TAMMP: An Effective Mixed-Precision Quantization Method for Enhancing Model Generalization in Low-Bit Scenarios

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have achieved remarkable progress across all domains. However, the substantial demands for memory and computational resources during deployment and inference hinder their adoption especially on edge devices. While quantization is a widely acknowledged compression technique successfully applied in various scenarios, models quantized to low bit-widths suffer from significant performance degradation compared to the original one. Furthermore, the magnitude of this degradation varies considerably across different tasks. To investigate the underlying causes, this paper formally introduces the critical parameter heterogeneity and hypothesize that this heterogeneity is a primary factor driving the non-uniform performance degradation observed across diverse downstream tasks. Addressing this hypothesis, a Task-Adaptive Multi-Granularity Mixed Precision Training (TAMMP) method is proposed. This approach incorporates critical parameter probing based on multi-source knowledge, multi-granular bit-width allocation, and a mixed-precision training framework. Finally, through comprehensive evaluations, we demonstrate that TAMMP achieves superior generalization performance compared to existing low-bit quantization approaches.

1 Introduction

LLMs (Touvron et al., 2023; Bubeck et al., 2023; Ying et al., 2024) have shown incredibleble performance in various complex language tasks. However, their large number of parameters results in high memory requirements (Kim et al., 2023b), which restrict their deployment in hardware resource-limited devices. To solve these problems, quantization methods (Dettmers et al., 2022) have been developed through reducing the bit width of the tensors, and effectively reduce the memory and computational requirements of LLMs. The research on model quantization mainly focuses on

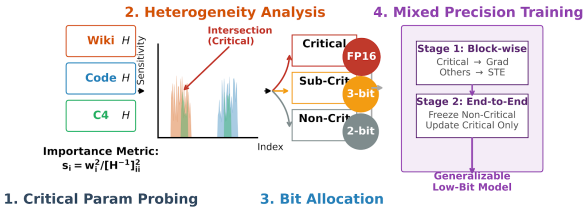

Figure 1: Schematic of the TAMMP framework for LLMs.

two areas (Gong et al., 2025): quantization aware training (QAT) (Liu et al., 2023; Xu et al., 2024; Chen et al., 2024a) and post-training quantization (PTQ) (Frantar et al., 2022; Shao et al., 2023). For quantization bit-widths of 4-bit and above, existing QAT and PTQ methods have achieved satisfactory progress (Kim et al., 2023a; Lin et al., 2024). Consequently, current research endeavors are striving for higher compression ratios, extending from low-bit quantization to the extreme compression found in Binary Neural Networks (Wang et al., 2023; Ma et al., 2024b,a; Xu et al., 2024). However, these efforts often overlook the impact of low-bit quantization on the model's generalization ability. For example, consider 3-bit GPTQ (Frantar et al., 2022): while it preserves 90% of the Llama2 7B model's capability on commonsense reasoning, its performance declines to 85% for mathematical reasoning. The disparity becomes even more serious on the tasks such as world knowledge and code generation, where GPTQ retains only 50% of the full-precision performance. Detailed experimental results are provided in the **Appendix**.

To investigate the varying degrees of performance degradation in low-bit quantized models across different downstream tasks, we assume that LLMs must activate specific parameter regions storing relevant knowledge to handle distinct tasks. Accordingly, this paper adopts a critical parameter perception approach based on multi-

source knowledge (i.e., C4, Wikipedia, GitHub, StackExchange, ArXiv) to probe the variations in knowledge-critical parameter regions during the quantization process. As a result, a phenomenon of critical parameter heterogeneity is observed: significant differences exist among the critical parameters corresponding to different knowledge sources. Quantization inflicts varying degrees of disruption on these parameters. In other words, different types of knowledge suffer from unequal levels of impairment. Based on this observation, this paper hypothesize that critical parameter heterogeneity is one of the primary causes for the decline in the generalization capability of quantized models. To address this challenge, we propose a Task-Adaptive Multi-Granularity Mixed Precision Training (TAMMP) framework, as illustrated in Figure 1. Guided by the distribution of critical parameters identified via multi-source knowledge probing, our method employs a multi-granular bit allocation strategy to mitigate the issue of critical parameter heterogeneity. Furthermore, we utilize a two-stage mixed precision training scheme to efficiently update multi-granular quantized parameters. Our main contributions are summarized as follows:

- We identify the phenomenon of critical parameter heterogeneity in model quantization and provide an in-depth analysis of the causes behind the generalization degradation of low-bit quantized models.
- To address the generalization challenges in quantized models arising from parameter heterogeneity, the proposed TAMMP method innovatively employs multi-source knowledge probing for critical parametersm, a multi-granular bit-width allocation and a mixed-precision training framework.
- Extensive experiments demonstrate that under low-bit quantization regimes, our method achieves superior accuracy and stronger generalization capabilities compared to existing state-of-the-art approaches.

2 Related Work

2.1 Outliers in Language Model Quantization

LLMs suffer from outlier issues in both parameters and activations. Despite their scarcity, these outliers significantly impact model outputs and introduce severe quantization difficulties. Identifying these outliers and minimizing their quantization error is thus essential for achieving high-performance

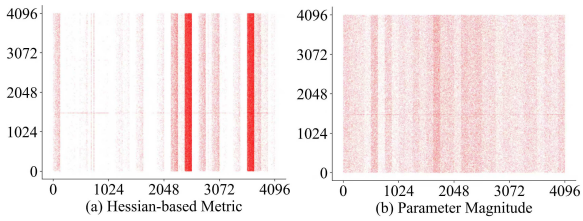

Figure 2: Scatter plot of the top 1% parameters identified by the Hessian-based metric and parameter magnitude, respectively. Each point in the scatter plots represents a parameter within the self_attn.o_proj tensor of the 25th transformer block in the Llama2 7B model. The color intensity corresponds to the density of critical parameters.

quantized models (Dettmers et al., 2023). CherryQ (Cui and Wang, 2024) select the heterogeneity-based outliers of parameter as the important parameter, and optimize high-precision cherry parameters and low-precision normal parameters by end-to-end mixed-precision quantization. OWQ (Lee et al., 2024) propose a sensitivity-aware mixed-precision scheme to identify the outliers (weak columns) by Hessian metric. Inspired by the aforementioned works, we employ a Hessian-based metric to evaluate parameter importance and utilize it to analyze the heterogeneity of critical parameters.

2.2 Impact of Calibration Data

While existing literature has begun to recognize the critical role of calibration data in quantization, efforts have been largely confined to calibration dataset filtering (Williams and Aletras, 2024). There is a notable lack of investigation into the underlying causes of generalization drops at the model parameter level. A representative work is COLA (He et al., 2025), which examines how data composition and domains affect LLM tasks and introduces a three-stage calibration data curation framework to curate data for improved generalization.

