DB-LLM: Accurate Dual-Binarization for Efficient LLMs

Hong Chen[1*], Chengtao Lv[1*], Liang Ding[2], Haotong Qin[3], Xiabin Zhou[5], Yifu Ding[1], Xuebo Liu[4], Min Zhang[4], Jinyang Guo[1], Xianglong Liu[1†], Dacheng Tao[6]

[1]Beihang University [2]The University of Sydney [3]ETH Zürich

[4]Harbin Institute of Technology, Shenzhen [5]Jiangsu University [6]Nanyang Technological University

{18373205, lvchengtao, xlliu}@buaa.edu.cn, haotong.qin@pbl.ee.ethz.ch, liangding.liam@gmail.com

Abstract

Large language models (LLMs) have significantly advanced the field of natural language processing, while the expensive memory and computation consumption impede their practical deployment. Quantization emerges as one of the most effective methods for improving the computational efficiency of LLMs. However, existing ultra-low-bit quantization always causes severe accuracy drops. In this paper, we empirically investigate the micro and macro characteristics of ultra-low bit quantization and present a novel **D**ual-**B**inarization method for **LLM**s, namely **DB-LLM**. For the micro-level, we take both the accuracy advantage of 2-bit-width and the efficiency advantage of binarization into account, introducing *Flexible Dual Binarization* (**FDB**). By splitting 2-bit quantized weights into two independent sets of binaries, FDB ensures the accuracy of representations and introduces flexibility, utilizing the efficient bitwise operations of binarization while retaining the inherent high sparsity of ultra-low bit quantization. For the macro-level, we find the distortion that exists in the prediction of LLM after quantization, which is specified as the deviations related to the ambiguity of samples. We propose the *Deviation-Aware Distillation* (**DAD**) method, enabling the model to focus differently on various samples. Comprehensive experiments show that our DB-LLM not only significantly surpasses the current State-of-The-Art (SoTA) in ultra-low bit quantization (*e.g.*, perplexity decreased from 9.64 to 7.23), but also achieves an additional 20% reduction in computational consumption compared to the SOTA method under the same bit-width. Our code is available at https://github.com/Hon-Chen/DB-LLM.

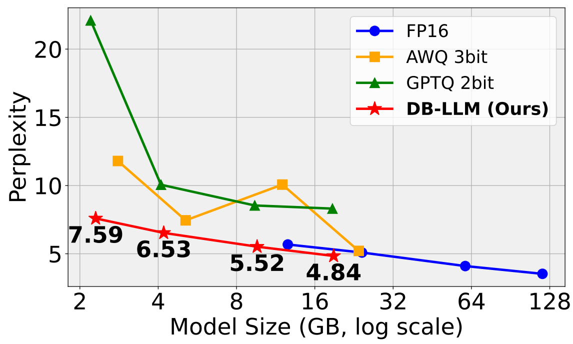


Figure 1: **The perplexity on WikiText2 for LLaMA family models.** 2-bit DB-LLM is close to FP results and surpasses 3-bit AWQ by a large margin.

1 Introduction

Recently, Large Language Models (LLMs), such as ChatGPT and LLaMA (Touvron et al., 2023a) have catalyzed a paradigm shift in various natural language processing tasks (Zhong et al., 2023; Peng et al., 2023; Lu et al., 2023). Their unprecedented capabilities evolved from a massive memory footprint (*e.g.*, billion-scale parameters), which constrains the widespread application of LLMs on resource-limited devices. Several compression schemes are thus proposed to reduce the memory demands of LLMs, which can be roughly categorized into weight quantization (Frantar et al., 2022; Lin et al., 2023), network pruning (Sun et al., 2023; Ma et al., 2023; He et al., 2022), knowledge distillation (Gu et al., 2023; Zhong et al., 2024) and low-rank factorization (Xu et al., 2023; Yuan et al., 2023). Among these methods, weight quantization is highly effective and practical since it achieves the best trade-off between the performance and the cost of the compression process. Nevertheless, although many works (Shao et al., 2023; Shang et al., 2023) attempt to quantize LLMs to ultra-low-bit (*e.g.*, 2-bit), their performance is unsatisfactory and falls far short of industrial application requirements.

