Mixture of Scales: Memory-Efficient Token-Adaptive Binarization for Large Language Models

Dongwon Jo [1] **Taesu Kim** [2] **Yulhwa Kim** [3]* **Jae-Joon Kim** [1]*
[1] Seoul National University [2] SqueezeBits Inc. [3] Sungkyunkwan University
{dongwonjo, kimjaejoon}@snu.ac.kr
{taesu.kim}@squeezebits.com
{yulhwakim}@skku.edu

Abstract

Binarization, which converts weight parameters to binary values, has emerged as an effective strategy to reduce the size of large language models (LLMs). However, typical binarization techniques significantly diminish linguistic effectiveness of LLMs. To address this issue, we introduce a novel binarization technique called Mixture of Scales (BinaryMoS). Unlike conventional methods, BinaryMoS employs multiple scaling experts for binary weights, dynamically merging these experts for each token to adaptively generate scaling factors. This token-adaptive approach boosts the representational power of binarized LLMs by enabling contextual adjustments to the values of binary weights. Moreover, because this adaptive process only involves the scaling factors rather than the entire weight matrix, BinaryMoS maintains compression efficiency similar to traditional static binarization methods. Our experimental results reveal that BinaryMoS surpasses conventional binarization techniques in various natural language processing tasks and even outperforms 2-bit quantization methods, all while maintaining similar model size to static binarization techniques.

1 Introduction

Though large language models (LLMs) have delivered impressive results in a variety of natural language processing (NLP) tasks, their massive size often complicates deployment. One common method to compress LLMs is through the quantization of weight parameters, which reduces model sizes by lowering the precision of weight values [1, 8, 2, 30, 31, 32, 33, 3, 4]. Existing quantization approaches such as GPTQ [2], AWQ [3], and OWQ [4] have successfully managed to reduce model sizes by converting 16-bit floating point weights to 4-bit representations, achieving a fourfold decrease in size. Binarization pushes this concept even further by reducing weight values to 1-bit, resulting in a 16-fold size reduction.

However, such aggressive compression through binarization drastically limits the representational capacity of weights, leading to a significant degradation in the linguistic capabilities of LLMs. To address this limitation and improve the accuracy of binarized LLMs, recent research has actively explored binarization techniques tailored for LLMs [29, 5, 6, 7]. Nonetheless, previous efforts often compromise the inherent advantages of binarization by introducing high memory overhead, and they continue to struggle to achieve sufficient accuracy with binarized LLMs.

In this paper, we propose a novel binarization technique named as Mixture of Scales (BinaryMoS). Typical binarization methods use scaling factors to control the effective values of binarized weights. Although these scaling factors occupy a tiny fraction of the overall model size, they are crucial

*Corresponding Author

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

https://doi.org/10.52202/079017-4367

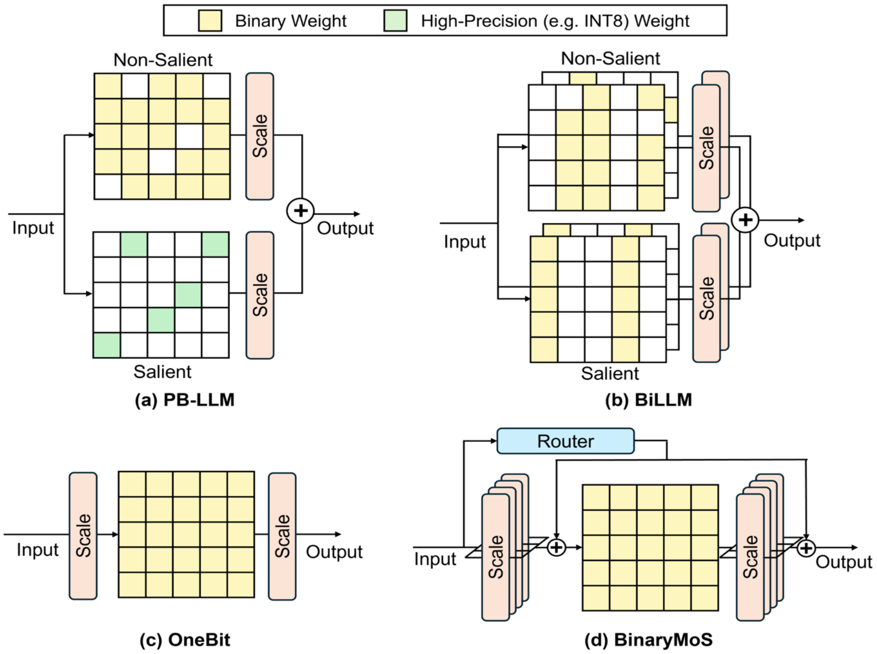


Figure 1: A brief overview of various LLM binarization methods. PB-LLM involves both a binary weight matrix and a high-precision, sparse weight matrix, and BiLLM stores four types of binary weight matrices. OneBit simplifies the layer structure by introducing scaling factors for input and output dimensions respectively. BinaryMoS introduces multiple scaling experts to enhance the capacity of binarized models.

in reducing binarization error. BinaryMoS advances the functionality of these scaling factors by incorporating token-adaptive scaling factors. Inspired by the Mixture of Experts (MoE) approach [26, 27, 34], which empolys multiple expert layers to enhance the model capacity, BinaryMoS adopts multiple scaling factors as experts to improve the representational capacity of binarized LLMs in a memory-efficient way. During inference, BinaryMoS linearly combines these scaling experts based on the context to generate token-adaptive scaling factors, thus dynamically adjusting the represented values of binarized weights to maximize the expressive power of the model. As a result, BinaryMoS can improve the linguistic performance of binarized LLMs with minimal memory overhead.

2 Background

2.1 Binarization of LLMs

Binarization stands out as an extreme yet effective method for reducing model sizes in deep learning. This method achieves size reduction by transforming high-precision weight parameters into 1-bit values. The binarization process is typically governed by the following equation:

$$W_B = \alpha \cdot \text{Sign}(W_{FP} - \overline{W}_{FP}) \tag{1}$$

Here, $W_{FP} \in \mathbb{R}^{n \times m}$ is the full-precision weight matrix of a linear layer where n and m represent the size of output and input dimension, respectively, and $W_B \in \mathbb{R}^{n \times m}$ denotes its binarized version. $\alpha \in \mathbb{R}^n$ represents scaling factors that are responsible for adjusting the binary weight values along the output dimension. In general, the scaling factors are analytically derived as the absolute mean of FP weight values to minimize the $L2$ error between full-precision and binarized weights, and these scaling factors play a vital role in bridging the gap between the original full-precision weights and their binarized counterparts.