2.3 Mixed-precision Methods

Given the significant impact of outliers on model output, researchers have shown substantial interest in mixed-precision quantization strategies, which differentiate between outlier and non-outlier weights or activations, categorizing parameters as critical or non-critical, and then quantizing them to different bit-widths (Dettmers et al., 2022; Kim et al., 2023b). To explore even higher compression rates, PB-LLM (Shang et al., 2023) pioneered the

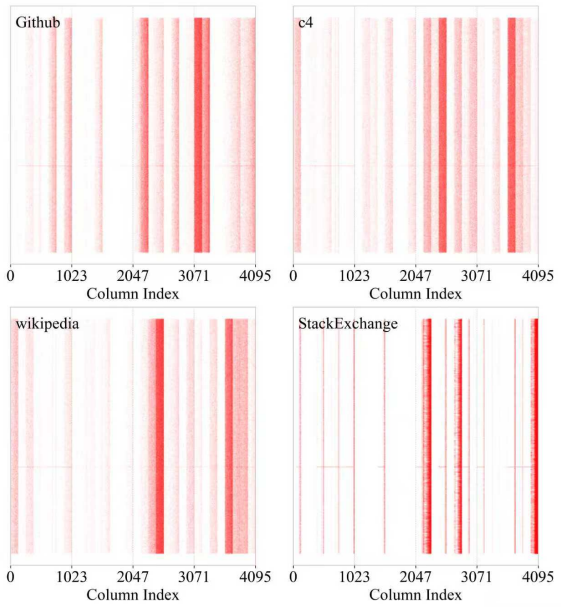

Figure 3: Distribution of critical parameters across multi-source knowledge domains (GitHub, C4, Wikipedia, and StackExchange) within the self_attn.o_proj tensor of the 25th transformer block in the Llama2 7B model.

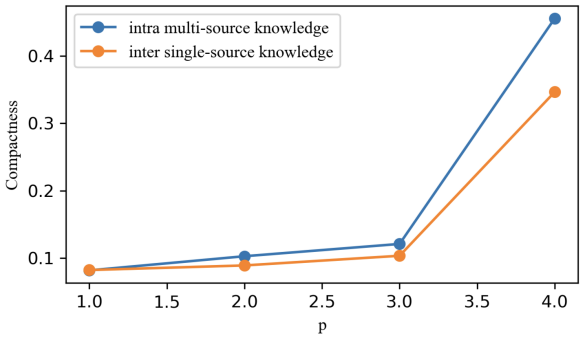

Figure 4: Disparity in critical parameter heterogeneity between multi-source and single-source knowledge. The p represents the threshold for the number of dataset occurrences required to define a parameter as a shared critical parameter.

binarization of non-significant weights in LLMs, allocating high precision to 10-30% of significant weights. More recently, GEAR (Kang et al., 2024) extended the mixed-precision concept to KV cache compression, leveraging low-rank matrix approximation to quantize residuals. CMPQ (Chen et al., 2024b) introduce a novel mixed-precision quantization method that allocates quantization precision in a channel-wise pattern based on activation distributions.

Existing mixed-precision training methods typically identify critical parameters using a fixed-source calibration dataset, restricting parameter representation and updates to a dual-granularity scheme (i.e., full-precision and low-precision). In contrast, to address the issue of important parameter heterogeneity, this paper adopts a multi-granular strategy for bit allocation and parameter updates.

3 Critical Parameter Heterogeneity

Full-precision LLMs rely on activating specific knowledge-bearing parameters to handle downstream tasks, implying that parameter importance is inherently task-dependent. To validate this behavior within the context of quantization, we employ calibration datasets from varying knowledge domains to probe the distribution of corresponding critical parameters. This allows us to explicitly analyze the phenomenon of critical parameter heterogeneity.

Instead of relying on weight magnitude, we utilize a Hessian-based metric (The specific methodology is detailed in the following section.) to probe critical parameters. As illustrated in the comparison between the left and right plots of Figure 2, the Hessian-based metric demonstrates a stronger capability to distinguish critical parameters from non-critical ones. Furthermore, another phenomenon observed is that critical parameters tend to cluster within specific columns. Consequently, our subsequent probing for critical parameters adopts a column-wise search strategy.

To validate critical parameter heterogeneity across multi-source knowledge, we selected the calibration data sources originates from pre-training data, comprising five subsets: GitHub, C4, Wikipedia, StackExchange and ArXiv. As illustrated in Figure 3, the distributions of critical parameters probed by different knowledge sources exhibit observable variations. The distribution of critical parameters probed from GitHub exhibits a significant divergence compared to that of the other knowledge sources.

To contrast the differences in critical parameter heterogeneity between multi-source and single-source knowledge, we perform repeated sampling within each of the five data sources, constructing five calibration datasets per source to probe their corresponding critical parameters. To quantify the differences in critical parameter heterogeneity between multi-source and single-source knowledge, we employ a compactness metric based on Jaccard similarity. For details regarding the calculation of the compactness metric, please refer

to the Appendix. Figure 4 illustrates the variations in critical parameter heterogeneity at both the inter-source level (among multiple sources) and the intra-source level (within a single source). The results reveal that the critical parameters compactness for multi-source knowledge is approximately 35%, whereas the compactness within the single-knowledge source reaches 50%. These results demonstrate that although there is some heterogeneity within single-source knowledge, the critical parameter heterogeneity across multi-source knowledge is non-negligible and crucial.

4 Methods

4.1 Critical Parameter Probing

The quantization often involves layer-wise quantization, where weights of the model are quantized sequentially from the input to the output layer. When quantizing a given layer, the objective is to find a quantized weight matrix $\hat{W}$ that minimizes the difference between the output activations of the original full-precision model and the quantized model. Given an input feature matrix $X \in R^{C_{in} \times N}$ (C_{in} is the number of input channels and N is the sequence length) and a full-precision weight matrix $W \in R^{C_{out} \times C_{in}}$ (C_{out} is the number of output features), the objective function to find $\hat{W}$ is typically defined as minimizing the squared error (Lee et al., 2024):

$$\arg\min_{\hat{W}} E = \arg\min_{\hat{W}} ||WX - \hat{W}X||_F^2 \quad (1)$$

The quantization error for each output channel i can be approximated using a Taylor expansion, leading to

$$E_i \approx \Delta W_{i,:} H \Delta W_{i,:}^T \quad (2)$$

where $\Delta W_{i,:} = W_{i,:} - \hat{W}_{i,:}$ is the weight perturbation for the i-th output channel. The Hessian matrix with respect to the layer-wise quantization error, denoted as $H \in R^{C_{in} \times C_{in}}$, would be calculate as:

$$H = 2XX^T \quad (3)$$

Inspired by the concept of weight sensitivity, we adopt a parameter importance metric by leveraging the Hessian matrix to identify critical parameters within the parameter matrices (Huang et al., 2024):

$$s_i = \frac{w_i^2}{[H^{-1}]_{ii}^2} \quad (4)$$

where H represents the Hessian matrix for each layer. As shown in Equation 3, the Hessian matrix for each layer can be computed from the input feature matrix X, and w_i represents the weight of each parameter.