Ultra-low-bit quantization ($\leq$ 4 bits), as an extremely efficient form of quantization, enjoys over $8\times$ memory compression ratio. Despite these specialized weight-only quantization schemes achiev-

*Equal contribution.
†Corresponding author.

Findings of the Association for Computational Linguistics ACL 2024, pages 8719–8730
August 11-16, 2024 ©2024 Association for Computational Linguistics

ing savings in storage consumption, they still cannot avoid costly floating-point arithmetic. Moreover, we notice that they will cause catastrophic degradation in accuracy.

For instance, despite the application of advanced 2-bit quantization techniques, a 65B model still falls marginally short of the performance level attained by a 7B model (Shao et al., 2023). And fully binarized Large Language Models are almost impracticable (Shang et al., 2023). The rationale lies in two important aspects: From the *micro-level* perspective: We empirically observe that the symmetric Gaussian distribution of pre-trained weights poses great challenges when quantizing to extremely low-bit (1-bit and 2-bit). Binarization suffers from poor representation capability, leading to a collapse in performance. While 2-bit quantization alleviates this issue to some extent, it still exhibits limited efficiency and presents optimization obstacles. Thus, directly applying the aforementioned strategies to LLMs is suboptimal which necessitates a novel specialized operator. From the *macro-level* perspective: We make in-depth investigations of the prediction preferences and discover the low-bit LLMs exhibit a form of distortion, far from the original long-tail distribution of the full-precision models. Especially, the extremely low-bit LLMs tend to potentially predict head classes when encountering ambiguous samples. This tendency highlights a potential bias in their performance.

To address these issues, we propose a novel **D**ual **B**inarization method to achieve accurate 2-bit **LLM**s in a data-free manner, dubbed as **DB-LLM**. Specifically, we (1) introduce a *Flexible Dual Binarization* (FDB) to enhance the representation capability by flexible dual-binarizer, while fully leveraging the efficiency benefits of the binarized parameter. Explicitly, we initialize an INT2 counterpart as the intermediary, splitting its weights into dual-binarized representations deftly in our DB-LLM. Then, in a data-free manner, we fine-tune the scales to further enhance the representation capability. Second, we (2) propose a *Deviation-Aware Distillation* (DAD) to mitigate the distorted preferences. DAD jointly leverages the student-teacher entropy as an ambiguous indicator and further amplifies the sample-wise ambiguity by re-weighting the distillation loss. This method enables the low-bit LLMs to perceive the uncertainty of each sample, which fulfills the balanced knowledge transfer.

Extensive experiments on several benchmark datasets and model families show that DB-LLM outperforms the existing state-of-the-art (SOTA) quantization methods by a convincing margin (see Figure 1). For example, our DB-LLM achieves perplexities of 5.52 and 4.84 under 2-bit weight on LLaMA-1-30B and LLaMA-1-65B respectively, comparable to full-precision LLaMA-1-7B (perplexity of 5.68) and even surpassing the 3-bit AWQ (Lin et al., 2023), which highlights its superiority and versatility. To summarize, our main **contributions** are:

- We present Flexible Dual Binarization, which transcends data format constraints, maximizing representation capability while maintaining the efficiency of binary operations.
- We analyze the distortion related to prediction preference in the ultra-low bit LLMs and introduce a Deviation-aware Distillation to emphasize the ambiguous samples.
- Extensive experiments on Llama1&2 families spanning 7$\sim$70B show that our DB-LLM significantly and consistently outperforms prior quantization strategies on various tasks.

2 Related Work

2.1 LLM Quantization

The quantization schemes of LLM can be briefly classified into two fields: weight-only quantization (Frantar et al., 2022; Lin et al., 2023; Chee et al., 2023) and weight-activation quantization (Wei et al., 2023; Xiao et al., 2023; Shao et al., 2023; Zhu et al., 2023). The first approach concentrates on reducing the model storage while the second one simultaneously accelerates the inference speed. For the weight-only quantization, GPTQ (Frantar et al., 2022) proposes a layer-wise quantization that compensates the rounding errors with second-order information. AWQ (Lin et al., 2023) prioritizes preserving the salient weights by the activation magnitude. QuIP (Chee et al., 2023) introduces quantization with incoherence processing, optimizing quantization in large language models but introducing additional overhead during inference. For the weight-activation quantization, several efforts (Xiao et al., 2023; Wei et al., 2023; Liu et al., 2023a; Shao et al., 2023) shift the challenge of outliers from activations to weights with per-channel scaling transformation, including optimization-free methods (Wei et al., 2023; Xiao et al., 2023) and optimization-based methods (Shao et al., 2023; Liu et al., 2023a). However, these works undergo non-trivial performance degradation in ultra-low-bit (*e.g.*, 2-bit). In contrast, our method achieves satisfactory accuracy.

