PRUNE-QUANTIZE-PRUNE: TOWARDS BETTER JOINT MODEL COMPRESSION BY GETTING FREE SPARSITY FROM SYMMETRIC QUANTIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The deployment of large language models (LLMs) at scale imposes substantial demands on memory bandwidth and computational throughput, making model compression essential for efficient inference. Among post-training compression techniques, 2:4 semi-structured sparsity and low-bit quantization are particularly attractive, yet effectively combining them remains challenging. Existing approaches typically apply pruning and quantization sequentially, either in a prune-then-quantize or quantize-then-prune manner, without explicitly exploiting the additional sparsity naturally induced by quantization. To address this limitation, we propose a three-stage prune-quantize-prune framework tailored to the combination of semi-structured sparsity and symmetric quantization. In the first stage, we introduce a *Quantization-aware Mask Selection* strategy that avoids prematurely enforcing the exact 2:4 constraint and instead reserves room for quantization-induced free sparsity. After applying low-bit symmetric quantization, the resulting additional zeros are explicitly exploited. In the final stage, we re-prune the quantized weights to strictly satisfy the target 2:4 sparsity pattern. To mitigate the reconstruction error introduced by this post-quantization pruning, we further propose a *Grid-constrained Compensation* method, which performs second-order weight compensation directly in the quantized space while preserving the discrete quantization grid. Extensive experimental results on popular LLMs demonstrate that this method outperforms existing state-of-the-art methods.

1 INTRODUCTION

Large language models (LLMs) (DeepSeekAI et al., 2024; Touvron et al., 2023a;b) have become a fundamental infrastructure for modern artificial intelligence, enabling remarkable progress across a range of fields, including natural language understanding, code generation, advanced reasoning, and multimodal applications. However, the exceptional capabilities of LLMs are typically achieved by continuously increasing model size, which results in a substantial memory and computation costs. These growing demands pose challenges for deploying LLMs on resource-constrained devices and also lead to increased operational costs in large-scale data-center environments. As a result, the development of effective post-training compression techniques has emerged as a crucial and urgent area of focus for making LLM deployment more practical and efficient.

Among existing compression techniques, pruning and quantization are two representative and widely adopted approaches. Pruning reduces model redundancy by removing less important weights (Hassibi et al., 1993), thereby decreasing the number of effective parameters. Recent post-training pruning methods introduce activation-aware or second-order criteria to better estimate the contribution of each weight to the layer output reconstruction error (Frantar & Alistarh, 2023; Sun et al., 2024). Quantization, on the other hand, compresses LLMs by representing full-precision weights or activations with low-bit numerical formats. Compared with pruning, quantization directly reduces the bit-width of each retained parameter and is therefore highly effective for memory-efficient deployment. Quantization is well supported by modern hardware accelerators, whereas sparsity generally requires a structured 2:4 pattern to achieve hardware acceleration. Therefore, it is necessary to consider the combined 2:4 sparsity and quantization to effectively leverage hardware acceleration.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Pruning and quantization have garnered significant research attention in rece years. Existing studies primarily focus on whether pruning or quantization should come first (Kim et al., 2026; Harma et al., 2025), providing detailed theoretical analyses. Despite some research consensus that sparse pruning followed by quantization is generally effective, many methods still treat pruning and quantization as two relatively independent operations, often overlooking their interdependencies. The 2:4 sparsity pattern, typically associated with symmetric quantization, also introduces additional sparsity. Given that pruning and quantization are strongly coupled, it is essential to investigate how the effects of quantization should be considered during the pruning stage itself.

Figure 1 shows our motivation. In current pruning (Sun et al., 2024) for LLMs, the importance of a weight is not determined solely by its magnitude. A weight with a small absolute value may still be preserved if its corresponding input activation has a large norm. However, such a small retained weight is more likely to be rounded to zero under the prevalent symmetric quantization. It can be treated as a form of **free sparsity** that can be exploited to reduce the number of weights that need to be explicitly pruned. If a retained weight is likely to be rounded to zero after quantization, then the pruning stage can preserve more full-precision weights in that group and rely on quantization to contribute part of the final sparsity. In this way, the pruning mask becomes aware of the subsequent quantization behavior, and the final sparse-quantized model can better balance sparsity, quantization error, and reconstruction accuracy.

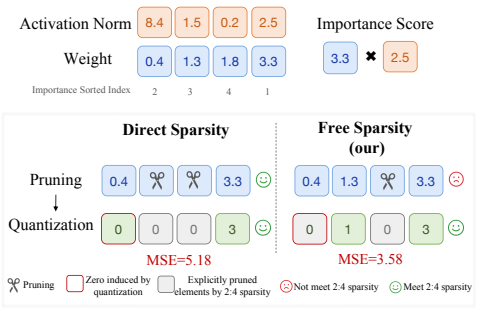

Figure 1: Comparison between our quantization-aware pruning and direct 2:4 sparse + quantization. The direct method may preserve small but important weights during pruning, only for them to be quantized to zero later. Our method leverages this 'free sparsity' to prune fewer weights at the pruning stage, achieving lower reconstruction error while still satisfying 2:4 sparsity after quantization.

Based on this insight, we propose a prune-quantize-prune (PQP) joint compression framework for LLMs. First, we introduce a *Quantization-aware Mask Selection* strategy. Rather than determining the pruning mask solely from the full-precision weights, we perform a trial pruning-quantization pass to estimate the additional zeros induced by low-bit quantization. This information is then used to adjust the pruning mask on the original full-precision weights, allowing groups that are likely to acquire extra zeros after quantization to retain more weights during the initial pruning stage. Second, we propose *Grid-constrained Compensation*. Since the sparsity pattern after quantization may not always be predicted exactly, some groups may still violate the target sparsity type and require an additional pruning step. Second-order Hessian-based compensation, such as OBS (Hassibi et al., 1993), is a widely adopted approach for mitigating the reconstruction error introduced by pruning. However, direct OBS compensation produces continuous weight updates, which can move the already quantized weights away from the discrete quantization grid. To address this issue, we extend OBS to a grid-constrained compensation scheme, where weight updates are restricted to the quantized space while still minimizing the reconstruction error introduced by post-quantization pruning.

Our primary contributions are summarized as follows:

- We identify the extra free sparsity introduced by quantization and propose a prune-quantize-prune compression framework for better combination of semi-structured sparsity and symmetric quantization.
- We design *Quantization-aware mask selection* strategy, where fewer weights are pruned in groups that are likely to produce more zeros after quantization, and introduce a second pruning step on the quantized weights to enforce the desired 2:4 sparsity constraint, and further propose *Grid-constrained Compensation* strategy to mitigate the reconstruction error.
- Empirically, extensive evaluations on prevalent models suggest our method outperforms the SOTA joint pruning and quantization methods.

2 Related Work

2.1 LLM Pruning

Sparsification for LLMs can be broadly categorized into two paradigms. The first involves weight update compensation after pruning, where remaining weights are adjusted to minimize reconstruction error. Unlike conventional training-based methods that demand substantial computational resources, weight adjustment in LLMs is performed locally in a layer-wise manner with efficient closed-form solutions. Pioneered by SparseGPT (Frantar & Alistarh, 2023), which leverages approximate Hessian inverses to compensate residual weights column by column, subsequent work has focused on improving the accuracy of this compensation. The second paradigm involves mask selection without any weight update, where the core challenge lies in designing superior importance criteria to identify the optimal pruning mask. Pioneered by Wanda (Sun et al., 2024), which defines weight importance as the product of weight magnitude and the corresponding input activation norm, this approach significantly outperforms naive magnitude-based pruning. Subsequent works have further refined importance estimation from various perspectives, including outlier-aware metrics that account for the prevalent outlier activations (Yin et al., 2024) in LLMs. Furthermore, for semi-structured sparsity patterns such as NVIDIA's N:M sparse format, a line of work employs learnable mask selection (Zhou et al., 2021), directly optimizing sparse patterns in an end-to-end manner.

