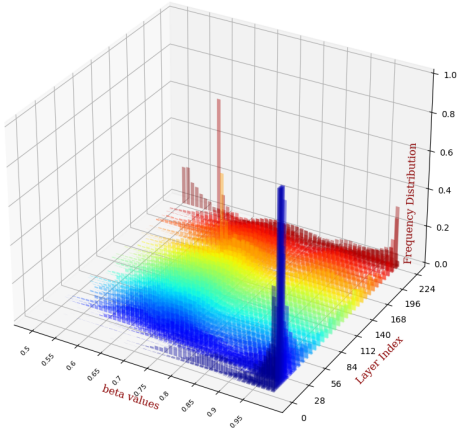
Mistral-7B, alpha values



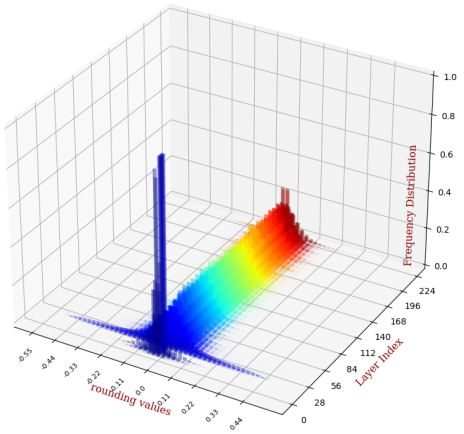
Llama-2-7B, alpha values



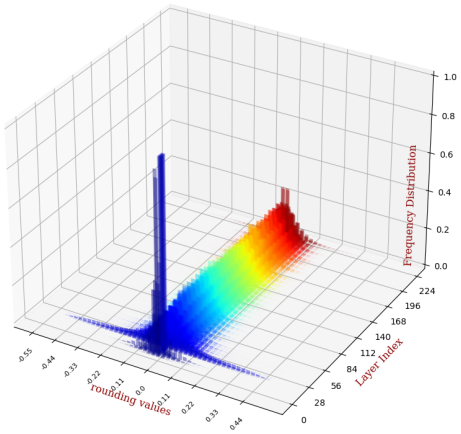
Mistral-7B, beta values



Llama-2-7B, beta values



Mistral-7B, V values



Llama-2-7B, V values

Figure 2: The distribution of the magnitude of $\mathbf{V}$ in Eq. 3 and α, β in Eq. 4 for Mistral-7B-v0.1 and Llama-2-7B at W4G-1, each color in the distribution represents a specific layer index in the models, with blue indicating shallow layers closer to the data layer, and red representing deeper layers.