While binarization has been effectively applied in traditional deep learning models like Convolutional Neural Networks (CNNs) for image classification without losing accuracy [9, 11, 10, 12], LLMs tend to be more sensitive to such extreme quantization, often experiencing significant accuracy degradation with standard binarization techniques. Therefore, various binarziation techniques tailored for LLMs

https://doi.org/10.52202/079017-4367

have been developed, as shown in Figure 1. PB-LLM [5] partially binarizes weight parameters while maintaining salient weight parameters as high-precision values (e.g., Float16 or INT8). However, this method results in considerable memory overhead. For instance, quantizing 10% of weight parameters as INT8 while binarizing the remaining 90% results in an average bit-width of 1.7 bits for the weight parameters, which is closer to 2 bits than 1 bit.

Furthermore, despite this partial binarization strategy of PB-LLM, the significant information loss inherent in binarization still causes considerable accuracy degradation. To reduce the binarization error and enhance accuracy, BiLLM [6] adopts a more refined approach to assigning scaling factors. Assuming that weight parameters follow a bell-shaped distribution, BiLLM categorizes weight parameters based on their proximity to the mean value: concentrated weights, close to the mean, and sparse weights, distant from the mean. Distinct scaling factors are then assigned to each group to minimize binarization errors. Then, to reduce the memory overhead associated with maintaining information of salient weights, BiLLM preserves this information by binarizing the difference between the binarized values and their full-precision counterparts. Consequently, each salient weight is represented by two 1-bit values, effectively amounting to a 2-bit representation. Despite significantly reducing binarization error, BiLLM complicates the structure of binarized LLMs, adding complexity to the inference process. This complexity arises from the need to manage additional sparse and salient weights alongside regular concentrated weights, requiring extra matrix multiplication during inference.

Meanwhile, unlike conventional binarization methods that typically employ scaling factors only for the output dimension of weights, OneBit [7] enhances the binarization process by incorporating scaling factors for both the input and output dimensions. This dual-dimension scaling approach addresses binarization errors across both dimensions, potentially enhancing model accuracy. Additionally, the size of each scaling vector is substantially smaller compared to the weight matrix, making this approach memory efficient. For instance, in linear layers with a hidden dimension of h, the weight matrix size is $h \times h$, while each scaling vector is only $h \times 1$. Therefore, doubling these scaling factors adds a negligible memory overhead to the network. Moreover, as this approach of dual-dimensional scaling efficiently preserves enough information to significantly reduce binarization errors, OneBit eliminates the need to store separate information for salient weights, thereby simplifying the model structure. The result of matrix multiplication Y of a linear layer using the OneBit approach can be defined as follows:

$$Y = X[S_{in}^T \odot \text{Sign}(W_{FP}^T) \odot S_{out}] = [(X \odot S_{in})\text{Sign}(W_{FP}^T)] \odot S_{out} \tag{2}$$

Here, $X \in \mathbb{R}^{k \times m}$ is the matrix of input activation where k represents batch size, while $S_{in} \in \mathbb{R}^{1 \times m}$ and $S_{out} \in \mathbb{R}^{1 \times n}$ denote the scaling factors for input and output dimensions, respectively. As outlined in Equation 2, processing scaling factors for both input and output dimension can be simplified to scaling input and output of the linear layer before and after matrix multiplication, respectively.

Despite advances in binarization techniques for LLMs, a notable accuracy gap still exists between full-precision models and their binarized counterparts. Therefore, bridging this gap without sacrificing the fundamental benefits of binarization, particularly low memory usage, remains an important challenge in the field of LLM compression.

2.2 Mixture of Experts

The MoE approach is a widely adopted strategy to boost the capabilites of deep learning models by integrating multiple specialized experts into a single framework [26, 27, 34]. Typically, the MoE approach for LLMs involves duplication of layers and selecting the appropriate layers among these duplicates for a specific task during inference. In the MoE setup, the router is a key to selecting the appropriate expert. It generally consists of a linear layer followed by a softmax function, which calculates and assigns scores to each expert. During the inference, only the experts with the highest score are selected and processed.

While integrating the MoE approach with binarized LLMs offers potential for improving model accuracy, it presents a substantial memory trade-off. The duplication of layers inherent in MoE increases the model size proportionally with the number of experts, thus diminishing the memory efficiency benefits gained from binarization. To address these challenges, we propose BinaryMoS, a novel binarization technique that aims to enhance model capacity while maintaining memory efficiency. This approach leverages scaling factors as experts, improving accuracy of binarized

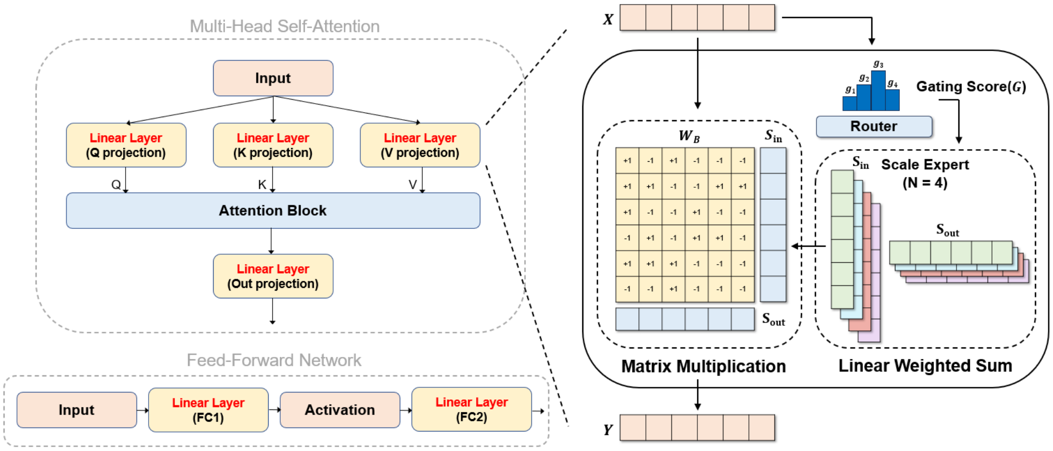


Figure 2: Illustration of the proposed BinaryMoS scheme. The proposed BinaryMoS introduce mixture of scale approach to generate token-adaptive scaling factors.

LLMs without the extensive memory overhead associated with traditional MoE configurations. In the following section, we will delve deeper into how BinaryMoS operates and its benefits over conventional techniques.

3 Proposed BinaryMoS

3.1 Binarization with Mixture of Scale

An overview of the proposed BinaryMoS is presented in Figure 2. Unlike previous binarization techniques that utilize a single scaling vector per input or output dimension, BinaryMoS integrates the concept of experts from the MoE framework into the scaling factors and utilizes multiple scaling experts for each dimension. As discussed in Section 2.1, although the size of scaling factors is relatively small, they play a crucial role in preserving the accuracy of binarized models. Therefore, introducing multiple scaling experts incurs minimal memory overhead while effectively leveraging the advantages of the MoE strategy to enhance the capabilities of binarized models.