2.2 Network Binarization

BNN (Hubara et al., 2016) is a radical quantization form to compress weights and activations into only 1 bit. Following the success of binarization in computer vision (Rastegari et al., 2016; Liu et al., 2018; Qin et al., 2020; Liu et al., 2020), its exploration in natural language processing also attracts wide research interest. BinaryBERT (Bai et al., 2021) equivalently splits the weights of well-trained TernaryBERT (Zhang et al., 2020) and further fine-tune it to enhance the performance. Subsequent works aim to binarize both weight and activations, which is more challenging. BiBERT (Qin et al., 2021) revisits the performance bottleneck (*i.e.*, softmax function) and proposes Bi-Attention to tackle information degradation. BIT (Liu et al., 2022) introduces a two-set binarization scheme, applying different mapping levels for non-negative and positive-negative activation layers. Most recently, PB-LLM (Shang et al., 2023) first attempts to bianrize the un-salient weights for LLM. Yet, such a mixed-precision manner limits its hardware deployment and extreme storage savings.

3 Methodologies

3.1 Preliminaries

In this section, we briefly review the necessary backgrounds. We consider the *quantization* and *binarization* as follows:

Uniform quantization is the most widely used method. For the k-bit setting, the quantization and de-quantization procedures can be written as:

$$w^q = \mathrm{clamp}(\lfloor \frac{w}{s} \rceil, -2^{k-1}, 2^{k-1} - 1), \quad (1)$$

$$\hat{w} = s \cdot w^q \approx w, \quad (2)$$

where W_q is the quantized integer and s is the scaling factor determined by $\frac{\max(|\mathbf{W}|)}{2^{k-1}}$. To overcome the non-differentiable issue in the backward propagation, the Straight-Through-Estimator (STE) (Courbariaux et al., 2015) is introduced to compute the approximate gradient.

The traditional BNNs binarize the network parameters (weights and activations) into 1-bit. The binarization on weights can be achieved by applying the sign function for the forward propagation:

$$w^b = \mathrm{sign}(w) = \begin{cases} 1 & \text{if } w \geq 0 \\ -1 & \text{otherwise} \end{cases}, \quad (3)$$

where w and w^b represent the 32-bit floating-point weight and 1-bit binarized weight.

3.2 Flexible Dual Binarization

These days, researchers discover the weights of LLMs exhibit symmetric Gaussian distribution and a small fraction of salient weights is critical to the quantization performance (Lin et al., 2023; Shao et al., 2023). We make in-depth investigations about the optimization from multi-low-bit perspectives (see Figure 4). The binarization suffers from poor representation capabilities. The remaining two levels converge towards 0 (shown in blue in Figure 3), which neglects the salient weights and is attributed to the highest loss values. Alternatively, 2-bit quantization naturally overcomes the representation bottleneck (expression span exceeds twice that of binarization in Figure 3). The minimum loss point is significantly reduced while the loss surface is still steep which brings the optimization difficulty.

To combine the notable efficiency inherent in binarization and the flexible representation capabilities of 2-bit quantization, we propose the *Flexible Dual Binarization* (FDB) whose loss landscape is flat and enjoys the lowest loss. Our FDB primarily consists of the initialization phase and the fine-tuning phase. It first inherits the considerably high-performing initialization from relatively high-bit LLMs and then fine-tunes the scales to further enhance the representation capability. In particular, we consider a 2-bit LLM as a proxy to be sufficient to tackle this obstacle (in Figure 3) thus we split its quantized weights into two separate 1-bit. We formulate such a splitting process as follows:

$$\hat{\boldsymbol{w}} = s \cdot \boldsymbol{w}^q = \alpha_1 \cdot \boldsymbol{w}_1^b + \alpha_2 \cdot \boldsymbol{w}_2^b, \quad (4)$$