LLaMA-V2			Wiki2.	Ptb	C4	Wiki.
7B		16 bits	5.47	37.92	7.26	8.79
	W4G-1	RTN	6.12	82.85	8.16	10.06
		GPTQ	5.84	1246	7.82	9.59
		AWQ	**5.81**	**57.09**	**7.70**	**9.42**
		Ours	7.85	3005.52	7.71	10.34
	W4G128	RTN	5.72	65.35	7.58	9.22
		GPTQ	**5.60**	246.28	7.48	9.05
		AWQ	5.61	**42.67**	**7.44**	9.03
		Ours	8.96	473.78	7.50	**9.01**
	W3G128	RTN	6.66	**55.10**	8.98	11.21
		GPTQ	6.32	2245	8.55	10.37
		AWQ	**6.24**	66.57	8.27	10.18
		Ours	8.09	164.90	**8.12**	**9.76**
	W2G128	RTN	4270	9646	4807	1.8e5
		GPTQ	**25.56**	**9429**	**34.87**	**79.65**
		AWQ	2.3e5	2.1e5	1.7e5	1.1e7
		Ours	NAN	NAN	NAN	NAN
13B		16 bits	4.88	50.93	6.73	7.90
	W4G-1	RTN	5.20	60.69	7.14	8.65
		GPTQ	5.12	55.99	7.04	942.3
		AWQ	5.07	55.39	6.96	8.39
		Ours	**5.00**	**51.71**	**6.89**	**8.33**
	W4G128	RTN	4.98	53.69	6.87	8.12
		GPTQ	4.98	52.43	6.85	10.86
		AWQ	4.97	54.18	6.84	**8.08**
		Ours	**4.96**	**51.62**	**6.83**	8.14
	W3G128	RTN	5.52	64.85	7.58	9.27
		GPTQ	5.39	72.96	7.47	334.2
		AWQ	5.30	57.66	7.30	8.81
		Ours	**5.23**	**53.82**	**7.18**	**8.68**
	W2G128	RTN	122.5	1212	131.8	1054
		GPTQ	11.30	**410.9**	15.11	270.6
		AWQ	1.2e5	1.1e5	9.7e4	5.5e6
		Ours	**7.64**	4250	**11.73**	**57.52**
70B		16 bits	3.32	24.25	5.71	4.54
	W4G-1	RTN	3.67	**23.56**	6.01	5.18
		GPTQ	3.57	23.76	5.89	5.00
		AWQ	3.48	24.93	5.85	4.81
		Ours	**3.44**	24.33	**5.81**	**4.78**
	W4G128	RTN	3.46	24.20	5.83	4.78
		GPTQ	3.42	24.01	5.78	4.71
		AWQ	3.41	24.36	**5.77**	4.70
		Ours	**3.40**	**23.69**	**5.77**	**4.68**
	W3G128	RTN	3.98	23.59	6.27	5.77
		GPTQ	3.83	24.78	6.09	5.50
		AWQ	3.73	25.68	6.03	5.31
		Ours	**3.68**	**24.26**	**5.99**	**5.23**
	W2G128	RTN	**27.01**	**758.9**	**47.57**	**298.3**
		GPTQ	NAN	NAN	NAN	NAN
		AWQ	7.2e4	8.1e4	NAN	2.5e6
		Ours	NAN	NAN	NAN	NAN
Mistral			**Wiki2.**	**Ptb**	**C4**	**Wiki.**
7B		16 bits	5.25	35.00	8.38	OOM
	W4G-1	RTN	5.99	44.88	9.47	OOM
		GPTQ	5.57	54.45	8.86	OOM
		AWQ	5.75	**42.21**	9.14	OOM
		Ours	**5.43**	81.67	**8.66**	OOM
	W4G128	RTN	5.42	34.08	8.62	OOM
		GPTQ	5.37	37.53	8.56	OOM
		AWQ	5.37	37.12	8.55	OOM
		Ours	**5.34**	**36.36**	**8.51**	OOM
	W3G128	RTN	6.16	49.97	9.68	OOM
		GPTQ	5.90	49.50	9.30	OOM
		AWQ	5.90	51.01	9.27	OOM
		Ours	**5.66**	**44.50**	**8.96**	OOM
	W2G128	RTN	1375	2351	1015	OOM
		GPTQ	16.59	269.2	22.38	OOM
		AWQ	3.7e4	3.4e4	3.7e4	OOM
		Ours	**8.70**	**86.08**	**12.54**	OOM

LLaMA-V1			Wiki2.	Ptb	C4	Wiki.
7B		16 bits	5.68	41.15	7.34	9.49
	W4G-1	RTN	6.29	48.65	8.12	10.62
		GPTQ	6.13	**47.18**	7.93	10.32
		AWQ	5.97	48.25	7.73	10.11
		Ours	**5.93**	54.84	**7.62**	**9.91**
	W4G128	RTN	5.96	**42.33**	7.70	10.00
		GPTQ	5.90	42.36	7.66	9.91
		AWQ	5.80	44.00	7.50	9.75
		Ours	**5.79**	56.45	**7.49**	**9.74**
	W3G128	RTN	7.01	56.28	9.18	12.11
		GPTQ	6.60	53.75	8.72	11.46
		AWQ	6.32	49.27	8.21	10.81
		Ours	**6.28**	**47.57**	**8.09**	**10.55**
	W2G128	RTN	1847	6574	936.2	1.3e4
		GPTQ	**28.52**	**638.3**	**37.85**	**128.0**
		AWQ	2.6e5	2.8e5	2.9e5	2.1e7
		Ours	641.8	824.9	2533	1876
13B		16 bits	5.09	28.10	6.80	14.06
	W4G-1	RTN	5.53	29.45	7.23	37.17
		GPTQ	5.34	30.23	7.09	13.09
		AWQ	5.25	30.34	7.01	**12.36**
		Ours	**5.21**	**27.81**	**6.93**	113.24
	W4G128	RTN	5.26	28.36	6.94	25.34
		GPTQ	5.19	29.36	6.91	**13.33**
		AWQ	5.19	28.34	6.90	15.25
		Ours	**5.18**	**27.80**	**6.88**	59.09
	W3G128	RTN	5.88	33.10	7.86	44.06
		GPTQ	5.56	32.52	7.48	95.24
		AWQ	5.53	29.63	7.34	22.26
		Ours	**5.45**	**28.13**	**7.21**	**15.44**
	W2G128	RTN	797.7	1695	449.1	1.5e4
		GPTQ	12.13	185.8	NAN	**546.1**
		AWQ	2.8e5	2.6e5	2.4e5	1.6e7
		Ours	**8.36**	**48.93**	**10.64**	1773
30B		16 bits	4.10	23.51	6.13	6.89
	W4G-1	RTN	4.54	25.49	6.54	8.03
		GPTQ	4.41	24.22	6.40	8.50
		AWQ	4.30	**24.20**	6.30	**6.88**
		Ours	**4.23**	27.97	**6.24**	6.90
	W4G128	RTN	4.23	**23.90**	6.26	**7.05**
		GPTQ	4.24	23.92	6.23	7.73
		AWQ	4.22	23.98	6.21	7.29
		Ours	**4.18**	31.38	**6.20**	7.39
	W3G128	RTN	4.87	26.99	6.85	NAN
		GPTQ	4.72	25.14	6.73	8.44
		AWQ	4.61	**25.05**	6.56	**7.84**
		Ours	**4.50**	67.01	**6.47**	7.90
	W2G128	RTN	68.40	566.8	114.2	1192
		GPTQ	9.21	59.75	12.50	**21.21**
		AWQ	2.3e5	2.2e5	2.4e5	1.5e7
		Ours	**7.13**	**55.40**	**12.02**	118.7
LLaMA-V1			**Wiki2.**	**Ptb**	**C4**	**Wiki.**
65B		16 bits	3.53	25.07	5.81	4.96
	W4G-1	RTN	3.92	28.07	6.07	5.60
		GPTQ	3.79	34.82	6.00	5.46
		AWQ	3.72	44.83	5.96	5.30
		Ours	**3.65**	**22.42**	**5.89**	**5.19**
	W4G128	RTN	3.67	25.61	5.90	5.21
		GPTQ	3.64	33.81	5.88	5.17
		AWQ	3.62	**24.46**	**5.87**	5.14
		Ours	**3.61**	35.87	**5.87**	**5.13**
	W3G128	RTN	4.25	50.00	6.33	6.25
		GPTQ	4.05	32.64	6.21	6.03
		AWQ	3.95	**23.48**	6.14	5.83
		Ours	**3.90**	29.15	**6.08**	**5.69**
	W2G128	RTN	15.21	276.7	**20.03**	29.39
		GPTQ	6.85	**37.79**	NAN	12.25
		AWQ	7.3e4	6.7e4	7.4e4	NAN
		Ours	**5.52**	NAN	NAN	**9.25**