In the MoE framework, the number of experts selected corresponds directly to the number of layers processed. As a result, the typical MoE framework selects only one or two experts per inference stage to manage the increased processing burden associated with more experts being selected. On the other hand, the scaling factors of binarized LLMs are solely involved in linear operations with matrix multiplication, as detailed in Equation 2. This linearity allows for the efficient management of multiple scaling experts by linearly combining them before executing the matrix multiplication. Hence, instead of selecting only a few experts, as done in the conventional MoE framework, BinaryMoS dynamically generates instructions on how to combine these scaling experts based on the context. This approach overcomes the limitations of fixed expert choices in typical MoE setups by enabling the creation of effectively infinite token-adaptive scaling factors through linear combinations. Consequently, by optimally utilizing the representational power of multiple scaling experts, BinaryMoS maximizes the potential of binarized models while maintaining memory efficiency.

3.2 Router Design

In order to generate the token-adaptive scaling factors, the proposed BinaryMoS designs the router for processing the following operations:

$$G = \text{Softmax}(XW_R) \tag{3}$$

$$\hat{S}_{in} = GS_{in}, \quad \hat{S}_{out} = GS_{out} \tag{4}$$

Here, $W_R \in \mathbb{R}^{m \times e}$ represents the weight parameters of router's linear layer, where e denotes the number of experts. $S_{in} \in \mathbb{R}^{e \times m}$ and $S_{out} \in \mathbb{R}^{e \times n}$ denote the scaling experts for input and output dimension, respectively. Initially, the router computes the gating score G, which represents

https://doi.org/10.52202/079017-4367

Table 1: Comparison of memory requirements for deploying Float16 and binarized models, with the number in parentheses denoting the compression ratio of binarized models over Float16 models.

Model	Float16	PB-LLM	BiLLM	OneBit	BinaryMoS
LLaMA-1/2-7B	13.51 GB	2.78 GB (4.86×)	2.28 GB (5.93×)	1.37 GB (9.86×)	1.40 GB (9.65×)
LLaMA-1/2-13B	26.20 GB	5.02 GB (5.22×)	4.06 GB (6.45×)	2.29 GB (11.44×)	2.33 GB (11.24×)

the significance of each scaling expert, using input activations and router weights, as outlined in Equation 3. Notably, as the gating scores are generated with the softmax function, the sum of gating scores for the scaling experts equals 1. These scores are used to linearly combine the scaling experts, resulting in the creation of token-adaptive scaling factors $\hat{S}_{in}$ and $\hat{S}_{out}$, as shown in Equation 4. Then, by replacing the static scaling factors S_{in} and S_{out} from Equation 2 with token-adaptive scaling factors, the result of matrix multiplication $\hat{Y}$ in a linear layer using the BinaryMoS approach can be revised as follows:

$$\hat{Y} = [(X \odot \hat{S}_{in})\text{Sign}(W_{FP}^{T})] \odot \hat{S}_{out} \tag{5}$$

We empirically find that using four scaling experts each for the input and output dimensions provides the optimal compromise between increasing model size and improving accuracy. Consequently, the proposed BinaryMoS utilizes four scaling experts for each dimension to enhance accuracy while maintaining efficiency.

3.3 Impact of BinaryMos on LLM Compression

The proposed BinaryMoS introduces additional memory overhead due to multiple scaling experts and the weights of the router. However, this overhead is relatively minor. For instance, in the LLaMA-1/2-7B model [16] with a hidden dimension h of 4096, the weight matrix for the linear layers is typically 4096×4096. If BinaryMoS adopts 4 scaling experts, this translates to four α's, each of dimension 4096×1, for both input and output dimensions. Additionally, the weights of the router would be 4096×4. Compared to the previous OneBit method, which requires a single α for both input and output dimensions, the additional components in BinaryMoS total 4096×10 parameters. The number of these extra parameters constitutes only 0.2% of the original weight parameters.

For a comprehensive examination of the impact of various binarization techniques, including BinaryMoS, on LLM compression, we evaluate the memory requirements of LLaMA models with Float16 parameters and after applying different binarization methods, as detailed in Table 1. Following standard practice, all binarization techniques exclude the embedding layer and lm-head from binarization. Our analysis reveals that BinaryMoS significantly reduces the memory footprint of models, achieving compression ratios ranging from 9.65× to 11.24×. As model size increases, the relative impact of additional parameters diminishes and the proportion of the unbinarized part decreases. Hence, we can achieve higher compression ratios for larger models. For instance, the original LLaMA-1/2-13B model, requiring 26.20 GB for deployment, is impractical for edge devices due to its size. However, BinaryMoS reduces this model to just 2.33 GB, representing an 11.24-fold decrease in memory requirements. This significant reduction facilitates deployment on edge devices with typically limited memory capacities of 4 GB.

In contrast, PB-LLM and BiLLM methods achieve relatively lower compression ratios of around 5× and 6×, respectively. This is primarily due to two reasons: first, PB-LLM and BiLLM methods must retain salient weight information, increasing the average bitwidth of weight parameters. Second, the handling of sparse weight matrices in these methods introduces overhead in indexing sparse weight matrices, limiting the achievable compression ratio. OneBit achieves the highest compression ratio by only introducing dual-dimension scaling factors. Remarkably, BinaryMoS achieves a comparable compression ratio to OneBit, despite incorporating additional components for scaling experts. While the memory requirement of binarized models with BinaryMoS increases by only 2% compared to OneBit, the inclusion of scaling experts offers much greater potential to significantly improve perplexity.

This analysis demonstrates that although BinaryMoS introduces additional parameters, the relative increase in memory requirement is modest. This makes BinaryMoS a viable option for enhancing accurcy of binarized models without imposing a significant memory burden.

https://doi.org/10.52202/079017-4367

3.4 Quantization-Aware Knowledge Distillation

Following training strategies adopted for network compression [13, 14], we adopt the knowledge distillation (KD) to transfer the knowledge of a full-precision teacher model to a binarized student model. We employ the cross entropy (CE) loss to distill the logit knowledge. This is calculated using the following equation:

$$\mathcal{L}_{CE} = -\frac{1}{n} \sum_{c} \sum_{i=1}^{n} p_c^{\mathcal{T}} \left(X_i \right) \log \left(p_c^{\mathcal{S}} \left(X_i \right) \right) \tag{6}$$

Here, $\mathcal{S}$ and $\mathcal{T}$ represent the student and teacher models respectively. n denotes batch size, and c is the number of classes. Additionally, to minimize the distributional discrepancies in layer outputs, we incorporate a mean-squared error (MSE) based layer-to-layer (L2L) loss as follows:

$$\mathcal{L}_{L2L} = \sum_{l=1}^{L} \mathrm{MSE} \left(\mathbf{H}_l^{\mathcal{T}}, \mathbf{H}_l^{\mathcal{S}} \right) \tag{7}$$

In this loss, $\mathbf{H}_l^{\mathcal{T}}$ and $\mathbf{H}_l^{\mathcal{S}}$ are the output logits from the l-th layer of the teacher and student models, respectively. The total loss function, integrating both CE and L2L distillation losses, is defined as:

$$\mathcal{L} = \mathcal{L}_{CE} + \alpha \cdot \mathcal{L}_{L2L} \tag{8}$$

where α is a hyperparameter that balances the contributions of the CE and L2L losses. For the training of BinaryMoS, we empirically set $\alpha = 10$.