where $\boldsymbol{w}_1^b, \boldsymbol{w}_2^b$ represent two 1-bit weights and α_1, α_2 are their corresponding scaling factors. To achieve the isometric step s between quantization levels in Equation 4 and maintain higher sparsity, we revisit the binarization levels and adjust it to $\{0, 1\}$. To illustrate, suppose that α_1 is positive and α_2 is negative in Figure 5. Thus the initial value of α_1, α_2 can be expressed as:

$$\alpha_1 := 2s, \alpha_2 := -s. \quad (5)$$

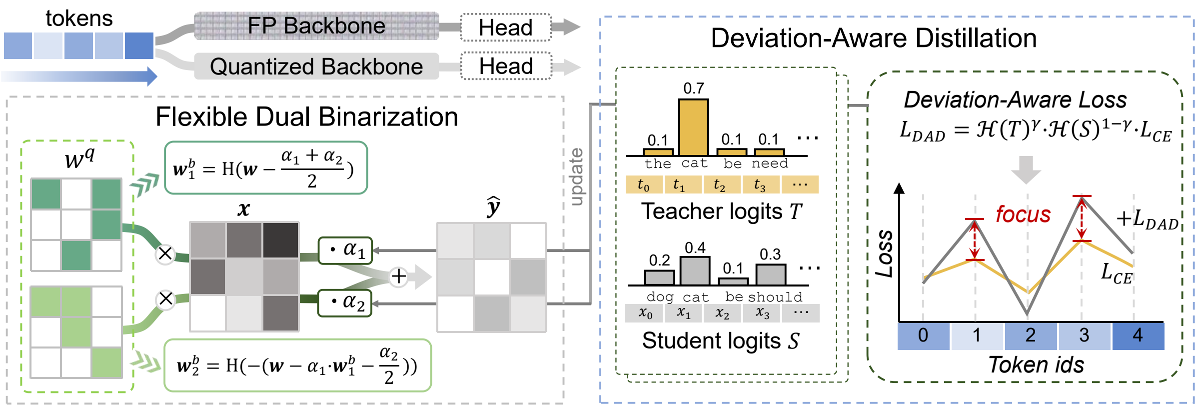


Figure 2: **Illustration of our proposed DB-LLM.** The *Flexible Dual Binarization* (**FDB**) approach, employing two independent 1-bit sparse weights for simultaneous matrix multiplication, significantly enhances the flexibility in weight representation. *Deviation-Aware Distillation* (**DAD**) steers the quantized model towards a heightened focus on ambiguous samples, enhancing its performance by refining quantization parameters.

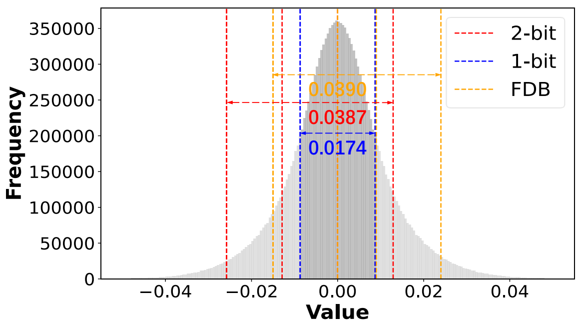


Figure 3: **Distributions of the first output projection's weight matrix** (LLaMA-1-7B). Colored levels, indicating the optimal solutions from grid search, minimize the proxy quantization error (MSE loss of outputs) for binarization, 2-bit quantization, and FDB. Influenced by the weight distribution's normality, binarization compresses the two levels closer to 0 due to the absence of a level representing 0, hindering the precise representation of numerous significant weights with higher values, whose expression span is less than half that of the 2-bit.

The quantization parameters, α_1 and α_2 will be optimized during the fine-tuning stage, which leads to the non-isometric quantization levels (in Figure 3). Therefore, our goal is to compare the magnitude between values and level center in Figure 5:

$$\boldsymbol{w}_1^b = \mathrm{H}(\boldsymbol{w} - \frac{\alpha_1 + \alpha_2}{2}), \tag{6}$$

$$\boldsymbol{w}_2^b = \mathrm{H}(-(\boldsymbol{w} - \alpha_1 \cdot \boldsymbol{w}_1^b - \frac{\alpha_2}{2})), \tag{7}$$

where $\mathrm{H}(\cdot)$ is the unit step function, defined as 0 for negative values and 1 for positive values. Therefore, the whole forward process of FDB is expressed as:

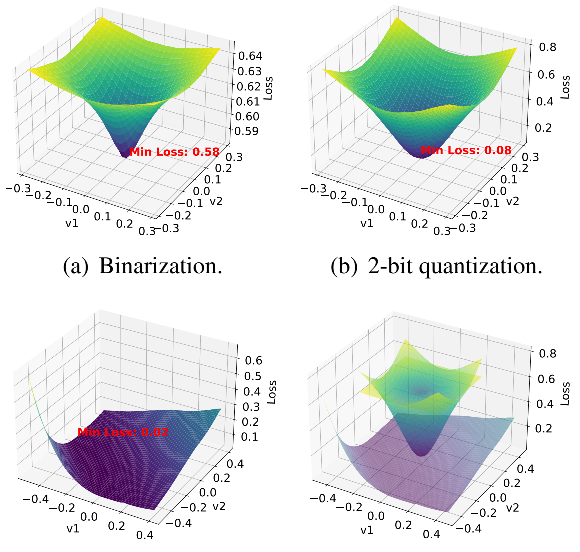
$$\hat{\boldsymbol{y}} = \alpha_1 \cdot (\boldsymbol{w}_1^b \otimes \boldsymbol{x}) + \alpha_2 \cdot (\boldsymbol{w}_2^b \otimes \boldsymbol{x}), \tag{8}$$

Where $\boldsymbol{x}$ and $\hat{\boldsymbol{y}}$ denote inputs and outputs of the current layer respectively, and $\otimes$ denotes the inner product with bitwise operation.

It is noteworthy that our elaborate Flexible Dual Binarization (FDB) enjoys multiple advantages: 1) it inherits and enhances the superior representation capacities of ultra-low bit quantization, 2) it capitalizes on the considerable efficiency derived from bitwise operation, 3) it maintains the notable high sparsity characteristic of ultra-low bit quantization.

Discussion on compression and acceleration. We have innovated the sparsity of neural network weights by decomposing traditional 2-bit weights into dual 1-bit representations. This method, applied in the LLaMA-1-7B model, significantly increases the average weight sparsity, exceeding 60%. Notably, there is a distinct variation in the degree of sparsity between $\boldsymbol{w}_1^b$ and $\boldsymbol{w}_2^b$, with the sparsity of $\boldsymbol{w}_2^b$ consistently surpassing 70%. This enhanced sparsity level is not only instrumental in drastically reducing the computational power requirements, potentially leading to significant acceleration in processing speed, but also facilitates more compression of $\boldsymbol{w}_2^b$ using various encoding methods (Van Leeuwen, 1976; Han et al., 2016). Theoretically, this approach could reduce the average bit size of the overall weights to approximately 1.88 bits (Shannon, 1948). These reductions, as previously mentioned, are significant when compared to traditional quantization methods, highlighting the superior efficiency of our approach.

Discussion on flexibility. As shown in Figure 4, we compare the layer-wise loss landscapes in bi-



(a) Binarization. (b) 2-bit quantization.

(c) FDB. (d) All Together.

Figure 4: **Loss landscape of a single quantized linear layer** based on binarization (a), 2-bit quantization (b), and our FDB (c). For (a), (b), and (c), we perturb the training parameters of the single layer and calculate the MSE loss, comparing the outputs of the quantized layer with those of the full-precision model. (d) highlights the disparity among the three surfaces by juxtaposing them within a single coordinate framework. The variables v1 and v2 represent perturbations applied to the training parameters along two orthogonal directions.

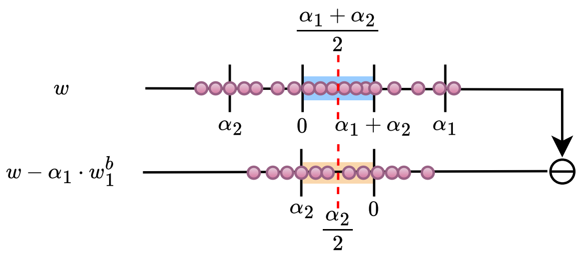
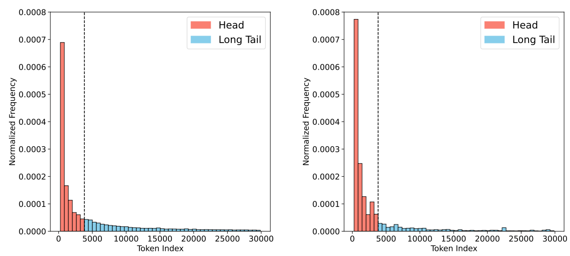


Figure 5: **The splitting procedure of FDB.** The dual separate 1-bit weight can be computed by comparing the central values.