Upon calculating the parameter importance metrics for each component of the LLMs, and recognizing that critical parameters tend to cluster within specific columns, this paper employs a Structural Searching Selection method to probe the critical columns within the model components. The detailed procedure of the Structural Searching Selection method is provided in the Appendix.

4.2 Multi-Granular Bit Allocation via Parameter Heterogeneity Analysis

Employing the multi-source knowledge probing method entails utilizing data from diverse sources to derive multiple sets of critical parameters. As revealed by our analysis of critical parameter heterogeneity, the compactness among these sets is merely around 50%. Failure to adequately preserve the information encoded in these critical parameters would degrade the quantized model's performance on relevant tasks. However, assigning high-precision bit-widths to the union of all detected critical parameters during the mixed-precision phase would cause the average bit-width to exceed the target low-bit budget.

To mitigate the impact of parameter heterogeneity while compressing the model, we adopt two straightforward yet efficient multi-granular parameter grouping strategies:

- Strategy 1: Specifically, the intersection of the critical parameter sets identified via multi-source probing is designated as the new critical parameters, and the non-overlapping portions are defined as sub-critical parameters. Finally, the remaining parameters are categorized as non-critical parameters.
- Strategy 2: The partitioning method for critical parameters remains unchanged. However, the union of all sets of non-critical columns is collectively designated as the actual non-critical columns. The remaining parameters are then classified as sub-critical columns.

Under Strategy 1, non-critical parameters dominate the distribution. Strategy 2 distinguishes itself by adding an explicit probing step for non-critical parameters. Consequently, this results in a notable expansion of the sub-critical parameter subset. Guided by QAT findings that 3-bit quantization retains generalization while 2-bit is adequate for basic capabilities, we allocate Float16, 3-bit, and

		Commonsense Reasoning								
Method	**Bits**	**PIQA**	**SocialIQA**	**HellaSwag**	**WinoGrande**	**ARC-c**	**ARC-e**	**OpenBookQA**	**CommonsenseQA**	**Avg.**
Llama2 7B										
-	FP16	0.781	0.460	0.571	0.691	0.435	0.763	0.314	0.575	0.574
GPTQ	2-bit	0.540	0.345	0.266	0.524	0.211	0.278	0.134	0.2	0.312
OmniQuant	2-bit	0.599	0.371	0.354	0.506	0.208	0.405	0.166	0.223	0.354
QUIP#	2-bi t	0.755	0.432	0.522	0.658	0.378	0.719	0.290	0.405	0.520
EfficientQAT	2-bit	0.744	0.431	0.497	0.653	0.352	0.71	0.258	0.256	0.488
TAMMP	2.367-bit	0.758	0.440	0.528	0.658	0.379	0.725	0.296	0.396	**0.522**
Llama2 13B										
-	FP16	0.792	0.473	0.600	0.721	0.484	0.794	0.35	0.683	0.612
GPTQ	2-bit	0.602	0.352	0.332	0.51	0.204	0.381	0.134	0.189	0.338
OmniQuant	2-bit	0.671	0.403	0.434	0.565	0.278	0.583	0.208	0.188	0.416
QUIP#	2-bit	0.770	0.455	0.552	0.698	0.417	0.752	0.328	0.615	0.573
EfficientQAT	2-bit	0.764	0.445	0.542	0.676	0.39	0.732	0.318	0.535	0.550
TAMMP	2.518-bit	0.782	0.466	0.578	0.696	0.443	0.760	0.332	0.636	**0.587**
Llama3.1 8B										
-	FP16	0.801	0.472	0.6	0.74	0.512	0.816	0.332	0.738	0.626
GPTQ	2-bit	0.519	0.344	0.262	0.519	0.195	0.256	0.156	0.215	0.308
OmniQuant	2-bit	0.545	0.336	0.264	0.502	0.188	0.285	0.138	0.198	0.307
EfficientQAT	2-bit	0.752	0.441	0.509	0.668	0.379	0.71	0.258	0.538	0.532
TAMMP	2.465-bit	0.783	0.450	0.541	0.695	0.415	0.74	0.282	0.664	**0.571**

		World Knowledge			Reading Comprehension			MATH			code
Method	**Bits**	**nq_open**	**TriviaQA**	**Avg.**	**SQuAD**	**BoolQ**	**Avg.**	**GSM8K**	**MATHQA**	**Avg.**	**MBPP**
Llama2 7B											
-	FP16	0.189	0.526	0.357	0.263	0.777	0.520	0.051	0.281	0.166	0.284
GPTQ	2-bit	0	0	0	0.075	0.443	0.259	0.006	0.203	0.104	0
OmniQuant	2-bit	0.001	0	0.001	0.104	0.535	0.320	0.010	0.208	0.109	0
QUIP#	2-bi t	0.110	0.272	0.19	0.202	0.708	0.455	0.039	0.261	**0.150**	0.148
EfficientQAT	2-bit	0.091	0.209	0.150	0.197	0.677	0.437	0.031	0.255	0.143	0.125
TAMMP	2.367-bit	0.096	0.330	**0.213**	0.248	0.738	**0.493**	0.039	0.256	0.148	**0.175**
Llama2 13B											
-	FP16	0.236	0.609	0.423	0.298	0.806	0.552	0.086	0.321	0.204	0.369
GPTQ	2-bit	0.001	0.002	0.002	0.076	0.543	0.310	0.013	0.214	0.114	0
OmniQuant	2-bit	0.020	0.059	0.039	0.098	0.618	0.358	0.014	0.237	0.125	0
QUIP#	2-bi t	0.161	0.393	0.277	0.279	0.787	0.523	0.056	0.283	0.170	**0.269**
EfficientQAT	2-bit	0.109	0.344	0.227	0.270	0.756	0.513	0.054	0.278	0.166	0.210
TAMMP	2.518-bit	0.167	0.449	**0.308**	0.291	0.761	**0.526**	0.068	0.292	**0.180**	0.218
Llama3.1 8B											
-	FP16	0.076	0.617	0.346	0.331	0.821	0.576	0.272	0.396	0.334	0.436
GPTQ	2-bit	0	0	0	0.009	0.425	0.217	0.010	0.194	0.102	0
OmniQuant	2-bit	0.001	0.000	0.001	0.079	0.378	0.228	0.006	0.223	0.114	0
EfficientQAT	2-bit	0.088	0.273	0.181	0.190	0.711	0.45	0.028	0.326	0.177	0.023
TAMMP	2.465-bit	0.092	0.358	**0.225**	0.260	0.771	**0.515**	0.094	0.326	**0.210**	**0.284**

Table 1: Performance comparison of 2-bit level quantization methods for Llama series models

2-bit precisions to the identified parameter groups, aiming to balance efficiency and performance.