4 Experiments

4.1 Experimental Settings

Models and Evaluation Datasets. In our study, we evaluate BinaryMoS on various models, including those from the LLaMA-1 [16], LLaMA-2 [17], and OPT [15] families. Specifically, we utilize the OPT models with 125M and 1.3B parameters, and the LLaMA-1 and LLaMA-2 models with 7B and 13B parameters for our evaluations. We measure language modeling capabilities of these models by evaluating their perplexity on the WikiText2 [24] and C4 [25] datasets. Additionally, we assess zero-shot accuracy on various Common Sense Reasoning Tasks such as BoolQ [19], PIQA [20], HellaSwag [21], WinoGrande [22], ARC-e, ARC-c [23]), utilizing the open-source LLM evaluation framework, LM-Evaluation-Harness [35].

Training Details. We initialize the parameters of binarized models using those from pre-trained models, which serve as teacher models for KD. For the training dataset, a mixed dataset composed of the WikiText2 training dataset and a selected partition from the C4 training dataset, with a sequence length of 2048. The training is conducted over three epochs using the AdamW [18] optimizer, with hyperparameters set to $\beta_1 = 0.9$, $\beta_2 = 0.999$, and zero weight decay. We implement a cosine decay learning rate scheduler, preceded by a warm-up phase constituting 0.03 of the total training duration. All training sessions are conducted on NVIDIA A100 GPUs.

Baselines. We compare BinaryMoS against previous LLM binarization methods, including PB-LLM [5], BiLLM [6], and OneBit [7], ensuring that all implementations adhere to the details provided in their respective papers. PB-LLM and BiLLM utilize the Post-Training Quantization (PTQ) approach for model calibration through the Optimal Brain Quantizer (OBQ) based method of GPTQ [2]. For PB-LLM, which allows variable ratios of salient weights to enhance accuracy, we have set the ratio of salient weights to 10% to ensure the average bit width of weight parameters remains below 2 bits. OneBit employs a Quantization-Aware Training (QAT) approach, and for fairness, its training setup is aligned with that of BinaryMoS. Given the significant accuracy improvements demonstrated by BinaryMoS over traditional binarization techniques, we also include a comparison with 2-bit quantization methods with PTQ approach, such as GPTQ [2] and OmniQuant [28], to broaden the evaluation scope.

4.2 Analysis on the Number of Scaling Experts

To determine the optimal number of scaling experts for BinaryMoS, which effectively maintains the accuracy of binarized LLMs while minimizing memory usage, we conduct evaluations with

https://doi.org/10.52202/079017-4367

Table 2: The impact of the numbers of scaling experts on the proposed BinaryMoS. Quick assessment conducted using the LLaMA-1-7B model trained on one-third of the training data.

# of Experts	Perplexity ↓		Zero-shot Accuracy ↑						
	Wiki2	**C4**	BoolQ	PIQA	Hella.	WinoG.	ARC-e	ARC-c	**Average**
1	9.33	12.54	60.27	67.84	46.77	52.09	38.38	27.98	48.89
2	9.19	12.18	62.69	68.55	48.36	55.09	40.23	28.92	50.64
4	8.92	11.85	60.51	67.46	49.95	55.24	41.16	29.35	50.61
8	9.17	12.28	58.68	67.46	47.51	53.67	39.52	29.43	49.38

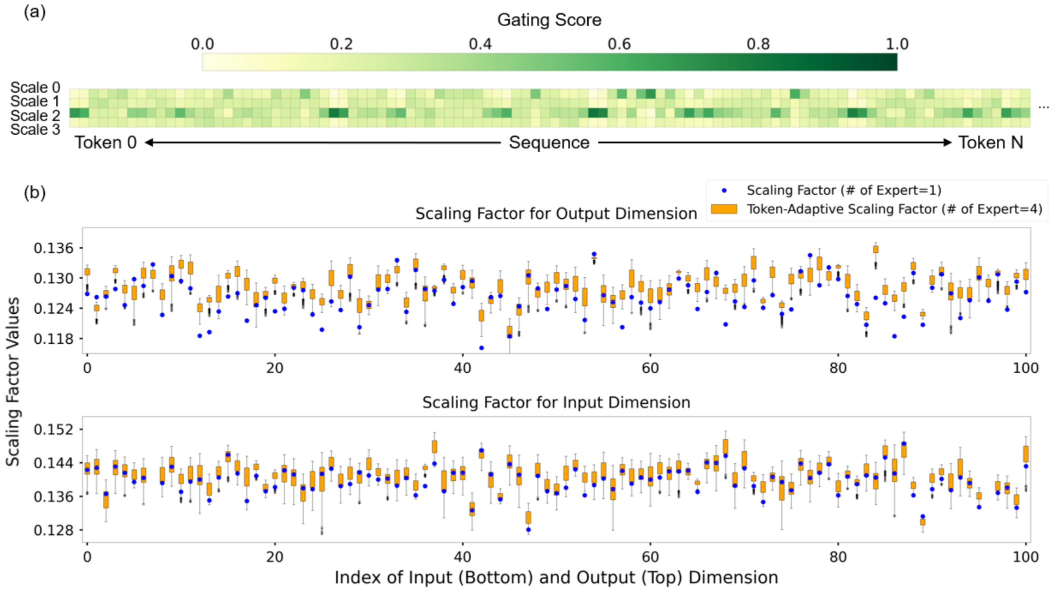


Figure 3: (**a**) Gating scores of 4 scaling experts in 18th layer of LLaMA-1-7B model for each token in the input sequence. (**b**) Distribution of values of token-adaptive scaling factors. The boxplot visually presents the distribution of token-adaptive scaling factors among processed tokens. The box spans the interquartile range, indicating the middle 50% of the scaling factors. Extending from the box are whiskers that reach the furthest data points within 1.5 times the interquartile range, providing insight into the overall range of the data.

LLaMA-1-7B using varying numbers of scaling experts. This evaluation is conducted using only one-third of the training data for quick assessment. As shown in Table 2, performance metrics, including perplexity and accuracy, generally improve as the number of experts increases from 1 to 4. However, a further increase to 8 experts leads to a decline in model performance. This decline arises from the challenge of training routers to appropriately assign scales to tokens as the number of scales increases. Based on these observations, we choose to employ 4 experts in the BinaryMoS approach.

4.3 Analysis on the Token-Adaptive Scaling Factors

In this section, we explore the effectiveness of the proposed mixture of scale approach in generating token-adaptive scaling factors. To accomplish this, we analyze the gating scores for scaling experts and the scaling factors derived from these scores. For this analysis, we utilize the LLaMA-1-7B model and input sequences sampled from the C4 dataset.