2.2 LLM Quantization

Model quantization represents high-precision weights and activations with a low-precision format to reduce the model's memory footprint and computational cost. Recent LLM quantization research has focused on post-training quantization. Scale-based quantization methods optimize per-channel or per-group scaling factors to balance the dynamic ranges of weights and activations. AWQ (Lin et al., 2024) protects critical weight channels by scaling them according to activation magnitudes while SmoothQuant (Xiao et al., 2023) further generalizes this by migrating quantization difficulty from activation outliers to weights. Rotation-based quantization leverages orthogonal or invertible linear transformations to reshape weight/activation distributions into more quantile-friendly forms, suppressing outliers. QuIP (Tseng et al., 2024a) and its variant QuIP# (Tseng et al., 2024b) employ random Hadamard rotations to induce incoherence between weights. QuaRot (Ashkboos et al., 2024) introduces static orthogonal rotations to decorrelate features, while SpinQuant (Liu et al., 2025) and FlatQuant (Sun et al., 2025) advance this by learning rotation matrices. Error-compensated quantization minimizes reconstruction loss via weight iterative updates and second-order guidance, directly compensating quantization residuals. GPTQ (Frantar et al., 2023) pioneered this approach by using the Hessian matrix to propagate and compensate quantization errors column-by-column. GPTAQ (Li et al., 2025) extends GPTQ with asymmetric calibration, while BoA (Kim et al., 2025) refines the compensation objective by modeling attention-aware error propagation.

2.3 Joint Sparsification and Quantization

Recent studies have shown that sparsification and quantization should not be regarded as two independent compression operators, because their errors can interact and accumulate in a non-trivial manner. Early joint-compression methods attempted to optimize the two dimensions within a unified framework. More recently, JSQ explicitly studies joint sparsification and quantization for large language models, introducing a sparsity metric that bridges pruning and quantization and an activation editor to alleviate outlier-induced degradation (Guo et al., 2024). OBR aligns pruning and quantization errors through a second-order Hessian objective and group-wise error compensation (Guo et al., 2026). In parallel, several works have analyzed whether the order of applying sparsity and quantization matters. The theory that the two operations are non-orthogonal and shows that applying sparsity before quantization usually minimizes additional computational error (Harma et al., 2025) are adopted in subsequent work (Guo et al., 2026). PQQP further formalizes compression-order optimization, suggesting that weaker perturbations should precede stronger ones in joint compression pipelines (Kim et al., 2026). These findings indicate that simple sequential pipelines, such as prune-then-quantize or quantize-then-prune, may be suboptimal under aggressive compression. Therefore, a more principled joint optimization framework is needed to simultaneously determine the sparse structure and quantized weights under a unified reconstruction or error-compensation objective.

3 PREREQUISITES

Optimal Brain Surgeon (OBS) Framework. Optimal Brain Surgeon (OBS) (Hassibi et al., 1993) estimates the sensitivity of network parameters through a Taylor expansion of the loss function. Its core idea is to prune the parameter that yields the smallest increase in the objective while compensating for the induced perturbation by optimally adjusting the remaining weights. For one row of weights $\mathbf{w}$, the local loss change caused by a perturbation $\Delta\mathbf{w}$ can be approximated as

$$\Delta\mathcal{L}(\Delta\mathbf{w}) \approx \frac{1}{2}\Delta\mathbf{w}^{\mathsf{T}}\mathbf{H}\Delta\mathbf{w}, \tag{1}$$

where $\mathbf{H} = 2\mathbf{X}\mathbf{X}^{\mathsf{T}}$ is the Hessian matrix. When a single weight w_q is removed, OBS solves the constrained optimization problem under the contrains $\mathbf{e}_q^{\mathsf{T}}\Delta\mathbf{w} + w_q = 0$ where $\mathbf{e}_q$ is the one-hot vector corresponding to the q-th weight. The resulting loss increase and optimal compensation are

$$\Delta\mathcal{L}_q = \frac{1}{2}\frac{w_q^2}{[\mathbf{H}^{-1}]_{qq}}, \quad \Delta\mathbf{w}_q = -\frac{w_q}{[\mathbf{H}^{-1}]_{qq}}\mathbf{H}^{-1}_{:,q}. \tag{2}$$

OBC (Frantar et al., 2022) further extends this problem to group-wise pruning. Let $\mathcal{P}$ denote a set of weights to be removed simultaneously, and let $\mathbf{w}_\mathcal{P}$ be the corresponding sub-vector. The group pruning constraint is $\mathbf{E}_\mathcal{P}^{\mathsf{T}}\Delta\mathbf{w} + \mathbf{w}_\mathcal{P} = \mathbf{0}$, where $\mathbf{E}_\mathcal{P}$ selects the coordinates indexed by $\mathcal{P}$. Solving this problem gives the group-wise OBS loss increase and weight correction:

$$\Delta\mathbf{w}_\mathcal{P} = -\mathbf{H}^{-1}_{:,\mathcal{P}}\left(\mathbf{H}^{-1}_{\mathcal{P},\mathcal{P}}\right)^{-1}\mathbf{w}_\mathcal{P}. \tag{3}$$

Symmetric Quantization. Symmetric quantization maps full-precision weights onto a signed integer grid centered at zero. For a b-bit signed quantizer, the integer range is usually given by $q_{\min} = -2^{b-1}$ and $q_{\max} = 2^{b-1} - 1$. Given a weight vector or a quantization group $\mathbf{w}$, a positive scale factor s is used to determine the spacing of the quantization grid. The scale can be obtained by different calibration strategies, such as max-based, percentile-based, or reconstruction-aware methods.

With the scale s, each weight is quantized by scaling, rounding, and clipping:

$$q_i = \mathrm{clip}\left(\left\lfloor\frac{w_i}{s}\right\rceil, q_{\min}, q_{\max}\right), \quad \widehat{w}_i = sq_i, \tag{4}$$

where $\lfloor\cdot\rceil$ denotes rounding to the nearest integer, and $\widehat{w}_i$ is the dequantized weight.

Since the integer grid is centered at zero and explicitly contains $q_i = 0$, symmetric quantization can exactly represent zero values. In particular, if a retained full-precision weight is sufficiently close to zero, *i.e.*,

$$|w_i| < \frac{s}{2}, \tag{5}$$

it will be rounded to $q_i = 0$, and its dequantized value becomes $\widehat{w}_i = 0$. Therefore, symmetric quantization not only preserves the zeros introduced by pruning, but may also convert some small-magnitude retained weights into exact zeros, leading to additional sparsity after quantization.

4 METHODOLOGY

In this section, the proposed method is introduced, consisting of three stages:

- **Pruning**: A trial 2:4 sparsity and quantization process is performed. The number of zeros introduced by quantization serves as a guide for the pruning process. Groups with more zeros require less pruning in the first phase.
- **Quantization**: The pruned weights undergo quantization, introducing additional free sparsity and refining the weight values.
- **Pruning**: A second pruning step is performed to correct any remaining groups that do not meet the 2:4 sparsity, with weight updates based on error correction using Hessian compensation.