Table 14: Perplexity(PPL) (↓) of Wikitext2, PTB, C4 and Wikitext tasks for LLaMA and Mistral models. we follow the source code of GPTQ for wikitext2, PTB and C4 PPL evaluation, while for wikitext, we adopt lm-eval-harness (Gao et al., 2023). NAN indicates not a number, while OOM denotes out of memory.

Model	Method	Mmlu	Lamb.	Hella.	Wino.	Piqa	Truth.	Open.	Boolq	RTE	ARC-e	ARC-c.	Avg.
Gemma-2b	BF16	32.87	63.44	52.73	65.04	76.71	22.03	29.80	69.27	64.26	74.20	40.19	53.69
	Ours	32.97	63.07	51.59	65.43	76.12	22.03	30.00	69.39	63.90	73.53	39.33	53.40
Llama-2-7b-chat-hf	FP16	46.40	71.05	57.80	66.38	76.39	30.23	33.40	79.76	69.68	73.82	44.20	59.01
	Ours	45.45	70.37	57.06	66.14	76.33	30.35	32.60	80.64	72.92	73.36	43.52	58.97
Llama-3-8B-Instruct	BF16	63.86	71.82	57.69	71.43	78.67	36.23	34.00	82.97	67.51	81.52	52.99	63.52
	Ours	63.06	72.00	56.99	72.38	77.97	35.37	33.00	83.09	68.59	80.89	51.02	63.12
Mistral-7B-Instruct-v0.2	BF16	59.06	71.41	66.02	73.95	80.52	52.51	36.00	85.35	70.40	81.61	54.35	66.47
	Ours	58.72	71.41	65.57	73.64	80.47	51.53	34.20	85.41	71.48	81.65	54.35	66.21
Mixtral-8x7B	BF16	68.02	78.27	64.90	76.48	82.48	34.27	35.40	85.23	70.76	84.30	56.66	66.98
	Ours	66.93	78.25	64.59	75.14	82.10	32.19	35.60	84.74	69.31	84.30	56.48	66.33
Mixtral-8x7B-Instruct	BF16	68.85	77.18	67.67	76.87	83.51	49.69	36.80	88.50	71.84	86.99	62.20	70.00
	Ours	68.24	77.90	67.45	77.19	83.35	48.84	37.20	87.83	70.04	87.12	62.29	69.77
Phi-3-mini-4k-instruct	BF16	67.97	68.08	60.64	74.03	80.30	39.53	38.80	86.21	77.98	83.54	55.72	66.62
	Ours	66.59	67.71	59.70	74.59	79.33	37.45	38.80	85.66	79.06	82.70	56.83	66.33

Table 15: The detail accuracies(↑) across 11 tasks(detailed in Section 4.1) with 1000 steps for LLMs at W4G128