Figure 3 showcases the gating scores and resulting token-adaptive scaling factors for out projection of the 18th layer across tokens of the input sequence. The experimental results reveal substantial variation in the gating scores for each expert across tokens. As depicted in Figure 3(b), while conventional binarization methods with static scaling factors, akin to having a single expert, offer a fixed scaling factor, the scaling experts of BinaryMoS successfully generate a diverse range of scaling factors. This highlights the efficacy of the mixture of scale approach, which adaptively determines the

https://doi.org/10.52202/079017-4367

Table 3: Perplexity and zero-shot accuracy results of Float16 and binarized LLMs.

Model	**Method**	**Wbits**	**Perplexity ↓**		**Zero-shot Accuracy ↑**						
			Wiki2	**C4**	BoolQ	PIQA	Hella.	WinoG.	ARC-e	ARC-c	**Average**
OPT-125M	Float16	16	27.65	24.60	55.47	62.02	31.33	50.19	39.98	22.86	43.64
	PB-LLM	1	3233.63	1509.33	37.83	50.60	26.67	50.43	27.02	23.63	36.02
	BiLLM	1	2989.53	1769.26	37.82	50.59	25.75	51.30	27.65	23.63	36.12
	OneBit	1	39.45	35.58	61.92	60.01	27.01	50.43	35.81	21.84	42.84
	BinaryMoS	1	**36.46**	**33.13**	61.83	60.17	27.16	51.38	36.74	22.95	**43.37**
OPT-1.3B	Float16	16	14.62	14.72	57.82	72.42	53.70	59.51	50.97	29.52	53.99
	PB-LLM	1	272.83	175.42	62.17	54.24	27.25	50.27	27.98	23.72	40.94
	BiLLM	1	69.45	63.92	61.92	59.52	33.81	49.32	34.38	22.35	43.55
	OneBit	1	20.36	20.76	57.85	66.53	39.21	54.61	42.80	23.97	47.50
	BinaryMoS	1	**18.45**	**18.83**	60.34	68.66	41.99	53.99	44.87	26.19	**49.34**
LLaMA-1-7B	Float16	16	5.68	7.08	73.21	77.42	72.99	66.85	52.53	41.38	64.06
	PB-LLM	1	198.37	157.35	60.51	53.53	27.23	49.17	27.48	26.02	40.66
	BiLLM	1	41.66	48.15	62.23	58.65	34.64	51.14	33.08	25.68	44.24
	OneBit	1	8.48	10.49	62.50	70.40	54.03	55.32	41.07	30.88	52.36
	BinaryMoS	1	**7.97**	**9.72**	64.59	71.82	58.18	58.88	42.09	31.31	**54.48**
LLaMA-1-13B	Float16	16	5.09	6.61	68.47	79.05	76.24	70.17	59.85	44.54	66.39
	PB-LLM	1	35.83	39.79	62.17	58.70	33.97	52.17	31.86	23.63	43.75
	BiLLM	1	14.56	16.67	62.53	68.17	52.24	59.43	41.91	29.94	52.37
	OneBit	1	7.65	9.56	63.30	71.98	60.61	59.43	42.85	32.42	55.10
	BinaryMoS	1	**7.16**	**8.81**	63.82	73.88	64.05	60.93	44.28	33.11	**56.68**
LLaMA-2-7B	Float16	16	5.47	6.97	71.07	76.87	72.95	67.16	53.45	40.78	63.71
	PB-LLM	1	76.75	85.92	62.17	52.82	26.87	50.11	26.89	24.31	40.53
	BiLLM	1	27.72	36.34	62.14	59.19	35.18	53.11	34.22	26.54	45.06
	OneBit	1	8.60	10.74	63.06	70.40	54.24	56.67	40.82	29.35	52.42
	BinaryMoS	1	**7.88**	**9.75**	65.02	71.55	59.41	56.18	41.84	30.03	**54.01**
LLaMA-2-13B	Float16	16	4.88	6.47	68.99	79.05	76.62	69.77	57.95	44.20	66.10
	PB-LLM	1	155.25	151.15	37.82	53.26	28.89	49.48	28.28	23.72	36.91
	BiLLM	1	20.71	27.19	62.20	62.51	38.05	56.35	40.69	27.73	47.92
	OneBit	1	7.56	9.67	65.66	71.60	60.07	56.91	45.76	31.74	55.29
	BinaryMoS	1	**7.08**	**8.91**	66.12	73.72	63.80	58.98	45.71	33.19	**57.09**

scaling factor for each token, leading to a wider representation range. Consequently, we can expect that BinaryMoS effectively enhances the capacity of binarized models and improves model accuracy.

4.4 Perplexity and Accuracy Results of Binarized Models

The perplexity and zero-shot accuracy results of previous binarization methods and the proposed BinaryMoS are presented in Table 3. BinaryMoS consistently outperforms earlier binarization techniques across all metrics, effectively narrowing the performance disparity with their Float16 counterparts.

In particular, smaller LLMs such as OPT-125M and OPT-1.3B typically face challenges in maintaining linguistic capabilities under model compression. Previous methods like PB-LLM and BiLLM result in significant increases in perplexity, often exceeding 1000 for the OPT-125M model. While OneBit made substantial improvements, perplexity increases remained above 10. BinaryMoS, however, significantly enhances these outcomes by keeping the increase in perplexity below 10. Moreover, it boosts the accuracy of binarized models and diminishes the zero-shot accuracy gap to within 0.3% compared to Float16 models. The distinct advantage of BinaryMoS over previous approaches, especially OneBit, lies in its use of scaling experts. This evaluation underlines the efficacy of the BinaryMoS with mixture of scales approach.

Table 4: Perplexity and zero-shot accuracy results for 2-bit quantization methods and BinaryMoS.

		Perplexity ↓ (Wikitext2)					
Method	Wbits	OPT-125M	OPT-1.3B	LLaMA-1-7B	LLaMA-1-13B	LLaMA-2-7B	LLaMA-2-13B
GPTQ	2	660.52	125.29	45.73	15.20	40.23	32.87
OmniQuant	2	245.47	28.82	9.75	7.84	11.20	8.25
BinaryMoS	1	**36.46**	**18.45**	**7.97**	**7.16**	**7.88**	**7.08**
		Perplexity ↓ (C4)					
Method	Wbits	OPT-125M	OPT-1.3B	LLaMA-1-7B	LLaMA-1-13B	LLaMA-2-7B	LLaMA-2-13B
GPTQ	2	213.60	45.43	27.87	15.15	31.37	26.23
OmniQuant	2	390.30	33.81	13.01	10.43	15.46	11.06
BinaryMoS	1	**33.13**	**18.83**	**9.72**	**8.81**	**9.75**	**8.91**
		Average Zero-shot Accuracy ↑					
Method	Wbits	OPT-125M	OPT-1.3B	LLaMA-1-7B	LLaMA-1-13B	LLaMA-2-7B	LLaMA-2-13B
GPTQ	2	37.59	40.36	43.75	49.65	43.31	45.03
OmniQuant	2	36.54	46.43	51.58	56.42	49.54	54.24
BinaryMoS	1	**43.37**	**49.34**	**54.48**	**56.68**	**54.01**	**57.09**