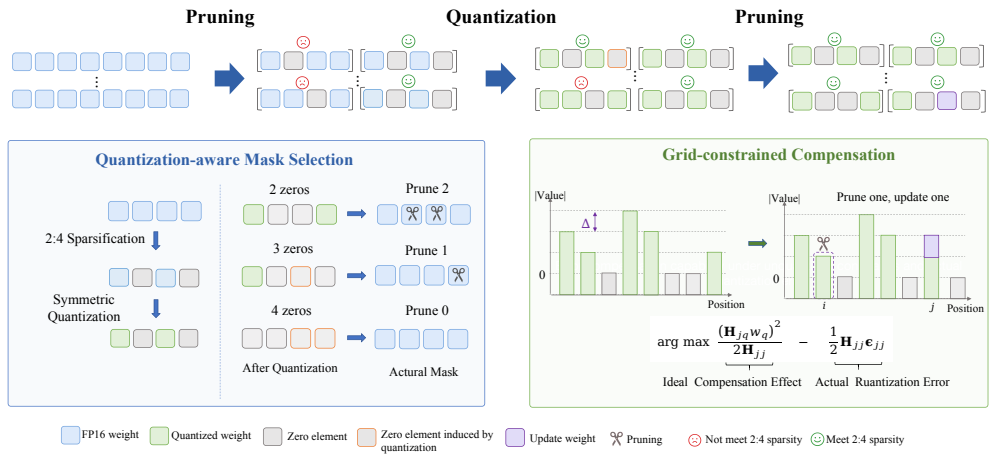

Figure 2: Illustration of the pruning-quantization-pruning process. (1) **Pruning**: *Quatization-aware Mask Selection* is performed. The number of zeros introduced by quantization serves as a guide for the pruning process. Groups with more zeros require less pruning in the first phase. (2) **Quantization**: The pruned weights are quantized, compressing the values and introducing additional free sparsity. This step helps refine the weight values, but some groups may still not meet the 2:4 sparsity due to zeros introduced by quantization. (3) **Pruning**: Pruning is performed again to address any remaining groups that do not meet the 2:4 sparsity requirement. Weights are updated based on our proposed *Grid-constrained Compensation*, ensuring all groups satisfy the 2:4 sparsity constraint.

4.1 First Pruning: Leave Space for Free Sparsity

This component aims to make the pruning mask aware of the extra sparsity introduced by quantization: (1) A trial pass with standard 2:4 pruning followed by quantization is first performed to obtain the quantized sparse weights. (2) The final pruning mask is then determined according to the number of zeros in each group after quantization. Groups with more quantization-induced zeros are assigned fewer explicit pruning operations.

A standard 2:4 pruning mask is first generated using the popular wanda criterion (Sun et al., 2024). For each group of four weights, two weights with smaller activation-aware importance scores are pruned. The importance score is defined as

$$s_{ij} = |\mathbf{W}_{ij}| \cdot \|\mathbf{X}_{:,j}\|_2, \tag{6}$$

where $\mathbf{W}_{ij}$ denotes the weight and $\|\mathbf{X}_{:,j}\|_2$ denotes the corresponding activation norm.

Let $\mathbf{M}^{\text{trial}}$ denote the trial pruning mask, where $\mathbf{M}^{\text{trial}}_{ij} = 1$ indicates that the weight is retained and $\mathbf{M}^{\text{trial}}_{ij} = 0$ indicates that the weight is pruned. Under this mask, the remaining weights are further updated by the second-order compensation step accordingg to 3 to reduce the pruning reconstruction error. The compensated sparse weights are then quantized. In this work, we use GPTQ for quantization. The retained weights that are mapped to zero after quantization are identified as quantization-induced extra zeros:

$$\mathbf{E}^{\text{trial}} = \left(\mathbf{M}^{\text{trial}} = 1\right) \wedge \left(\mathbf{W}^{\text{trial}}_{q} = 0\right). \tag{7}$$

For each group of four weights, the quantization-induced sparsity is estimated from $\mathbf{E}^{\text{trial}}$. Let $\mathcal{G}_g$ denote the g-th group of four weights, and the number of quantization-induced zeros in this group is computed as:

$$z^{\text{trial}}_g = \sum_{(i,j)\in\mathcal{G}_g} \mathbf{E}^{\text{trial}}_{ij}. \tag{8}$$

Based on z^{trial}_g, the final pruning mask $\mathbf{M}^{\text{final}}$ is generated on the original dense weights. Since the target 2:4 pattern requires two zeros in each group, the number of explicitly pruned weights in group

$\mathcal{G}_g$ is set to

$$p_g = \max(2 - z_g^{\text{trial}}, 0). \tag{9}$$

Here, p_g denotes the number of weights explicitly pruned before quantization. When $z_g^{\text{trial}} = 0$, the group follows the standard 2:4 pruning pattern and two weights are pruned. When $z_g^{\text{trial}} = 1$, only one weight is pruned, because one zero is expected to be introduced by quantization. When $z_g^{\text{trial}} \geq 2$, no weight is pruned in the dense weights, since quantization is expected to provide sufficient sparsity for this group.

4.2 Quantization: Gain Free Sparsity

In this stage, the pruned weights are quantized. This process compresses the weight values, introducing zeros where the values become insignificant. The resulting quantized weights help further define which elements are zeroed out, guiding the selection of the final mask. However, some groups may still not meet the desired 2:4 sparsity because of quantization-induced zeros.

4.3 Second Pruning: Refine Sparsity

This component re-prunes the quantized weights to satisfy the target sparsity pattern, while compensating for the induced reconstruction error under the quantization-grid constraint. We treat the pruning in both the first and second stages as part of the sparsity, and for elements that do not meet the sparsity requirements, we perform additional pruning. Each time an element is pruned, another element in the quantization grid is updated to ensure that the sparsity constraint is met. The method for selecting these elements and updating them is as follows.

When an additional weight w_q is pruned, we compensate for its removal by adjusting one retained weight. The perturbation vector has two non-zero entries: $\delta w_q = -w_q$ and $\delta w_j = c$ for some compensation weight w_j ($j \neq q$). Let $\mathbf{H} \in \mathbb{R}^{C \times C}$ denote the Hessian of the layer reconstruction loss with respect to one weight row. The second-order loss change caused by pruning w_q and compensating through w_j is

$$\delta L(j, c) = A - \mathbf{H}_{qj} w_q c + \frac{1}{2}\mathbf{H}_{jj} c^2, \qquad A = \frac{1}{2}\mathbf{H}_{qq} w_q^2. \tag{10}$$

Here, A is the direct pruning cost, the second term represents the error reduction brought by compensation, and the last term is the cost of adjusting w_j.

For a fixed compensation target j, the continuous optimal shift is obtained by minimizing $\delta L(j, c)$ with respect to c:

$$c_j^{\text{cont}} = \frac{\mathbf{H}_{qj} w_q}{\mathbf{H}_{jj}}. \tag{11}$$

However, since the weights have already been quantized, the compensation shift must lie on the quantization grid. Therefore, we project the continuous shift to the nearest feasible grid point:

$$c_j^* = s_r \cdot \text{round}\left(\frac{c_j^{\text{cont}}}{s_r}\right), \qquad \varepsilon_j = c_j^* - c_j^{\text{cont}}, \tag{12}$$

where s_r denotes the quantization grid spacing.

The net compensation gain of selecting w_j is then defined as

$$\text{gain}(j) = \frac{(\mathbf{H}_{qj} w_q)^2}{2\mathbf{H}_{jj}} - \frac{1}{2}\mathbf{H}_{jj}\varepsilon_j^2. \tag{13}$$

The first term measures the continuous second-order compensation benefit, while the second term penalizes the mismatch caused by projecting the shift onto the quantization grid.