(a) FP LLaMA-1-7B. (b) 2-bit LLaMA-1-7B.

Figure 6: **Frequency histograms depicting the distributions of prediction results** for the full-precision model (a) and extremely low-bit (2 bits) quantized model (b), gathered through random generation. The data is specifically presented for the [260,29870] interval, a range shaped by the construction of the BPE algorithm and connected to the long-tail distribution within the corpus.

narization, 2-bit quantization, and Flexible Dual Binarization (FDB). Our FDB achieves a minimum loss comparable to that of 2-bit quantization but significantly differs from binary quantization. FDB features a flatter optimization surface, which allows it to maintain a lower loss over a considerable range, reflecting its flexibility in net-wise optimization. Given that our FDB can be initialized through 2-bit quantization, the closeness of their lowest loss points also indicates that further optimization of FDB is likely to be easier.

Inspired by LLM-QAT (Liu et al., 2023b), we can further utilize distillation techniques to efficiently fine-tune the quantization parameters using the original full-precision model, without the need for introducing additional data. This data-free approach helps avoid the risk of overfitting.

3.3 Deviation-aware Distillation

The tokenizer construction of current mainstream Large Language Models is based on Byte Pair Encoding (BPE) (Gage, 1994; Sennrich et al., 2016), which leverages the long-tail corpus. Similarly, we observe that the prediction preference of full-precision LLMs obeys the long-tail distribution in Figure 6(a). However, the predictions of extremely low-bit models deviate from such long-tail distribution and exhibit increased distortion (in Figure 6(b)), which manifests as a bias towards high-frequency words. In particular, the quantized models are more inclined to predict head classes (*i.e.*, the higher frequency region of the vocabulary). Statistically, we count the prediction deviations given the same inputs, and the low-bit model is about 1.6 times more likely to predict commonly occurring head classes than the less frequent tail classes, indicating a bias towards more prevalent categories.

To delve deeper into the reason for distortion, we explore the failure predictions and utilize the information entropy (Shannon, 1948) to measure their corresponding uncertainty, defined by:

$$\mathcal{H}(\boldsymbol{P}) = -\sum_{i=1}^{C} p_i \log(p_i), \tag{9}$$

where C is the class number and p_i is the probabil-

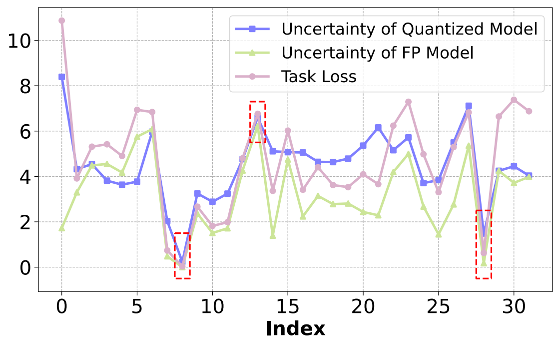


Figure 7: **The correlation between the uncertainty of model prediction results and task loss.** The uncertainties of the quantized and the original models are quantified as per Equation 9.

ity of i-th class. As shown in Figure 7, surprisingly, the entropy of the teacher/student model is consistent with the task loss (*i.e.*, cross-entropy). It means that the quantized model struggles with making predictions for ambiguous samples. Considering previous observations, it is reasonable to assume that the effectiveness of the quantized model decreases when dealing with ambiguous samples, leading to a preference for more conservative predictions.

Inspired by these findings, we propose the *Deviation-Aware Distillation* (DAD) which prioritizes uncertain samples by utilizing a pair of entropy (*i.e.*, teacher-student entropy) as a difficulty indicator. Specifically, the twin entropy is multiplied into the original loss function as two terms, which is defined as:

$$\ell_{DAD} = \mathcal{H}(P^t)^{\gamma} \cdot \mathcal{H}(P^s)^{1-\gamma} \cdot \ell_{CE}(P^t, P^s), \quad (10)$$

where superscript t and s denote teacher and student models respectively. $\ell_{CE}(P^t, P^s)$ is the cross-entropy loss between quantized student logits P^s and teacher logits P^t. The overall distill loss is:

$$\ell_{total} = \lambda \cdot \ell_{DAD} + \ell_{CE}, \quad (11)$$

where λ is the trade-off parameter.