4.5 Comparison between BinaryMoS and 2-bit Quantization

Since BinaryMoS consistently outperforms other binarization methods, we proceed to compare it with conventional 2-bit quantization techniques, GPTQ and OmniQuant. While these two approaches entail lower calibration overhead for quantization due to their use of the PTQ approach, they differ in their quantization methods. GPTQ and OmniQuant utilize a group-wise quantization approach, employing groups of 128 weights to finely quantize parameters and minimize quantization errors. Consequently, the memory demand during inference for these methods is more than double that of BinaryMoS. The comparison results, presented in Table 4, reveal that BinaryMoS even outperforms these 2-bit quantization methods, despite its lower memory requirement during inference. This once again underscores the effectiveness of integrating scaling experts.

5 Discussion and Future Work

BinaryMoS significantly improves the accuracy of binarized LLMs by increasing their representational capability with mixture of scales. This MoS approach holds promise for extension to multi-bit quantization, as multi-bit quantization techniques also involve scaling factors for regulating quantization step size. However, in this paper, our study does not delve into the effectiveness of the mixture of scales on multi-bit quanization schemes, leaving this avenue for future exploration.

Though BinaryMoS adopts the concept of MoE, it does not fully leverage advanced training techniques established in the field of MoE [26, 27, 34]. These advanced methods optimize routing functions and balance token assignments among experts, thereby enhancing MoE model accuracy. Thus, investigating these training techniques is another topic for future research.

6 Conclusion

This paper introduces BinaryMoS, a novel binarization technique designed to enhance the representation capability of binarized LLMs while preserving the fundamental advantage of binarization—low memory usage. BinaryMoS adopts the mixture of scale approach to dynamically adjust the scaling factors of binary weight values in a token-adaptive manner. Given that scaling factors play a crucial role in reducing binarization error and occupy a small portion of binarized models, this approach effectively mitigates information loss associated with binarization with minimal memory overhead. Our experimental findings demonstrate that BinaryMoS surpasses existing binarization approaches and even outperforms 2-bit quantization methods in both perplexity and zero-shot tasks.

Acknowledgements

This work was supported in part by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2021-0-00105: Development of model compression framework for scalable on-device AI computing on Edge applications , IITP-2023-RS-2023-00256081: artificial intelligence semiconductor support program to nurture the best talents, No. 2021-0-01343: Artificial Intelligence Graduate School Program (Seoul National University) , No.2021-0-02068: Artificial Intelligence Innovation Hub), in part by the Samsung Research Funding Center under Project SRFC-TC1603-53, and BK21 FOUR program at Seoul National University.

References

[1] Shiyao Li, Xuefei Ning, Luning Wang, Tengxuan Liu, Xiangsheng Shi, Shengen Yan, Guohao Dai, Huazhong Yang, Yu Wang, "Evaluating Quantized Large Language Models", *arXiv preprint arXiv:2402.18158*, 2024.

[2] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, Dan Alistarh, "GPTQ: Accurate Post-Training Quantization for Generative Pre-trained Transformers", *International Conference on Learning Representations (ICLR)*, 2023.

[3] Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, Song Han, "AWQ: Activation-aware Weight Quantization for LLM Compression and Acceleration", *arXiv preprint arXiv:2306.00978*, 2023.

[4] Changhun Lee, Jungyu Jin, Taesu Kim, Hyungjun Kim, Eunhyeok Park, "OWQ: Outlier-Aware Weight Quantization for Efficient Fine-Tuning and Inference of Large Language Models", *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2024.

[5] Yuzhang Shang, Zhihang Yuan, Qiang Wu, Zhen Dong, "PB-LLM: Partially Binarized Large Language Models", *International Conference on Learning Representations (ICLR)*, 2024.

[6] Wei Huang, Yangdong Liu, Haotong Qin, Ying Li, Shiming Zhang, Xianglong Liu, Michele Magno, Xiaojuan Qi, "BiLLM: Pushing the Limit of Post-Training Quantization for LLMs", *International Conference on Machine Learning (ICML)*, 2024.

[7] Yuzhuang Xu, Xu Han, Zonghan Yang, Shuo Wang, Qingfu Zhu, Zhiyuan Liu, Weidong Liu, Wanxiang Che, "OneBit: Towards Extremely Low-bit Large Language Models", *arXiv preprint arXiv:2402.11295*, 2024.

[8] Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang, " A survey on model compression for large language models", *arXiv preprint arXiv:2308.07633*, 2023.

[9] Haotong Qin, Ruihao Gong, Xianglong Liu, Xiao Bai, Jingkuan Song, and Nicu Sebe, "Binary neural networks: A survey", *arXiv preprint arXiv:2004.03333*, 2020.

[10] Zechun Liu, Wenhan Luo, Baoyuan Wu, Xin Yang, Wei Liu, and Kwang-Ting Cheng, "Bi-Real Net: Binarizing Deep Network Towards Real-Network Performance", *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[11] Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, Ali Farhadi, "XNOR-Net: ImageNet Classification Using Binary Convolutional Neural Networks", *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016.

[12] Zechun Liu, Zhiqiang Shen, Marios Savvides, Kwang-Ting Cheng, "ReActNet: Towards Precise Binary Neural Network with Generalized Activation Functions", *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[13] Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, Vikas Chandra, "LLM-QAT: Data-Free Quantization Aware Training for Large Language Models", *arXiv preprint arXiv:2305.17888*, 2023.

https://doi.org/10.52202/079017-4367

[14] Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, Qun Liu, “TernaryBERT: Distillation-aware Ultra-low Bit BERT”, *arXiv preprint arXiv:2009.12812*, 2020/

[15] Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, Luke Zettlemoyer, “OPT: Open Pre-trained Transformer Language Models”, *arXiv preprint arXiv:2205.01068*, 2022.

[16] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, Guillaume Lample, “LLaMA: Open and Efficient Foundation Language Models”, *arXiv preprint arXiv:2302.13971*, 2023.

[17] Touvron, Hugo and Martin, Louis and Stone, Kevin and Albert, Peter and Almahairi, Amjad and Babaei, Yasmine and Bashlykov, Nikolay and Batra, Soumya and Bhargava, Prajjwal and Bhosale, Shruti and others, “Llama 2: Open Foundation and Fine-Tuned Chat Models”, *arXiv preprint arXiv:2307.09288*, 2023.

[18] Ilya Loshchilov, Frank Hutter, “Decoupled Weight Decay Regularization”, *International Conference on Learning Representations (ICLR)*, 2019.

[19] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, Kristina Toutanova, “BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions”, *arXiv preprint arXiv:1905.10044*, 2019.