4.3 Mixed Precision Training

QAT methods often demand substantial hardware and data resources. For instance, the LLM-QAT method (Liu et al., 2023) required a single 8-GPU training node and 100,000 data samples for training completion. In contrast, EfficientQAT (Chen et al., 2024a) achieved quantization-aware training on a single A100 GPU by employing a two-stage training approach, encompassing both block-wise and end-to-end (e2e) phases. We adopt a training approach similar to EfficientQAT, leveraging the two-stage efficient training to update mixed-precision quantized model. In the first stage of block-wise training, the critical column parameters are updated by standard gradient descent method, while the sub-critical and non-critical columns parameters employ the Straight-Through Estimator (STE) for gradient descent. In the second stage of e2e training, all parameters except the critical columns are frozen, training only these specific parameters, which aims to mitigate the layer-dependency issues that can arise from layer-by-layer quantization. This design offers the dual advantage of preserving the mixed-precision method's ability to handle outliers while maintaining the training efficiency of a two-stage approach under limited hardware

		Commonsense Reasoning								
Method	**Bits**	**PIQA**	**SocialIQA**	**HellaSwag**	**WinoGrande**	**ARC-c**	**ARC-e**	**OpenBookQA**	**CommonsenseQA**	**Avg.**
					Qwen2.5-7B					
-	FP16	0.794	0.516	0.620	0.708	0.527	0.818	0.348	0.844	0.647
GPTQ	2-bit	0.546	0.333	0.304	0.496	0.222	0.311	0.132	0.199	0.318
EfficientQAT	2-bit	0.769	0.497	0.525	0.686	0.447	0.757	0.316	0.758	0.594
TAMMP	2.278-bit	0.770	0.501	0.542	0.684	0.462	0.760	0.302	0.749	**0.596**
					Qwen2.5-14B					
-	FP16	0.814	0.542	0.656	0.762	0.607	0.857	0.370	0.844	0.681
GPTQ	2-bit	0.558	0.342	0.314	0.533	0.220	0.297	0.172	0.206	0.330
EfficientQAT	2-bit	0.780	0.509	0.564	0.741	0.517	0.804	0.334	0.766	0.627
TAMMP	2.286-bit	0.787	0.527	0.577	0.733	0.540	0.814	0.350	0.782	**0.639**

		World Knowledge			Reading Comprehension			MATH			code
Method	**Bits**	**nq_open**	**TriviaQA**	**Avg.**	**SQuAD**	**BoolQ**	**Avg.**	**GSM8K**	**MATHQA**	**Avg.**	**MBPP**
					Qwen2.5-7B						
-	FP16	0.045	0.324	0.185	0.211	0.863	0.537	0.713	0.405	0.559	0.603
GPTQ	2-bit	0.000	0.000	0.000	0.034	0.434	0.234	0.012	0.210	0.111	0
EfficientQAT	2-bit	0.058	0.130	0.094	0.165	0.805	0.485	0.325	0.376	0.350	**0.416**
TAMMP	2.278-bit	0.038	0.161	**0.100**	0.208	0.803	**0.506**	0.397	0.356	**0.377**	0.358
					Qwen2.5-14B						
-	FP16	0.047	0.019	0.033	0.123	0.880	0.502	0.642	0.502	0.572	0.704
GPTQ	2-bit	0.000	0.000	0.000	0.023	0.523	0.273	0.007	0.216	0.112	0
EfficientQAT	2-bit	0.067	0.325	0.196	0.132	0.841	0.486	0.271	0.415	0.343	0.455
TAMMP	2.286-bit	0.053	0.402	**0.227**	0.222	0.836	**0.529**	0.334	0.443	**0.388**	**0.549**

Table 2: Performance comparison of 2-bit level quantization methods for Qwen series models

Method	**Bits**	**Commonsense Reasoning**	**World Knowledge**	**Reading Comprehension**	**Math**	**Code**
-	FP16	0.626	0.346	0.576	0.334	0.436
SMP(GitHub)	2.88	0.58	0.247	0.555	0.249	0.401
SMP(C4)	2.88	0.578	0.225	0.545	0.247	0.35
SMP((Wikipedia)	2.88	0.575	0.232	0.554	0.253	0.354
TAMMP	2.93	**0.603**	**0.303**	**0.556**	**0.278**	**0.443**

Table 3: Experimental results of single-source mixed-precision for Llama3.1 8B

resources.

5 Experiments

Our experiments are organized into four main parts: multi-source mixed-precision experiments, single-source mixed-precision experiments, perplexity (PPL) evaluation, and ablations.

5.1 Experimental Settings

Datasets For the purpose of parameter heterogeneity analysis and multi-granular grouping, we curated a multi-source calibration dataset by sampling 128 sequences (length 2048) from each of the six RedPajama (Together Computer, 2023) subsets (i.e., C4, Wikipedia, GitHub, StackExchange, ArXiv, and Commoncrawl). For the mixed-precision training phase, we respectively utilized 4096 and 50,000 samples with a context length of 2048 derived from RedPajama in block-wise and e2e training.

Bit Allocation In the multi-source mixed-precision experiments, we employ the two multi-granular parameter grouping strategies introduced in the preceding section. A further rationale for adopting these strategies is to facilitate comparative experiments against existing 2-bit and 3-bit quantization works, respectively. During the structural column search, to regulate the proportion of probed critical columns, we cap the maximum number ($topk$) of critical columns at 10 per group, with a group size of 128. The detailed proportions of parameter groups for each base model (i.e., Llama2, Llama3, Qwen3) are provided in the **Appendix**. For the single-source mixed-precision experiments, the selected knowledge sources are C4, Wikipedia, and GitHub. Here, we adopt a dual-granularity parameter grouping scheme, where critical parameters account for 6% and non-critical parameters for 94%, yielding an average quantization bit-width of 2.88 bits. For the PPL evaluation, instead of utilizing Structural Searching Selection method for critical parameter probing, we employ a group-

ing method based on sorting parameter importance metrics and set up 20 experiments with average bit-widths uniformly spaced between 2 bits and 4 bits.

Hyperparameters In the block-wise phase of mixed-precision training, each block is trained for 2 epochs with a batch size of 2. We configure the learning rate to 1×10^{-4} and 2×10^{-5} for quantization parameters and weights. Subsequently, in the e2e phase, we fine-tune the entire model for 1 epoch with a batch size of 32, setting the learning rate for the quantization step size to 2×10^{-5}.

Baselines The test results are compared against a variety of quantization methods, including:

- Uniform PTQ methods: GPTQ (Frantar et al., 2022), OmniQuant(Shao et al., 2023) and QUIP#(Tseng et al., 2024).
- Mixed-precision PTQ method: OWQ (Lee et al., 2024).
- QAT method: EfficientQAT (Chen et al., 2024a).