Finally, we select the compensation target with the largest positive gain:

$$j^* = \arg\max_{j \neq q} \text{gain}(j). \tag{14}$$

If $\text{gain}(j^*) > 0$, the compensation is applied. Otherwise, w_q is pruned without compensation.

Table 1: Changes in reconstruction error introduced by different components of our method using 128 WikiText2 calibration samples for the first Transformer block in LLaMA2-7B.

Layer Name	Quantization-aware Mask Selection			Grid-constrained Compensation		
	w/o	w/	Reduction	w/o	w/	Reduction
self_attn.q_proj	237284.40	228328.78	8955.62	15571.47	15571.43	0.04
self_attn.k_proj	155694.78	150021.42	5673.36	13062.86	13062.82	0.04
self_attn.v_proj	608.84	590.21	18.63	109.93	109.93	0.00
self_attn.o_proj	54.86	53.67	1.19	0.87	0.81	0.06
mlp.up_proj	17795.01	17275.85	519.16	4684.49	4653.05	31.44
mlp.gate_proj	19575.58	18971.49	604.09	4932.94	4900.74	32.20
mlp.down_proj	152.25	147.62	4.63	24.56	24.56	0.00

5 Experiments

5.1 Setups

Models and Datasets. We select the current public LLMs for evaluation, including the LLaMA2 series (Touvron et al., 2023b), LLaMA3 series (Grattafiori et al., 2024) and Mistral-7B (Jiang et al., 2023). These models range in size from 7 billion to 70 billion parameters. We assess the performance of pruned large language models through perplexity, using the WikiText2 (Merity et al., 2017) dataset. We also conduct experiments on six standard common-sense benchmark tasks: PIQA (Bisk et al., 2020), BoolQ (Clark et al., 2019), HellaSwag (Zellers et al., 2019), ARC-Easy (Clark et al., 2018), ARC-Challenge (Clark et al., 2018), and WinoGrande (Sakaguchi et al., 2020). All zero-shot tasks are uniformly performed based on the lm-eval-harness framework (Sutawika et al., 2024).

Comparison Methods. We compare our method with several counterpart methods, including: (1) SparseGPT (Frantar & Alistarh, 2023) combined with GPTQ (Frantar et al., 2023): SparseGPT removes weights with second-order compensation, and GPTQ quantizes the resulting sparse model. (2) OBR (Guo et al., 2026): a second-order restoration-based method that jointly handles quantization and sparsification to minimize reconstruction error.

Implementation Details. Our method is implemented with the PyTorch framework (Paszke et al., 2019) and HuggingFace Transformers (Wolf et al., 2020). Consistent with previous works (Guo et al., 2026), the calibration data contains 128 samples randomly selected from the first shard of the Wikitext2 dataset (Merity et al., 2017). Each sample contains sequences of 2048 tokens. Following the prior OBR (Guo et al., 2026), our method employs a Hadamard transform to smooth outliers using QuaRot (Ashkboos et al., 2024). For OBR and our method, we both adopt GPTQ for quantizer.

5.2 Layer-wise Analysis

We further analyze the reconstruction error on LLaMA2-7B, with results shown in Table 1. Without our proposed components, the model follows the standard prune-then-quantize pipeline using OBR. *Quantization-aware Mask Selection* consistently reduces reconstruction error across all layers by utilizing quantization-induced zeros, thereby requiring fewer weights to be explicitly pruned to satisfy the 2:4 sparsity constraint. *Grid-constrained Compensation* further reduces the error in several layers, particularly in the MLP projections, while providing smaller gains in others. This variation is mainly related to the amount of additional sparsity introduced during the second pruning stage. Overall, both components effectively reduce layer-wise reconstruction error..

5.3 Main Benchmark Results

2:4 Sparsity with Weight-only Quantization. Table 2 presents the experimental results of LLaMA2-7B under 2:4 sparsity with 4-bit weight-only quantization. Our proposed method consistently outperforms both SparseGPT+GPTQ and OBR across nearly all models and metrics. For example, on LLaMA2-13B, PQP achieves a WikiText perplexity of 7.84 and an average zero-shot accuracy of 60.47%, compared to 8.29 and 59.59% for OBR.

Table 2: WikiText2 performance (↓) and zero-shot accuracy (↑) for various LLMs under 2:4 sparsity with 4bit weight-only quantization.

Model	Method	Wiki (↓)	Zero-shot Accuracy (↑)						
			ARC-C	ARC-E	Boolq	HellaS.	PIQA	WinoG.	*Avg*
2-7B	FP16	5.47	46.25	74.49	77.68	76.00	79.05	69.06	70.42
	SparseGPT+GPTQ	21.08	23.38	41.04	62.08	35.85	58.98	54.70	46.01
	OBR	10.38	29.69	54.71	65.57	48.68	66.10	59.98	54.12
	PQP (ours)	**9.77**	**30.46**	**55.30**	**64.50**	**49.26**	**66.38**	**62.27**	**54.69**
2-13B	FP16	4.88	49.15	77.44	80.58	79.37	80.58	72.14	73.21
	SparseGPT+GPTQ	9.61	34.22	59.34	64.40	54.71	69.75	62.35	57.46
	OBR	8.29	34.30	62.58	70.43	56.53	70.84	62.83	59.59
	PQP (ours)	**7.84**	**34.39**	**62.46**	**72.23**	**58.17**	**71.87**	**63.69**	**60.47**
2-70B	FP16	3.32	57.42	81.06	83.73	83.84	82.70	77.98	77.79
	SparseGPT+GPTQ	5.59	46.93	74.71	**74.10**	71.22	76.55	71.98	69.25
	OBR	6.29	49.06	74.83	70.70	70.83	76.77	**74.03**	69.37
	PQP (ours)	**5.03**	**49.32**	**77.27**	73.36	**74.37**	**78.02**	73.95	**71.05**
3-8B	FP16	6.14	52.99	77.69	80.92	79.14	80.85	72.77	74.06
	SparseGPT+GPTQ	18.14	28.92	49.71	**65.11**	42.88	66.05	59.27	51.99
	OBR	14.42	29.52	51.26	63.00	47.45	**66.10**	59.59	52.82
	PQP (ours)	**14.34**	**30.38**	**51.18**	62.97	**48.12**	65.29	**60.46**	**53.07**
3-70B	FP16	2.86	64.25	86.15	85.23	84.85	84.55	80.43	80.91
	SparseGPT+GPTQ	23.59	24.06	41.75	61.35	37.21	63.06	56.75	47.36
	OBR	9.32	43.77	69.87	79.36	64.17	73.72	72.14	67.17
	PQP (ours)	**8.78**	**44.88**	**70.71**	**79.72**	**65.41**	**74.16**	**73.56**	**68.07**

Table 3: WikiText2 performance (↓) and zero-shot task accuracy (↑) for Mistral-7B under 2:4 sparsity with different weight-activation quantization settings.