[20] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, Yejin Choi, “PIQA: Reasoning about Physical Commonsense in Natural Language”, *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[21] Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, Yejin Choi, “HellaSwag: Can a Machine Really Finish Your Sentence?”, *arXiv preprint arXiv:1905.07830*, 2019.

[22] Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, Yejin Choi, “WinoGrande: An Adversarial Winograd Schema Challenge at Scale”, *arXiv preprint arXiv:1907.10641*, 2019.

[23] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, Oyvind Tafjord, “Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge”, *arXiv preprint arXiv:1803.05457*, 2018.

[24] Stephen Merity, Caiming Xiong, James Bradbury, Richard Socher, “Pointer Sentinel Mixture Models”, *arXiv preprint arXiv:1609.07843*, 2016.

[25] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu, “Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer”, *arXiv preprint arXiv:1910.10683*, 2019.

[26] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, Jeff Dean, “Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer”, *arXiv preprint arXiv:1701.06538*, 2017.

[27] William Fedus, Barret Zoph, Noam Shazeer, “Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity”, *arXiv preprint arXiv:2101.03961*, 2021.

[28] Wenqi Shao, Mengzhao Chen, Zhaoyang Zhang, Peng Xu, Lirui Zhao, Zhiqian Li, Kaipeng Zhang, Peng Gao, Yu Qiao, Ping Luo, “OmniQuant: Omnidirectionally Calibrated Quantization for Large Language Models”, *International Conference on Learning Representations (ICLR)*, 2024.

[29] Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, Furu Wei, “BitNet: Scaling 1-bit Transformers for Large Language Models”, *arXiv preprint arXiv:2310.11453* , 2023.

[30] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, Luke Zettlemoyer, “QLoRA: Efficient Finetuning of Quantized LLMs”, *Advances in Neural Information Processing Systems, (NeurIPS)*, 2023.

[31] Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, Dan Alistarh, “SpQR: ASparse-Quantized Representation for Near-Lossless LLM Weight Compression”, *arXiv preprint arXiv:2306.03078*, 2023.

[32] Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W. Mahoney, Kurt Keutzer, “SqueezeLLM: Dense-and-Sparse Quantization”, *International Conference on Machine Learning (ICML)*, 2024.

[33] Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, Christopher De Sa, “QuIP: 2-Bit Quantization of Large Language Models With Guarantees”, *Advances in Neural Information Processing Systems, (NeurIPS)*, 2023.

[34] Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, William Fedus, “ST-MoE: Designing Stable and Transferable Sparse Expert Models”, *arXiv preprint arXiv:2202.08906*, 2022.

[35] Subhabrata Mukherjee, Xiaodong Liu, Guoqing Zheng, Saghar Hosseini, Hao Cheng, Greg Yang, Christopher Meek, Ahmed Hassan Awadallah, Jianfeng Gao, “Few-Shot Learning Evaluation in Natural Language Understanding”, *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2).*, 2021.

[36] Taesu Kim, Jongho Lee, Daehyun Ahn, Sarang Kim, Jiwoong Choi, Minkyu Kim, Hyungjun Kim, “QUICK: Quantization-aware Interleaving and Conflict-free Kernel for Efficient LLM Inference“, *arXiv preprint arXiv:2402.10076*, 2024.

A Appendix

A.1 Ablation Study on Datasets

To determine the optimal dataset for training binarized models, we conduct a comparative analysis using various training datasets, as summarized in Table 5. The results indicate that models trained solely on the WikiText2 dataset, due to its relatively small dataset size, tend to exhibit overfitting tendencies and struggle to generalize to other datasets. While these models demonstrate considerable perplexity improvement on the WikiText2 evaluation, their perplexity on the C4 dataset and zero-shot accuracy is notably poor. Conversely, models trained exclusively on the C4 dataset perform well across a wide range of tasks, except for the evaluation on WikiText2. Following the approach of previous research [13], we also experiment with a generated dataset synthesized using the LLaMA-1-7B model. Although this dataset generally performs satisfactorily across various language modeling tasks, its performance lags behind that of the C4 dataset. Therefore, to enhance overall model performance, we opt to train the models on a mixed dataset comprising both C4 and WikiText2. Moreover, the accessibility of both C4 and WikiText2 as open-source datasets further facilitates their adoption for training purposes.

Table 5: Evaluation of binarized LLaMA-1-7B model trained with various training datasets. We train the model on a subset of the dataset with the same training step. †: Generated dataset synthesized by LLaMA-1-7B model. ‡: Mixed dataset of Wikitext2 and C4.

Training	**Perplexity ↓**		**Zero-shot Accuracy ↑**						
Dataset	**Wiki2**	**C4**	**BoolQ**	**PIQA**	**Hella.**	**WinoG.**	**ARC-e**	**ARC-c**	**Avg**
Generated †	12.54	13.04	60.51	66.10	45.91	54.69	41.41	27.90	49.42
Wiki2	9.65	28.61	57.95	57.67	36.78	54.45	38.46	26.62	45.32
C4	13.76	11.97	60.33	67.79	49.69	53.74	39.26	29.52	50.06
Mixed ‡	8.92	11.85	60.51	67.46	49.95	55.24	41.16	29.35	50.61

A.2 Latency Measurement

To assess the latency of GEMV operation for our BinaryMoS, we have evaluated the latency of previous binarized models and the BinaryMoS by developing appropriate CUDA kernels for 1-bit matrix multiplication, modifying the CUDA kernel for multi-bit matrix multiplication [36]. Additionally, we further customize the CUDA kernel of BinaryMoS to fuse scaling experts and routing operations on top of the 1-bit matrix multiplication CUDA kernel. We measure the latency of the linear layers in LLaMA-7B and LLaMA-13B with batch size of 1 and results are presented in Table 6. All experiments are conducted on NVIDIA A6000 GPUs.

Previous methods like PB-LLM and BiLLM require extra matrix multiplications, making them very slow. OneBit, which employs the simplest binarization scheme, achieves significant improvement over the original Float16 model and shows the minimum latency. Meanwhile, our BinaryMoS introduces additional operations for processing scaling experts, which require far fewer operations compared to matrix multiplication. Consequently, BinaryMoS also shows similar latency results to OneBit. This demonstrates that the multi-scaling factor module in BinaryMoS improves performance in terms of perplexity and zero-shot accuracy with minimal overhead to latency.

Table 6: Latency (μsec) of linear layer in LLaMA-1/2-7B and LLaMA-1/2-13B.