The 2-bit and 3-bit quantization of GPTQ were implemented via Auto-GPTQ. For all other quantization methods, we quantized the model via the open-source code provided by their respective authors. The absence of results for certain baselines in our experiments is attributed to limitations in their open-source implementations.

Evaluation Dimensions for Model Capabilities To demonstrate that our proposed TAMMP method possesses stronger generalization capabilities compared to quantized models of similar scale, we evaluated its performance across five key dimensions: commonsense reasoning (PIQA, social_iqa, HellaSwag, WinoGrande, ARC-c, ARC-e, OpenBookQA, CommonsenseQA)), world knowledge (nq_open, TriviaQA), reading comprehension (SQuAD, BoolQ), math (GSM8K, MATHQA), and code (MBPP). Each dimension task comprises several relevant public datasets, and we used average accuracy as the metric to quantify model capability. Detailed test results for the quantized model across all tasks are available in the Appendix. For the MBPP testing, we utilized OpenCompass to compute pass@1. For all other downstream tasks, we employed lm_eval v0.4.2 to compute accuracy.

5.2 Main Results

Firstly, the multi-source experiments are designed to validate the ability of our TAMMP method to address the impact of the critical parameter heterogeneity by leveraging critical parameter probing based on multi-source knowledge and multi-granular bit allocation. In the 2-bit (Table 1, Table 2) and 3-bit (Table 8, Table 7 in **Appendix**) quantization regime, TAMMP exhibits comprehensive superiority over current approaches. Secondly, the single-source experiments serve to verify that retaining the precision of critical parameters by knowledge localization, indeed boosts the quantized model's capabilities on task-specific benchmarks (Table 3). Thirdly, the PPL evaluation assesses the stability of our mixed-precision models under different bit-width settings (Figure 5). Finally, the ablation studies are presented in Table 4.

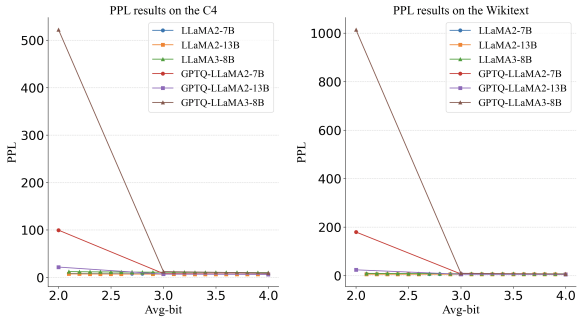

Figure 5: Perplexity results of mixed-precision models and GPTQ.

5.3 Multi-source Mixed-Precision Experiments

Model quantization at 2-bit level as shown in Table 1, a significant performance gap exists between PTQ methods and full-precision models, with PTQ methods even demonstrating inability in tasks requiring world knowledge and code generation for Llama series models. Our proposed TAMMP method with multi-granular parameter grouping strategy 1, significantly enhances the model's generalization capabilities. It achieves 90% of the full-precision model's performance on commonsense reasoning and reading comprehension tasks. Furthermore, on world knowledge and code generation tasks, TAMMP yields approximately a 30% relative improvement compared to the EfficientQAT method. Similar conclusions were validated on the Qwen series models, as shown in Table 2. Results for Strategy 2 at the 3-bit level are detailed in the **Appendix**. TAMMP consistently surpasses baselines across the majority of tasks.

		Commonsense Reasoning								
Method	**Bits**	**PIQA**	**SocialIQA**	**HellaSwag**	**WinoGrande**	**ARC-c**	**ARC-e**	**OpenBookQA**	**CommonsenseQA**	**Avg.**
Llama2 7B	FP16	0.781	0.460	0.571	0.691	0.435	0.763	0.314	0.575	0.574
strategy1 topk=10	2.367-bit	0.758	0.440	0.528	0.658	0.379	0.725	0.296	0.396	0.522
strategy1 topk=20	2.51-bits	0.751	0.441	0.535	0.658	0.389	0.732	0.312	0.396	0.527
strategy1 topk=30	2.84-bits	0.758	0.444	0.528	0.662	0.387	0.722	0.286	0.421	0.526
strategy2 topk=10	2.918-bit	0.759	0.448	0.548	0.676	0.410	0.740	0.318	0.525	0.553
strategy2 topk=20	2.89-bits	0.760	0.439	0.548	0.683	0.404	0.722	0.418	0.464	**0.555**
strategy2 topk=30	3.24-bit	0.768	0.452	0.553	0.667	0.410	0.745	0.330	0.495	0.553

		World Knowledge			Reading Comprehension			MATH			Code
Method	**Bits**	**nq_open**	**TriviaQA**	**Avg.**	**SQuAD**	**BoolQ**	**Avg.**	**GSM8K**	**MATHQA**	**Avg.**	**MBPP**
Llama2 7B	FP16	0.189	0.526	0.357	0.263	0.777	0.520	0.051	0.281	0.166	0.284
strategy1 topk=10	2.367-bit	0.096	0.330	0.213	0.248	0.738	0.493	0.039	0.256	0.148	0.175
strategy1 topk=20	2.51-bits	0.105	0.363	0.234	0.248	0.739	0.493	0.041	0.275	0.158	0.179
strategy1 topk=30	2.84-bits	0.104	0.297	0.200	0.233	0.744	0.489	0.041	0.263	0.152	0.187
strategy2 topk=10	2.918-bit	0.123	0.415	0.269	0.342	0.758	0.550	0.443	0.039	**0.276**	**0.280**
strategy2 topk=20	2.89-bits	0.083	0.415	0.249	0.375	0.745	**0.560**	0.045	0.274	0.159	0.218
strategy2 topk=30	3.24-bit	0.116	0.437	**0.277**	0.296	0.766	0.531	0.049	0.262	0.155	0.265

Table 4: Multi-source mixed-precision experimental results under different bit allocation strategies and column search ranges

5.4 Single-source Mixed-Precision Experiments

To validate that preserving the precision of critical parameters in mixed-precision quantization effectively enhances performance on corresponding downstream tasks, we conducted single-source mixed-precision experiments on the Llama3 8B model. As shown in Table 3 , when the knowledge source used for critical parameter probing is GitHub, the resulting quantized model significantly outperforms models calibrated with C4 and Wikipedia on code generation tasks. This validates the rationale of enhancing performance on relevant tasks through knowledge-based critical parameter probing and mixed-precision training. Despite comparable average bit-widths, single-source methods significantly underperform the 3-bit TAMMP across the board. This highlights the distinct advantage of our multi-granular bit-width allocation derived from multi-source knowledge.

5.5 PPL Evaluation

The results from perplexity experiments, shown in Figure 5, demonstrate that when the mixed-precision model's average bit-width falls between 2 and 4 bits, the PPL metric remains stable within a small range. This is a significant improvement over methods like GPTQ, where a substantial gap in PPL often appears between 2-bit and 4-bit quantized models. This robust performance further indicates that our proposed mixed-precision design method yields quantized models with stable performance and strong bit-width manipulability.