W-A-KV	Method	Wiki (↓)	Zero-shot Accuracy (↑)						
			ARC-C	ARC-E	BoolQ	HellaS.	PIQA	WinoG.	*Avg*
16-16-16	-	5.25	54.01	79.46	83.70	81.09	82.10	74.03	75.73
8-8-16	SparseGPT+GPTQ	8.80	33.87	60.48	68.41	57.02	69.64	62.83	58.71
	OBR	8.13	**36.01**	64.27	68.17	58.13	**70.57**	63.85	60.17
	PQP (ours)	**8.11**	**36.01**	**64.44**	**69.11**	**58.47**	70.46	**63.93**	**60.40**
4-8-16	SparseGPT+GPTQ	9.65	31.40	59.64	67.80	54.66	68.39	62.59	57.41
	OBR	8.97	33.62	60.40	69.45	55.61	**68.99**	**62.75**	58.47
	PQP (ours)	**8.51**	**34.39**	**62.46**	**70.92**	**56.64**	68.88	62.51	**59.30**
4-8-8	SparseGPT+GPTQ	9.65	31.66	59.68	67.68	54.58	68.61	**63.61**	57.64
	OBR	8.97	33.70	60.31	69.33	55.72	68.50	62.90	58.41
	PQP (ours)	**8.51**	**33.96**	**62.54**	**70.73**	**56.70**	**68.66**	62.75	**59.22**
4-4-16	SparseGPT+GPTQ	11.66	28.33	50.84	64.83	47.32	65.29	56.27	52.15
	OBR	10.45	29.01	53.87	66.06	50.27	64.69	57.22	53.52
	PQP (ours)	**9.90**	**30.12**	**54.84**	**67.52**	**50.55**	**66.10**	**57.93**	**54.51**
4-4-4	SparseGPT+GPTQ	12.30	26.45	49.79	63.73	45.75	63.38	55.49	50.77
	OBR	10.91	**28.50**	51.94	65.99	49.12	**65.51**	57.14	53.03
	PQP (ours)	**10.32**	28.07	**52.44**	**66.91**	**49.69**	64.53	**59.27**	**53.48**

2:4 Sparsity with Weight-Activation Quantization. Table 8 presents the results of Mistral-7B under different weight-activation-KV cache quantization settings combined with 2:4 sparsity. PQP consistently achieves lower perplexity while maintaining competitive zero-shot accuracy across all configurations. Notably, under the W8A8KV16 setting, PQP achieves a perplexity of 8.11 and an average accuracy of 60.40%, outperforming OBR, demonstrating that PQP remains effective even when the quantization error is relatively small at 8-bit precision. Under the more aggressive W4A4KV4 setting, PQP further reduces perplexity from 8.97 to 8.51 and improves average accuracy from 58.47% to 59.30%. Similar advantages are also observed under W4A4KV16 and W4A4KV4, confirming the robustness of PQP across different quantization precisions.

Table 4: Ablation study of different components of our method on LLaMA2-13B under 2:4 Sparsity and W4A16KV16 quantization.

Component	Wiki (↓)	Avg. Acc. (↑)
Dense	4.88	73.21
OBR	8.29	59.59
Quantization-aware mask (ours)	8.05	59.94
Grid-constraint compensation (ours)	7.84	60.47

Table 5: Ablation study of different weight quantizers on LLaMA2-7B under 2:4 Sparsity and W4A4KV16 quantization.

Quantizer	Method	Wiki (↓)	Avg. Acc. (↑)
RTN	OBR	13.74	48.45
	PTP (ours)	12.90	49.41
GPTQ	OBR	12.97	48.60
	PTP (ours)	12.08	49.82

5.4 Ablation Study

Components of Our Method. In this ablation study, we investigate the impact of different components. Results on LLaMA2-13B are shown in Table 4. Introducing the quantization-aware mask improves performance, with Wiki dropping to 8.05 and average accuracy accurancy increasing to 59.94. This shows that leveraging quantization-induced sparsity allows fewer weights to be pruned, improving performance. However, this is done without compensation during the third pruning phase, in which weights are directly pruned. The final modification, grid-constraint compensation, provides the best performance overall, demonstrating that our grid-based compensation method helps improve results after pruning and quantization.

Weight Quantizers. As shown in Table 5, our method does not rely on a specific weight quantizer and can be readily combined with different quantization schemes. We evaluate our method with both RTN and GPTQ under 2:4 sparsity and W4A4KV16 quantization. With RTN, PTP reduces the WikiText perplexity from 13.74 to 12.90 while improving the average zero-shot accuracy from 48.45 to 49.41. Similar improvements are observed with GPTQ, where the perplexity decreases from 12.97 to 12.08 and the average accuracy increases from 48.60 to 49.82. These consistent gains demonstrate that our method is largely quantizer-agnostic and can effectively complement different weight quantization methods.

5.5 Practical Speedup Analysis

To validate the practical hardware efficiency of jointly applying low-bit quantization and semi-structured sparsity, we benchmark different inference configurations. As shown in Figure 3, the comparison of FP16 Dense, INT4 Dense, and INT4 2:4 Sparse across various batch sizes demonstrates the significant performance advantages of INT4 2:4 Sparse in latency. As batch size increases, INT4 2:4 Sparse shows consistent improvements, achieving a latency of 3.5 ms at batch size 256, a 5.7x speedup compared to FP16 Dense. As batch size increases, INT4 2:4 Sparse shows consistent improvements, achieving a latency of 3.5 ms at batch size 256, a 5.7x speedup compared to FP16 Dense.

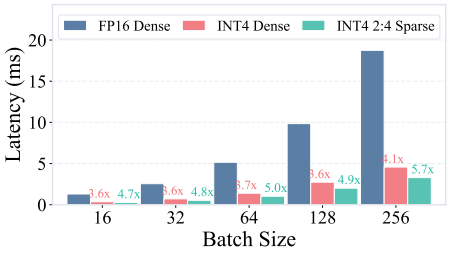

Figure 3: Speedup performance comparison on a single NVIDIA A100-SXM4-80GB GPU, with a sequence length of 512.

6 Conclusion

In this paper, we propose a pruning-quantization-pruning joint compression framework for 2:4 sparsity and quantization, which utilizes the free sparsity in mainstream low-bit symmetric quantization. Unlike traditional joint compression methods, our approach first performs pruning in the initial stage, without immediately meeting the sparsity requirement. We employ *Quantization-aware Mask Selection* to leave space for free sparsity. Then, low-bit symmetric quantization is applied to capture this free sparsity. In the third stage, we re-prune to refine the sparsity, ensuring it meets the required sparsity constraints. Additionally, we propose the *Grid-constrained Compensation* method to reduce the errors caused by second pruning. Experimental results on relevant LLMs show that our method outperforms the existing SOTA methods.

AI USE STATEMENT

Generative AI tools were used solely for language polishing and grammatical refinement. They were not involved in method design, experimental design, implementation, data analysis, or scientific conclusions. All methods, experiments, results, and claims were produced and verified by the authors, and the authors take full responsibility for the content of this paper.

REFERENCES

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated llms. In NeurIPS, 2024.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In AAAI, 2020.

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

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

DeepSeekAI, Aixin Liu, Bei Feng, Bin Wang, Bingxuan Wang, Bo Liu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Hao Yang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J.L. Cai, et al. Deepseek-v2: A strong, economical, and efficient mixture-of-experts language model. arXiv preprint arXiv:2405.04434, 2024.

Elias Frantar and Dan Alistarh. Sparsegpt: Massive language models can be accurately pruned in one shot. In ICML, 2023.

Elias Frantar, Sidak Pal Singh, and Dan Alistarh. Optimal brain compression: A framework for accurate post-training quantization and pruning. In NeurIPS, 2022.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Gptq: Accurate post-training quantization for generative pre-trained transformers. In ICLR, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad AlDahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, et al. The llama 3 herd of models. arXiv preprint arXiv:2407.21783, 2024.

Hang Guo, Luca Benini, and Yawei Li. Optimal brain restoration for joint quantization and sparsification of llms. In ICLR, 2026.

Jinyang Guo, Jianyu Wu, Zining Wang, Jiaheng Liu, Ge Yang, Yifu Ding, Ruihao Gong, Haotong Qin, and Xianglong Liu. Compressing large language models by joint sparsification and quantization. In ICML, 2024.