Model Config	**LLaMA-1/2-7B**			**LLaMA-1/2-13B**		
Weight Size	**4096 × 4096**	**4096 × 11008**	**11008 × 4096**	**5120 × 5120**	**5120 × 13824**	**13824 × 5120**
Float16	68.2	151.7	143.5	95.6	224.1	213.6
PB-LLM	96.1	177.5	168.3	122.7	243.7	234.7
BiLLM	87.1	96.4	104.2	95.2	124.2	131.0
OneBit	32.7	33.7	34.9	33.4	41.4	42.6
BinaryMoS	34.5	36.9	37.0	35.6	43.4	44.5

A.3 Experimental Results for LLaMA-1-30B

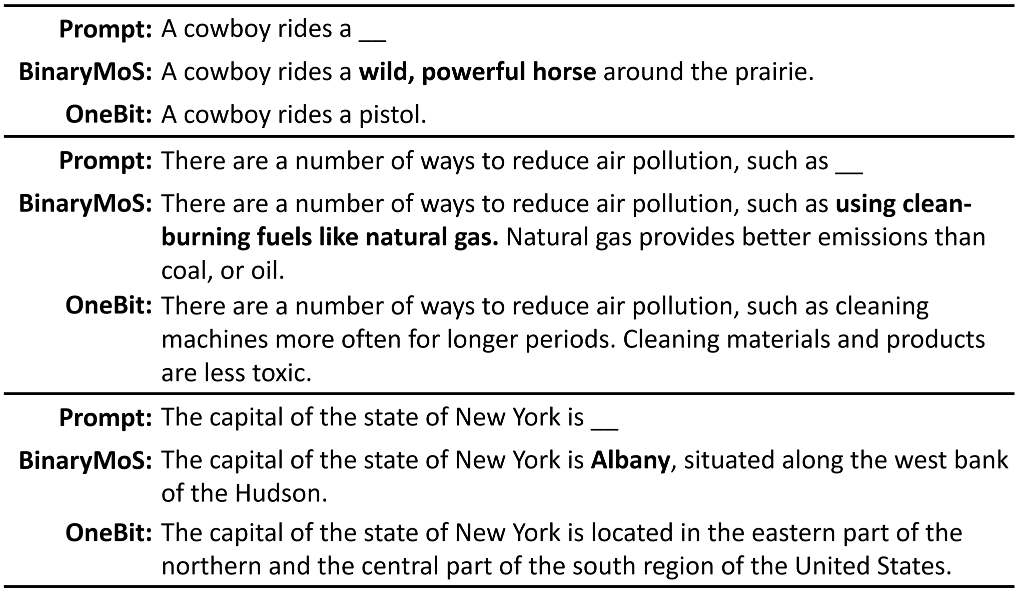
Table 7 provides further experimental results on the LLaMA-1-30B model. In line with the trends observed in Table 3, BinaryMoS consistently surpasses other binarization approaches for this 30B model. This assessment highlights the effectiveness of BinaryMoS for large-scale LLMs.

Table 7: Perplexity and zero-shot accuracy results of Float16 and binarized LLMs for LLaMA-1-30B

Model	Method	Wbits	Perplexity ↓		Zero-shot Accuracy ↑						
			Wiki2	C4	BoolQ	PIQA	Hella.	WinoG.	ARC-e	ARC-c	Average
LLaMA-1-30B	Float16	16	4.10	5.98	68.37	80.08	79.22	72.69	58.96	45.39	67.45
	PB-LLM	1	32.24	34.59	62.18	60.22	33.75	53.91	33.50	24.40	44.66
	BiLLM	1	10.10	12.28	62.39	71.38	59.49	66.54	44.95	33.19	56.32
	BinaryMoS	1	**6.63**	**8.36**	67.76	75.62	67.38	64.09	49.62	35.07	**59.92**

A.4 Generation Quality

Figure 4 compares the generation quality between BinaryMoS and OneBit on LLaMA-1-13B model. BinaryMoS can generate contextually proper answers, whereas OneBit fails to generate correct answers. These results demonstrate that the BinaryMoS processes each token with token-adaptive scaling factors which contain contextual information and the improvement of BinaryMoS over OneBit is substantial enough to make binarized LLMs more applicable in practice.



Prompt: A cowboy rides a __

BinaryMoS: A cowboy rides a **wild, powerful horse** around the prairie.

OneBit: A cowboy rides a pistol.

Prompt: There are a number of ways to reduce air pollution, such as __

BinaryMoS: There are a number of ways to reduce air pollution, such as **using clean-burning fuels like natural gas.** Natural gas provides better emissions than coal, or oil.

OneBit: There are a number of ways to reduce air pollution, such as cleaning machines more often for longer periods. Cleaning materials and products are less toxic.

Prompt: The capital of the state of New York is __

BinaryMoS: The capital of the state of New York is **Albany**, situated along the west bank of the Hudson.

OneBit: The capital of the state of New York is located in the eastern part of the northern and the central part of the south region of the United States.

Figure 4: Comparison of generation quality on the LLaMA-1-13B models with BinaryMoS and OneBit.

A.5 Limitations

While BinaryMoS effectively enhances the linguistic capabilities of binarized LLMs, its application to extremely large models such as LLaMA-2-70B poses challenging due to the high training cost inherent to QAT-based strategies. Hence, to facilitate the use of BinaryMoS in such large-scale models, it may be necessary to adapt it into a PTQ-based approach or to integrate it with parameter-efficient training strategies in future work. Additionally, it is important to note that, despite the advancements brought about by by BinaryMoS, the reduction in linguistic performance of binarized LLMs coampred to their Float16 counterparts remains substantial. Consequently, to make LLM binarization practical for real-world applications, further advancements in binarization techniques are required.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims in the abstract and introduction accurately reflect the paper's contributions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in the "Discussion and Future Work" and "Limitations" section, including challenges and areas for future exploration.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

https://doi.org/10.52202/079017-4367

Answer: [NA]

Justification: The paper does not include theoretical results that require formal proofs. The focus is on empirical evaluation of the proposed method.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides comprehensive details on the experimental settings, including models used, datasets, evaluation metrics, and training processes. It ensures that the steps to reproduce the results are clear.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

https://doi.org/10.52202/079017-4367

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The paper provide open access to the data and code to faithfully reproduce the main experimental results, as described in supplemental material.

   Guidelines:

   - The answer NA means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies all necessary training details, ensuring that the experimental results can be understood and replicated.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [No]

   Justification: The paper does not report error bars or statistical significance for the experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

https://doi.org/10.52202/079017-4367

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The paper provides sufficient information about the compute resources used for the experiments, mentioning the use of NVIDIA A100 GPUs.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms to the NeurIPS Code of Ethics, focusing on improving model efficiency and performance in a responsible manner without evident ethical concerns.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: The paper discusses societal impacts, highlighting the potential for efficient deployment of LLMs on edge devices and acknowledging the importance of responsible use.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not pose high risks for misuse that would require specific safeguards.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper properly credits existing assets, such as datasets and models, and follows appropriate licenses and terms of use.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

https://doi.org/10.52202/079017-4367

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not introduce new assets. It focuses on methodological improvements and uses existing datasets and models for evaluation.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve human subjects research, thus IRB approvals are not applicable.

    Guidelines:

https://doi.org/10.52202/079017-4367

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

https://doi.org/10.52202/079017-4367