5.6 Ablations

Table 4 presents the ablation results for different allocation strategies and $topk$. Even with comparable average bit-widths, Strategy 2 demonstrates comprehensive superiority over Strategy 1. This observation indicates that despite the importance of critical parameters, their scarcity limits their overall impact. Therefore, it proves to be more cost-effective to appropriately reduce the ratio of critical parameters in exchange for a higher proportion of sub-critical parameters.

6 Conclusion

This paper addresses the issue of generalization in low-bit quantized models by pioneering the concept of critical parameter heterogeneity. We posit that the neglect of this phenomenon prior works is a primary cause of generalization deterioration.

Consequently, we design TAMMP, which leverages multi-source knowledge probing and multi-granular bit allocation to alleviate the performance degradation stemming from important parameter heterogeneity. Extensive experiments on the Llama and Qwen series confirm that TAMMP achieves superior generalization compared to existing approaches across a wide range of tasks. Moreover, in addition to offering flexible bit-width operability, our method successfully averts the precipitous performance decline often seen in quantized models as they drop from 3-bit to 2-bit.

7 Limitations

Despite exhibiting superior generalization performance, TAMMP relies on a heuristic strategy (i.e.,

fixed allocation of FP16/3-bit/2-bit for different parameter groups) without integrating hardware-aware metrics or latency factors. As a result, it entails greater computational costs during inference relative to models with uniform precision. It is worth noting that this trade-off is a prevalent issue across mixed-precision quantization methods. We look forward to subsequent studies addressing these efficiency challenges to achieve lower inference latency.

References

Sebastien Bubeck, Varun Chandrasekaran, Ronen Eldan, and 1 others. 2023. Sparks of artificial general intelligence: Early experiments with GPT-4. *arXiv preprint arXiv:2303.12712*.

Mengzhao Chen, Wenqi Shao, Peng Xu, and 1 others. 2024a. EfficientQAT: Efficient quantization-aware training for large language models. *arXiv preprint arXiv:2407.11062*.

Zhe Chen, Bowen Xie, Jian Li, and 1 others. 2024b. Channel-wise mixed-precision quantization for large language models. *arXiv preprint arXiv:2410.13056*.

Wenbin Cui and Qi Wang. 2024. Cherry on top: Parameter heterogeneity and quantization in large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 60607–60625.

Tim Dettmers, Mike Lewis, Younes Belkada, and 1 others. 2022. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*, volume 35, pages 30318–30332.

Tim Dettmers, Roman Svirschevski, Vladimir Egiazarian, and 1 others. 2023. SPQR: A sparse-quantized representation for near-lossless LLM weight compression. *arXiv preprint arXiv:2306.03078*.

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*.

Rui Gong, Yao Ding, Zhiyong Wang, and 1 others. 2025. A survey of low-bit large language models: Basics, systems, and algorithms. *Neural Networks*, page 107856.

Bo He, Li Yin, Hao Zhen, and 1 others. 2025. Preserving LLM capabilities through calibration data curation: From analysis to optimization. *arXiv preprint arXiv:2510.10618*.

Wenqi Huang, Yifan Liu, Hao Qin, and 1 others. 2024. BiLLM: Pushing the limit of post-training quantization for LLMs. *arXiv preprint arXiv:2402.04291*.

Hyunwoo Kang, Qi Zhang, Souvik Kundu, and 1 others. 2024. GEAR: An efficient KV cache compression recipe for near-lossless generative inference of LLMs. *arXiv preprint arXiv:2403.05527*.

Jaeho Kim, Joon-Ho Lee, Seong Kim, and 1 others. 2023a. Memory-efficient fine-tuning of compressed large language models via sub-4-bit integer quantization. In *Advances in Neural Information Processing Systems*, volume 36, pages 36187–36207.

Sehoon Kim, Connor Hooper, Amir Gholami, and 1 others. 2023b. SqueezeLLM: Dense-and-sparse quantization. *arXiv preprint arXiv:2306.07629*.

Chang Lee, Jaeho Jin, Taewoo Kim, and 1 others. 2024. OWQ: Outlier-aware weight quantization for efficient fine-tuning and inference of large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 13355–13364.

Ji Lin, Jian Tang, Hao Tang, and 1 others. 2024. AWQ: Activation-aware weight quantization for on-device LLM compression and acceleration. In *Proceedings of Machine Learning and Systems*, volume 6, pages 87–100.

Zihan Liu, Baris Oguz, Chen Zhao, and 1 others. 2023. LLM-QAT: Data-free quantization aware training for large language models. *arXiv preprint arXiv:2305.17888*.

Lin Ma, Ming Sun, and Zhi Shen. 2024a. Fbi-llm: Scaling up fully binarized llms from scratch via autoregressive distillation. *arXiv preprint arXiv:2407.07093*.

Shuming Ma, Haoyu Wang, Liang Ma, and 1 others. 2024b. The era of 1-bit LLMs: All large language models are in 1.58 bits.

Yu Shang, Zhe Yuan, Qian Wu, and 1 others. 2023. PB-LLM: Partially binarized large language models. *arXiv preprint arXiv:2310.00034*.

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

Together Computer. 2023. RedPajama: An open dataset for training large language models.

Hugo Touvron, Louis Martin, Kevin Stone, and 1 others. 2023. LLaMA 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Alex Tseng, Jiaming Chee, Qifan Sun, and 1 others. 2024. Quip#: Even better LLM quantization with hadamard incoherence and lattice codebooks. *arXiv preprint arXiv:2402.04396*.

Haoyu Wang, Shuming Ma, Li Dong, and 1 others. 2023. BitNet: Scaling 1-bit transformers for large language models. *arXiv preprint arXiv:2310.11453*.

Mark Williams and Nikolaos Aletras. 2024. On the impact of calibration data in post-training quantization and pruning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10100–10118.

Yuhui Xu, Xiaotian Han, Zhen Yang, and 1 others. 2024. OneBit: Towards extremely low-bit large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 66357–66382.

Kun Ying, Fan Meng, Jian Wang, and 1 others. 2024. MMT-Bench: A comprehensive multimodal benchmark for evaluating large vision-language models towards multitask AGI. *arXiv preprint arXiv:2404.16006*.

Appendix

A Calculation of Jaccard-based Compactness

Consensus-based Jaccard Similarity. Assume that the dataset is partitioned into K sub-datasets $\{A_1, A_2, \ldots, A_K\}$, where each A_k is a set of elements. For any element x in the universe Ω, we define its occurrence frequency across sub-datasets as

$$f(x) = \sum_{k=1}^{K} \mathbf{1}(x \in A_k), \tag{5}$$

where $\mathbf{1}(\cdot)$ is the indicator function.