Simla Burcu Harma, Ayan Chakraborty, Elizaveta Kostenok, Danila Mishin, Dongho Ha, Babak Falsafi, Martin Jaggi, Ming Liu, Yunho Oh, Suvinay Subramanian, and Amir Yazdanbakhsh. Effective interplay between sparsity and quantization: From theory to practice. In ICLR, 2025.

B. Hassibi, D.G. Stork, and G.J. Wolff. Optimal brain surgeon and general network pruning. In NeurIPS, 1993.

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. Mistral 7b. arXiv preprint arXiv:2310.06825, 2023.

Junhan Kim, Ho-young Kim, Eulrang Cho, Chungman Lee, Joonyoung Kim, and Yongkweon Jeon. Boa: Attention-aware post-training quantization without backpropagation. In ICML, 2025.

Minjun Kim, Jaehyeon Choi, Hyunwoo Yang, Jongjin Kim, Jinho Song, and U Kang. Prune-then-quantize or quantize-then-prune? understanding the impact of compression order in joint model compression. In ICLR, 2026.

Yuhang Li, Ruokai Yin, Donghyun Lee, Shiting Xiao, and Priyadarshini Panda. Gptaq: Efficient finetuning-free quantization for asymmetric calibration. In ICML, 2025.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. In MLSys, 2024.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: Llm quantization with learned rotations. In ICLR, 2025.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In ICLR, 2017.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In NeurIPS, 2019.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In AAAI, 2020.

Mingjie Sun, Zhuang Liu, Anna Bair, and J. Zico Kolter. A simple and effective pruning approach for large language models. In ICLR, 2024.

Yuxuan Sun, Ruikang Liu, Haoli Bai, Han Bao, Kang Zhao, Yuening Li, Jiaxin Hu, Xianzhi Yu, Lu Hou, Chun Yuan, et al. Flatquant: Flatness matters for llm quantization. In ICML, 2025.

L Sutawika, L Gao, H Schoelkopf, S Biderman, J Tow, B Abbasi, B Fattori, C Lovering, J Phang, A Thite, et al. Eleutherai/lm-evaluation-harness: Major refactor, 2024.

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

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

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. Quip#: Even better llm quantization with hadamard incoherence and lattice codebooks. In ICML, 2024a.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. Quip#: Even better llm quantization with hadamard incoherence and lattice codebooks. In ICML, 2024b.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In EMNLP, 2020.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In ICML, 2023.

Lu Yin, You Wu, Zhenyu Zhang, Cheng-Yu Hsieh, Yaqing Wang, Yiling Jia, Mykola Pechenizkiy, Yi Liang, Zhangyang Wang, and Shiwei Liu. Outlier weighed layerwise sparsity (owl): A missing secret sauce for pruning llms to high sparsity. In ICML, 2024.

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

Aojun Zhou, Yukun Ma, Junnan Zhu, Jianbo Liu, Zhijie Zhang, Kun Yuan, Wenxiu Sun, and Hongsheng Li. Learning n:m fine-grained structured sparse neural networks from scratch. In ICLR, 2021.

A ALGORITHM

Algorithm 1 PQP

Input: Weight matrix $\mathbf{W} \in \mathbb{R}^{R \times C}$, calibration activations $\mathbf{X}$, sparsity pattern $N : M$, bit-width b
Output: Sparse and quantized weight matrix $\widehat{\mathbf{W}}$
$\mathbf{H} \leftarrow \mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I}$ ▷ Hessian approximation
$S_{ij} \leftarrow |W_{ij}| \cdot \|\mathbf{X}_{:,j}\|_2$ ▷ Activation-aware importance score
Stage I: Pruning – Quantization-Aware Mask
$\mathbf{M}_0 \leftarrow \mathrm{Top}_{M-N}(\mathbf{S})$ ▷ Initial $N : M$ pruning mask
$\widetilde{\mathbf{W}} \leftarrow Q_b(\mathbf{W} \odot \mathbf{M}_0)$
$\mathbf{Z}_q \leftarrow \mathbb{I}(\mathbf{M}_0 = 1) \odot \mathbb{I}(\widetilde{\mathbf{W}} = 0)$ ▷ Quantization-induced zeros
for each $N : M$ group g **do**
    $z_g \leftarrow \sum_{j \in g} Z_{q,j}$
    $n_g \leftarrow \max(0, N - z_g)$
    $\mathcal{P}_g \leftarrow \mathrm{Bottom}_{n_g}\{S_j \,|\, j \in g\}$ ▷ Select least-important weights
    $M^{(g)}_{q,j} \leftarrow \mathbb{I}(j \notin \mathcal{P}_g)$
end for
Stage II: Quantization
$\bar{\mathbf{W}}_\mathbf{q} \leftarrow Q_b(\mathbf{W} \odot \mathbf{M}_0)$ ▷ Apply true quantization
Stage III: Pruning – Grid-Constrained Compensation
for each $N : M$ group g **do**
    $r_g \leftarrow N - \sum_{j \in g} \mathbb{I}(\odot W_{q,j} = 0)$ ▷ Remaining pruning requirement
    if $r_g > 0$ **then**
        $\mathcal{P}_g \leftarrow \mathrm{Bottom}_{r_g}\{|\widehat{W}_{q,j}| \|\mathbf{X}_{:,j}\|_2\}_{j \in g}$ ▷ Select additional pruning weights
    end if
end for
$\mathcal{P} \leftarrow \bigcup_g \mathcal{P}_g$
for each $q \in \mathcal{P}$ **do**
    for each available compensation position j **do**
        $c_{qj} \leftarrow \dfrac{\mathbf{H}_{qj}}{\mathbf{H}_{jj}} \widehat{w}_q$ ▷ Continuous optimal compensation
        $c^*_{qj} \leftarrow \Delta_j \,\mathrm{round}\left(\dfrac{c_{qj}}{\Delta_j}\right)$ ▷ Quantization-grid projection
        $\epsilon_{qj} \leftarrow c^*_{qj} - c_{qj}$
        $G_{qj} \leftarrow \dfrac{(\mathbf{H}_{qj}\widehat{w}_q)^2}{2\mathbf{H}_{jj}} - \dfrac{\mathbf{H}_{jj}}{2}\epsilon^2_{qj}$ ▷ Net compensation gain
    end for
    $j^* \leftarrow \arg\max_j G_{qj}$ ▷ Best compensation position
    $\widehat{w}_q \leftarrow 0$ ▷ Prune selected weight
    if $G_{qj^*} > 0$ **then**
        $\widehat{w}_{j^*} \leftarrow \widehat{w}_{j^*} + c^*_{qj^*}$ ▷ Grid-constrained weight compensation
    end if
end for
return $\widehat{\mathbf{W}}$

B Layer-wise Analysis

B.1 Prediction Accuracy

Since the first stage of our method involves predicting quantization-induced zeros, the prediction accuracy is a prerequisite for the effectiveness of our method. To evaluate this, we calculate the prediction accuracy for all linear layers in the first Transformer block of LLaMA2-7B. Table 6 reports the average prediction accuracy between the zero positions predicted during trial quantization and those produced by actual quantization. The prediction accuracy increases from 44.12% at 8-bit to over 72% under 4-bit, 3-bit, and 2-bit quantization. This shows that the trial quantization can reliably identify quantization-induced zero positions at low bit-widths, providing useful guidance for our quantization-aware mask construction. Such consistency provides reliable guidance for constructing the quantization-aware sparsity mask and enables our method to effectively exploit quantization-induced zeros in advance.

Table 6: Prediction accuracy of quantization-induced zero positions under different bit-widths.

Bits	8	4	3	2
Prediction Accuracy (%)	44.12	72.17	72.82	74.70

B.2 Sparsity Source

To investigate the sources of sparsity under different quantization bit-widths, we further analyze how the final sparsity is contributed by different stages of the pruning and quantization process. Specifically, we aggregate the sparsity statistics across different linear layers weighted by their numbers of parameters. For the direct baseline, the pruning mask is fixed at 50% sparsity As shown in Figure 4, for the direct baseline, quantization-induced sparsity increases from 0.21% at 8-bit to 12.64% at 2-bit. Our method explicitly accounts for this quantization-induced sparsity and therefore reduces the sparsity introduced by the initial mask. Consequently, only a small amount of additional sparsity is required in Stage 3, ranging from 0.11% to 0.80%. These results show that our method effectively reallocates the sources of sparsity across different stages and better exploits the sparsity naturally introduced by low-bit quantization.

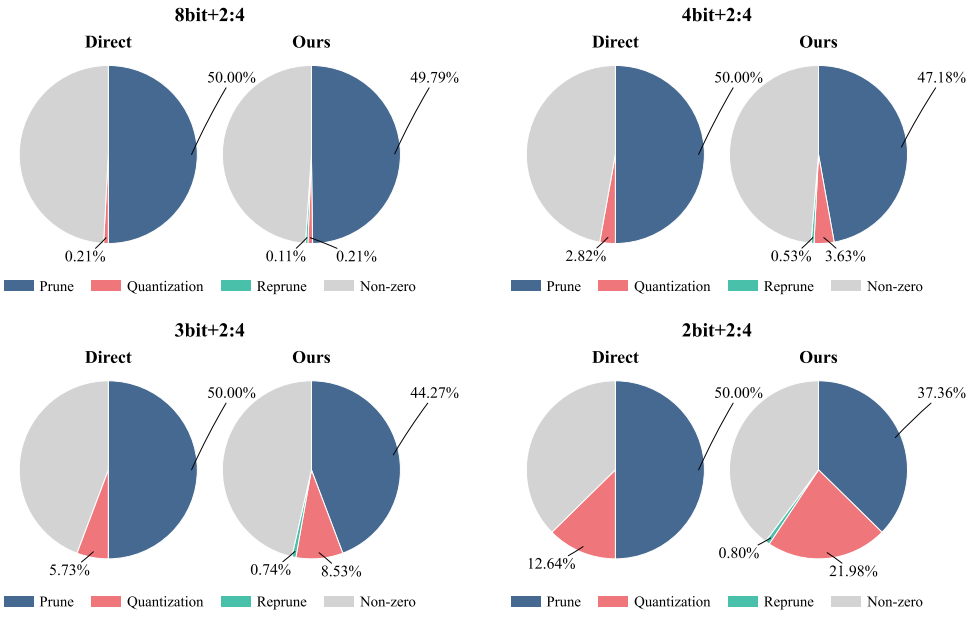

Figure 4: Sparsity sources under different quantization bit-widths.

Table 7 reports the utilization rate of quantization-induced zeros under different bit-widths, defined as the proportion of quantization-induced zeros that can be directly incorporated into the target 2:4 sparsity pattern. The utilization rate reaches 45.52% at 8-bit and increases to 63.09% at 4-bit, indicating that a substantial portion of the zeros naturally introduced by quantization can be effectively reused to satisfy the structured sparsity constraint. Although more zeros are generated as the bit-width decreases, not all of them can be utilized due to the structural constraint of the 2:4 pattern. Overall, these results demonstrate that quantization-induced sparsity provides a considerable source of reusable zeros for semi-structured pruning, which motivates explicitly incorporating such sparsity into the mask construction process.

Table 7: Utilization rate of quantization-induced zeros under different bit-widths.

Bits	8	4	3	2
Utilization Rate (%)	45.52	63.09	58.59	53.88

C More Experimental Results

Ablation of Calibration Data. As shown in Figure 5, we investigate the sensitivity of our method to the number of calibration samples. On WikiText2, our method consistently achieves lower perplexity than the official baseline across all calibration sizes, while the overall performance gradually improves as more calibration samples are used. Notably, the advantage becomes more evident with larger calibration sets, indicating that our method can effectively exploit additional calibration statistics. On C4, the average zero-shot accuracy remains relatively stable across different numbers of calibration samples, with our method achieving comparable or better performance than the baseline in most settings. These results suggest that our method is not highly sensitive to the calibration-set size and can already obtain competitive performance with a relatively small number of calibration samples.

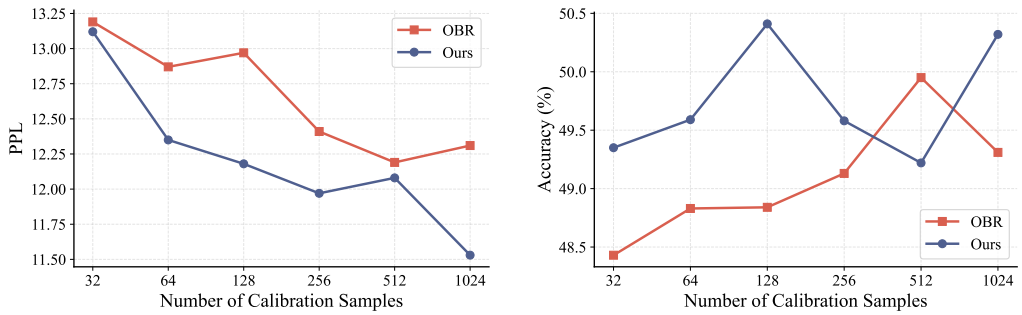

Figure 5: Ablation study of calibration samples on LLaMA2-7B by joint 2:4 sparsity and W4A4 quantization. (a) WikiText2 perplexity (PPL) (↓) under different numbers of calibration samples drawn from WikiText2. (b) Average zero-shot accuracy (↑) under different numbers of calibration samples drawn from C4.

Different Sparsity Patterns. We further evaluate the performance of our method under different sparsity patterns. As shown in Table 8, PQP consistently improves WikiText2 perplexity across different sparsity types, reducing the PPL from 12.97 to 12.08, 10.03 to 9.56, and 9.04 to 8.77 compared with OBR under 2:4, 4:8, and 8:16 sparsity, respectively. For 2:4 sparsity, PQP also achieves the best average zero-shot accuracy of 49.82 and obtains the best results on five of the six individual downstream tasks. Similar improvements are observed under 4:8 sparsity, where PQP achieves the highest average accuracy of 53.61 while providing the best performance on ARC-E, HellaSwag, and PIQA. Under the less restrictive 8:16 sparsity, PQP maintains the lowest perplexity and achieves competitive zero-shot accuracy, with the best results on ARC-C, HellaSwag, and PIQA. Overall, these results demonstrate the effectiveness and robustness of PQP for joint sparsity and quantization across different structured sparsity patterns, consistently improving language modeling performance while maintaining competitive downstream accuracy.

Table 8: WikiText2 performance (↓) and zero-shot task accuracy (↑) for LLaMA2-7B under W4A4KV16 quantization with different sparsity types by QuaRot.