Given a threshold $p \in (0, K]$, we construct a consensus set A_{core} as

$$A_{\text{core}} = \{x \in \Omega \mid f(x) \geq p\}. \tag{6}$$

The compactness between the k-th sub-dataset and the global structure is then quantified using the Jaccard similarity with respect to the consensus set:

$$J_k^{\text{core}} = \frac{|A_k \cap A_{\text{core}}|}{|A_k \cup A_{\text{core}}|}. \tag{7}$$

Finally, the overall compactness among sub-datasets is defined as the average consensus-based Jaccard similarity:

$$\bar{J}^{\text{core}} = \frac{1}{K} \sum_{k=1}^{K} J_k^{\text{core}}. \tag{8}$$

B Bit-width Allocation Results

This section provides additional information on the bit-width allocation results. Using our data-driven bit-width allocation method, model parameters are grouped into critical, semi-critical, and non-critical columns, which are then represented using float16, 3-bit, and 2-bit precision, respectively. The main text outlines two distinct bit-width allocation strategies. Table 5 presents the proportion of each parameter group and the final average bit-width for both strategies on the Llama and Qwen series models.

Algorithm 1 Bit-width Allocation

Input: Weight matrix $W \in R^{n \times m}$, Hessian inverse $H^c \in R^{n \times n}$

Output: `critical_c`, `semi_critical_c`, `non_critical_c`

$S \leftarrow W^2 \oslash \left(H^c_{b:b+\beta,\, b:b+\beta}\right)^2$

`rows1` $\leftarrow$ `topk`$(\sum \|S\|, \dim = 0)$

`rows2` $\leftarrow$ `topk`$(-\sum \|S\|, \dim = 0)$

`critical_c` $\leftarrow$ `Structural_Searching`$(W, H^c,$ `rows1`$)$

`non_critical_c` $\leftarrow$ `Structural_Searching`$(W, H^c,$ `rows2`$)$

`sub_critical_c` $\leftarrow \{j \mid j \notin$ `critical_c` $\wedge\ j \notin$ `non_critical_c`$\}$

return `critical_c`, `sub_critical_c`, `non_critical_c`

C Algorithmic procedure

D Additional Experimental Results

In this section, we provide further experimental results that were omitted from the main paper due to space limitations. The quantized model's capabilities were evaluated across five dimensions: commonsense reasoning, reading comprehension, world knowledge, mathematics, and code generation. Each dimension included several benchmark tasks, and calculate the average score for the final metric. Table **??** presents the quantization results for the Llama model specifically for the commonsense reasoning dimension, while Table **??** shows the corresponding results for the Qwen series models. The experimental results for reading comprehension, world knowledge, mathematics, and code generation for the Llama and Qwen series models can be found in Table **??** and Table **??** , respectively.

model	Strategy 1				Strategy 2			
	critical	sub	non-critical	Avg. bit-width	critical	sub	non-critical	Avg. bit-width
Llama2 7B	0.018	0.117	0.865	2.367	0.019	0.654	0.327	2.918
Llama2 13B	0.021	0.226	0.753	2.518	0.02	0.66	0.32	2.944
Llama3.1 8B	0.027	0.091	0.882	2.465	0.02	0.652	0.328	2.93
Qwen2.5-7B	0.015	0.062	0.923	2.278	0.015	0.709	0.276	2.926
Qwen2.5-14B	0.017	0.044	0.939	2.286	0.017	0.72	0.263	2.962

Table 5: Bit-width allocation results of different models with the two strategies.

		Commonsense Reasoning								
Method	**Bits**	**PIQA**	**SocialIQA**	**HellaSwag**	**WinoGrande**	**ARC-c**	**ARC-e**	**OpenBookQA**	**CommonsenseQA**	**Avg.**
Llama2 7B										
-	FP16	0.781	0.460	0.571	0.691	0.435	0.763	0.314	0.575	0.574
GPTQ	3-bit	0.769	0.45	0.525	0.665	0.387	0.720	0.262	0.455	0.529
OmniQuant	3-bit	0.774	0.441	0.544	0.675	0.400	0.741	0.306	0.46	0.543
OWQ	3.1-bit	0.756	0.426	0.524	0.667	0.375	0.719	0.294	0.421	0.523
QUIP#	3-bit	0.772	0.450	0.558	0.683	0.418	0.745	0.312	0.491	0.553
EfficientQAT	3-bit	0.769	0.449	0.559	0.683	0.428	0.749	0.304	0.504	**0.556**
TAMMP	2.918-bit	0.759	0.448	0.548	0.676	0.41	0.740	0.318	0.525	0.553
Llama2 13B										
-	FP16	0.792	0.473	0.600	0.721	0.484	0.794	0.35	0.683	0.612
GPTQ	3-bit	0.770	0.456	0.572	0.707	0.423	0.756	0.316	0.62	0.578
OmniQuant	3-bit	0.781	0.442	0.583	0.695	0.448	0.782	0.336	0.641	0.588
OWQ	3.1-bit	0.767	0.445	0.527	0.682	0.433	0.768	0.304	0.611	0.567
QUIP#	3-bit	0.781	0.466	0.583	0.725	0.446	0.779	0.322	0.654	0.594
EfficientQAT	3-bit	0.787	0.473	0.592	0.712	0.469	0.784	0.35	0.668	0.605
TAMMP	2.944-bit	0.792	0.47	0.592	0.717	0.463	0.781	0.356	0.674	**0.606**
Llama3.1 8B										
-	FP16	0.801	0.472	0.6	0.74	0.512	0.816	0.332	0.738	0.626
GPTQ	3-bit	0.760	0.426	0.535	0.688	0.408	0.724	0.254	0.618	0.551
OmniQuant	3-bit	0.769	0.449	0.532	0.689	0.41	0.736	0.306	0.57	0.557
EfficientQAT	3-bit	0.795	0.469	0.581	0.732	0.482	0.798	0.304	0.704	**0.608**
TAMMP	2.93-bit	0.784	0.461	0.575	0.714	0.476	0.788	0.32	0.706	0.603