W-A-KV	Method	Wiki (↓)	Zero-shot Accuracy (↑)						
			ARC-C	ARC-E	BoolQ	HellaS.	PIQA	WinoG.	*Avg*
Dense	-	5.47	46.25	74.49	77.68	76.00	79.05	69.06	70.42
2:4	SparseGPT+GPTQ	21.08	23.38	41.04	62.08	35.85	58.98	54.70	46.01
	OBR	12.97	26.71	45.79	62.23	41.19	60.45	**55.25**	48.60
	PQP (ours)	**12.08**	**27.39**	**48.99**	**63.18**	**43.12**	**61.37**	54.85	**49.82**
4:8	SparseGPT+GPTQ	11.71	27.56	51.94	63.73	48.75	64.53	**61.17**	52.95
	OBR	10.03	**30.55**	51.64	**65.50**	50.64	63.93	57.46	53.29
	PQP (ours)	**9.56**	29.61	**52.86**	63.46	**52.45**	**65.56**	57.70	**53.61**
8:16	SparseGPT+GPTQ	9.29	31.48	56.40	**65.93**	54.56	68.28	**62.98**	**56.60**
	OBR	9.04	32.17	**57.20**	64.83	55.48	67.36	58.96	56.00
	PQP (ours)	**8.77**	**32.94**	56.27	65.87	**55.82**	**69.53**	58.88	56.55

Different Weight Bits. We evaluate PQP under different weight bit-widths and the results are shown in Table 9. At 8-bit weight quantization, PQP achieves comparable performance to OBR, with only marginal differences in WikiText perplexity and zero-shot accuracy. As the weight precision is further reduced to 4-bit and 3-bit, the advantage of PQP becomes more evident. Compared with OBR, PQP consistently improves both language modeling performance and downstream task accuracy. These results demonstrate that the proposed quantization-aware pruning and grid-constrained compensation become increasingly effective under more aggressive quantization settings. Interestingly, under extreme 2-bit quantization, PQP still maintains a clear advantage over OBR, achieving substantially lower WikiText perplexity and higher zero-shot accuracy. However, SparseGPT+GPTQ obtains better results than OBR in this setting. We hypothesize that this phenomenon is related to the joint optimization strategy adopted by SparseGPT+GPTQ, which performs pruning and quantization simultaneously. At extremely low precision, 2-bit quantization introduces additional quantization-induced sparsity, and the iterative optimization process may better exploit this interaction between sparsity and quantization. We leave the exploration of jointly optimizing pruning and quantization at ultra-low bit-widths as an interesting direction for future work.

Table 9: WikiText2 performance (↓) and zero-shot task accuracy (↑) performance for LLaMA2-7B under different weight bits with 2:4 sparsity by QuaRot.

Weight Bit	Method	Wiki (↓)	Zero-shot Accuracy (↑)						
			ARC-C	ARC-E	Boolq	HellaS.	PIQA	WinoG.	*Avg*
16 bit	Dense	5.47	46.25	74.49	77.68	76.00	79.05	69.06	70.42
8 bit	SparseGPT+GPTQ	17.19	23.12	41.58	62.23	37.02	59.41	55.33	46.45
	OBR	9.00	32.34	58.29	**69.66**	53.44	**69.86**	**62.35**	**57.66**
	PQP (ours)	**8.99**	**32.68**	**58.46**	68.50	**53.66**	69.64	62.27	57.54
4 bit	SparseGPT+GPTQ	21.08	23.38	41.04	62.08	35.85	58.98	54.70	46.01
	OBR	10.38	29.69	54.71	65.57	48.68	66.10	59.98	54.12
	PQP (ours)	**9.77**	**30.46**	**55.30**	**64.50**	**49.26**	**66.38**	**62.27**	**54.69**
3 bit	SparseGPT+GPTQ	38.09	21.08	34.05	57.83	30.49	54.30	51.07	41.47
	OBR	19.31	22.95	38.93	**61.53**	34.21	56.31	52.57	44.42
	PQP (ours)	**17.13**	**23.81**	**41.04**	61.38	**34.81**	**58.11**	**54.46**	**45.60**
2 bit	SparseGPT+GPTQ	189.31	**24.49**	27.19	**41.87**	27.57	51.20	48.70	**36.84**
	OBR	467.45	26.28	26.60	37.68	27.03	50.65	48.93	36.20
	PQP (ours)	**150.22**	23.04	**27.27**	37.83	**28.13**	**52.83**	**50.67**	36.63

Different Rotations. Table 10 evaluates the impact of different rotation strategies on the joint 2:4 sparsity and quantization framework. Across all three settings, our PQP consistently achieves the best WikiText2 perplexity and zero-shot accuracy, demonstrating the effectiveness of the proposed quantization-aware pruning and grid-constrained compensation. However, the benefit of applying rotations is not always consistent for sparse quantized models. Compared with the setting without

rotation, QuaRot provides clear improvements by redistributing activation outliers and reducing quantization errors, while SpinQuant only brings marginal changes in this scenario. Although rotations can alleviate quantization difficulty by dispersing outlier channels, they may simultaneously spread the importance of original weights across more channels, making the importance-based pruning process more challenging. Therefore, rotation strategies introduce a trade-off between improving quantization robustness and preserving accurate weight importance estimation for structured sparsity.

Table 10: WikiText2 performance (↓) and zero-shot task accuracy(↑) performance for LLaMA2-13B under W4A16KV16 quantization with different rotations.

Rotation	Method	Wiki (↓)	Zero-shot Accuracy(↑)						
			ARC-C	ARC-E	Boolq	HellaS.	PIQA	WinoG.	*Avg*
-	Dense	4.88	49.15	77.44	80.58	79.37	80.58	72.14	73.21
w/o	SparseGPT+GPTQ	9.61	34.04	59.30	64.40	54.73	69.86	62.51	57.47
	OBR	9.31	33.19	58.71	65.66	52.84	70.18	61.72	57.05
	PQP (ours)	**8.45**	**35.84**	**61.28**	**67.40**	**57.45**	**71.38**	**64.09**	**59.57**
QuaRot	SparseGPT+GPTQ	9.61	34.22	59.34	64.40	54.71	69.75	62.35	57.46
	OBR	8.29	34.30	62.58	70.43	56.53	70.84	62.83	59.59
	PQP (ours)	**7.84**	**34.39**	**62.46**	**72.23**	**58.17**	**71.87**	**63.69**	**60.47**
SpinQuant	SparseGPT+GPTQ	11.07	27.05	51.18	63.01	48.12	64.42	57.85	51.94
	OBR	8.47	33.19	61.62	68.29	55.60	69.80	62.43	58.48
	PQP (ours)	**7.94**	**33.62**	**61.91**	**71.22**	**58.01**	**71.00**	**63.14**	**59.92**

D Running Consumption Analyses

To evaluate the computational efficiency of our framework, we further compare the calibration runtime of PQP with the baseline method. As shown in Table 11, our PQP method introduces approximately $2\times$ runtime overhead compared with OBR on both LLaMA2-7B and LLaMA2-13B. This additional cost mainly comes from the quantization-aware mask selection in the first stage, which requires trial compensation before quantization to determine a more accurate mask. Despite this extra process, the overall calibration cost remains within a practical range, indicating that the computational overhead of our method is acceptable for post-training deployment.

Table 11: Runtime comparisonon LLaMA2-7B and LLaMA2-13B.

Method	LLaMA2-7B	LLaMA2-13B
OBR	1h18min	2h43min
PQP (ours)	2h35min	5h47min

E Future Work

In future work, we plan to explore a globally iterative framework for joint sparsification and quantization, where pruning and quantization are optimized in a more coordinated manner across layers. In particular, we aim to better exploit the additional sparsity naturally induced by low-bit quantization, rather than treating it merely as a by-product of quantization. We will also investigate more efficient and accurate compensation strategies to reduce reconstruction errors while substantially lowering the computational overhead of the compensation process, thereby improving the overall efficiency and scalability of joint sparsity-quantization methods.