		World Knowledge			Reading Comprehension			MATH			code
Method	**Bits**	**nq_open**	**TriviaQA**	**Avg.**	**SQuAD**	**BoolQ**	**Avg.**	**GSM8K**	**MATHQA**	**Avg.**	**MBPP**
Llama2 7B											
-	FP16	0.189	0.526	0.357	0.263	0.777	0.520	0.051	0.281	0.166	0.284
GPTQ	3-bit	0.058	0.202	0.130	0.329	0.671	0.500	0.034	0.261	0.147	0.160
OmniQuant	3-bit	0.099	0.415	0.257	0.280	0.718	0.499	0.044	0.278	0.161	0.230
OWQ	3.1-bit	0.090	0.300	0.190	0.270	0.720	0.500	0.030	0.270	0.150	0.128
QUIP#	3-bit	0.174	0.487	**0.331**	0.291	0.756	0.524	0.044	0.279	0.162	0.218
EfficientQAT	3-bit	0.093	0.391	0.242	0.258	0.784	0.521	0.046	0.275	0.161	0.272
TAMMP	2.918-bit	0.123	0.415	0.269	0.342	0.758	**0.550**	0.443	0.039	**0.276**	**0.280**
Llama2 13B											
-	FP16	0.236	0.609	0.423	0.298	0.806	0.552	0.086	0.321	0.204	0.369
GPTQ	3-bit	0.170	0.457	0.313	0.320	0.765	0.542	0.058	0.293	0.175	0.276
OmniQuant	3-bit	0.189	0.498	0.343	0.244	0.778	0.511	0.065	0.300	0.183	**0.331**
OWQ	3.1-bit	0.154	0.490	0.322	0.280	0.769	0.525	0.061	0.300	0.180	0.221
QUIP#	3-bit	0.215	0.571	0.393	0.294	0.800	0.547	0.072	0.294	0.183	0.272
EfficientQAT	3-bit	0.224	0.595	**0.409**	0.300	0.790	0.545	0.071	0.302	0.186	0.167
TAMMP	2.944-bit	0.206	0.527	0.366	0.306	0.789	**0.547**	0.070	0.304	**0.187**	0.179
Llama3.1 8B											
-	FP16	0.076	0.617	0.346	0.331	0.821	0.576	0.272	0.396	0.334	0.436
GPTQ	3-bit	0.041	0.232	0.136	0.237	0.684	0.461	0.032	0.302	0.167	0.078
OmniQuant	3-bit	0.044	0.346	0.195	0.249	0.731	0.490	0.066	0.319	0.193	0.342
EfficientQAT	3-bit	0.067	0.468	0.267	0.289	0.797	0.543	0.166	0.379	0.273	0.058
TAMMP	2.93-bit	0.133	0.474	**0.303**	0.300	0.811	**0.556**	0.174	0.383	**0.278**	**0.444**

Table 6: Performance result of World Knowledge, Reading Comprehension, Math and Code for Llama series models

		Commonsense Reasoning								
Method	**Bits**	**PIQA**	**SocialIQA**	**HellaSwag**	**WinoGrande**	**ARC-c**	**ARC-e**	**OpenBookQA**	**CommonsenseQA**	**Avg.**
					Qwen2.5-7B					
-	FP16	0.794	0.516	0.620	0.708	0.527	0.818	0.348	0.844	0.647
GPTQ	3-bit	0.764	0.5	0.582	0.664	0.495	0.772	0.324	0.763	0.608
EfficientQAT	3-bit	0.795	0.545	0.578	0.721	0.534	0.830	0.330	0.818	**0.644**
TAMMP	2.926-bit	0.785	0.532	0.570	0.710	0.532	0.821	0.330	0.822	0.638
					Qwen2.5-14B					
-	FP16	0.814	0.542	0.656	0.762	0.607	0.857	0.370	0.844	0.681
GPTQ	3-bit	0.806	0.487	0.620	0.717	0.532	0.801	0.348	0.815	0.641
EfficientQAT	3-bit	0.813	0.547	0.615	0.786	0.562	0.845	0.370	0.842	**0.673**
TAMMP	2.962-bit	0.806	0.540	0.605	0.772	0.570	0.842	0.356	0.839	0.666

		World Knowledge			Reading Comprehension			MATH			code
Method	**Bits**	**nq_open**	**TriviaQA**	**Avg.**	**SQuAD**	**BoolQ**	**Avg.**	**GSM8K**	**MATHQA**	**Avg.**	**MBPP**
						Qwen2.5-7B					
-	FP16	0.045	0.324	0.185	0.211	0.863	0.537	0.713	0.405	0.559	0.603
GPTQ	3-bit	0.006	0.212	0.109	0.145	0.835	0.490	0.324	0.350	0.337	0
EfficientQAT	3-bit	0.095	0.458	0.276	0.340	0.854	0.597	0.582	0.450	0.516	**0.646**
TAMMP	2.926-bit	0.109	0.458	**0.283**	0.347	0.854	**0.6**	0.610	0.443	**0.527**	0.576
						Qwen2.5-14B					
-	FP16	0.047	0.019	0.033	0.123	0.880	0.502	0.642	0.502	0.572	0.704
GPTQ	3-bit	0.003	0.017	0.010	0.117	0.869	0.493	0.347	0.416	0.381	0.549
GPTQ	2-bit	0.000	0.000	0.000	0.023	0.523	0.273	0.007	0.216	0.112	0
EfficientQAT	3-bit	0.077	0.190	0.134	0.187	0.876	0.531	0.465	0.507	0.486	0.638
EfficientQAT	2-bit	0.067	0.325	0.196	0.132	0.841	0.486	0.271	0.415	0.343	0.455
TAMMP	2.962-bit	0.068	0.449	**0.258**	0.192	0.875	**0.533**	0.674	0.510	**0.592**	**0.689**

Table 7: Performance result of World Knowledge, Reading Comprehension, Math and Code for Qwen series models

		Commonsense Reasoning								
Method	**Bits**	**PIQA**	**SocialIQA**	**HellaSwag**	**WinoGrande**	**ARC-c**	**ARC-e**	**OpenBookQA**	**CommonsenseQA**	**Avg.**
					Llama3.1 8B					
-	FP16	0.801	0.472	0.6	0.74	0.512	0.816	0.332	0.738	0.626
SMP(GitHub)	2.88-bit	0.766	0.459	0.540	0.682	0.443	0.752	0.298	0.700	**0.580**
SMP(C4)	2.88-bit	0.770	0.455	0.543	0.691	0.423	0.748	0.290	0.700	0.578
SMP((Wikipedia)	2.88-bit	0.762	0.454	0.541	0.688	0.433	0.748	0.288	0.686	0.575

		World Knowledge			Reading Comprehension			MATH			code
Method	**Bits**	**nq_open**	**TriviaQA**	**Avg.**	**SQuAD**	**BoolQ**	**Avg.**	**GSM8K**	**MATHQA**	**Avg.**	**MBPP**
						Llama3.1 8B					
-	FP16	0.076	0.617	0.346	0.331	0.821	0.576	0.272	0.396	0.334	0.436
SMP(GitHub)	2.88-bit	0.110	0.384	**0.247**	0.336	0.773	**0.555**	0.158	0.339	0.249	**0.401**
SMP(C4)	2.88-bit	0.102	0.347	0.225	0.326	0.764	0.545	0.162	0.331	0.247	0.350
SMP((Wikipedia)	2.88-bit	0.101	0.362	0.232	0.343	0.765	0.554	0.156	0.350	**0.253**	0.354

Table 8: Performance result of World Knowledge, Reading Comprehension, Math and Code for Llama series models