Eventually, we analyze the issue of head class convergence in ultra-low-bit student models and propose the DAD loss to address it. DAD utilizes the teacher-student entropy as a challenge indicator and pays sufficient attention to the ambiguous samples by reweighting the distillation loss, which promotes the more balanced transfer of knowledge from full-precision teacher models.

4 Experiments

Models and datasets We conduct extensive experiments on LLaMA-1 (Touvron et al., 2023a) and LLaMA-2 (Touvron et al., 2023b) families. To evaluate the effectiveness of our DB-LLM, we measure the perplexity for the language generation tasks (*i.e.*, WikiText2 (Merity et al., 2016) and C4 (Raffel et al., 2020), and accuracy for the zero-shot tasks (*i.e.*, PIQA (Bisk et al., 2020), ARC (Clark et al., 2018), HellaSwag (Zellers et al., 2019) and WinoGrande (Sakaguchi et al., 2021).

Baselines We mainly compare DB-LLM with the state-of-the-art weight-only quantization methods, including RTN (round-to-nearest quantization), GPTQ (Frantar et al., 2022), AWQ (Lin et al., 2023), QuIP (Chee et al., 2023), OmniQuant (Shao et al., 2023) and the partially binarized strategy PB-LLM (Shang et al., 2023). To unify the model weights to a 2-bit representation, we set the ratio of salient weights (8-bit representation) in the PB-LLM to $\frac{1}{7}$ ($\frac{1}{7} \times 8 + \frac{6}{7} \times 1 = 2$bits).

Implementations Following LLM-QAT (Liu et al., 2023b), we construct the data-free calibration set which comprises 20k samples. Note that the quantization parameters are optimized for only 1 epoch with a batch size of 2. The γ and λ in Deviation-Aware Distillation is set to 0.1 equally. We adopt the AdamW (Loshchilov and Hutter, 2018) as an optimizer and the learning rate is set to $1e^{-5}$.

4.1 Main results

We conduct extensive experiments on LLaMA families across different model sizes (7B$\sim$70B) and evaluation tasks (the detailed results of LLaMA-2 can be found in Appendix A.2). Note that we focus on the performance of extremely low-bit settings (*i.e.*, W2A16).

For the language generation tasks, as seen in Table 1 and Table 2, some previous methods, such as AWQ, suffer from non-trivial performance degradation (Perplexity at the level of e^5). Fortunately, our DB-LLM consistently achieves lower perplexity for all the datasets. For instance, DB-LLM averages a 1.68 improvement in perplexity over OmniQuant on LLaMA-1-7B. When the model becomes larger, DB-LLM still obtains approximately 0.80 reduction in perplexity, which showcases the effectiveness and versatility. Notably, we found that our scheme even surpasses the RTN, and AWQ under W3A16, which further indicates the strong performance of DB-LLM. To the best of our knowledge, our 2-bit LLaMA-1-30B outperforms the full-precision LLaMA-1-7B with 3.7× storage savings.

#Bits	Method	LLaMA-1-7B		LLaMA-1-13B		LLaMA-1-30B		LLaMA-1-65B	
		WikiText2	C4	WikiText2	C4	WikiText2	C4	WikiText2	C4
W16A16	-	5.68	7.08	5.09	6.61	4.10	5.98	3.53	5.62
W2A16†	RTN	188.32	151.43	101.87	76.00	19.20	30.07	9.39	11.34
W3A16	RTN	25.73	28.26	11.39	13.22	14.95	28.66	10.68	12.79
W2A16†	AWQ	2.5e5	2.8e5	2.7e5	2.2e5	2.3e5	2.3e5	7.4e4	7.4e4
W3A16	AWQ	11.88	13.26	7.45	9.13	10.07	12.67	5.21	7.11
W2A16†	GPTQ	22.10	17.71	10.06	11.70	8.54	9.92	8.31	10.07
W2A16†	QuIP	13.19	24.84	8.60	13.23	7.18	10.57	5.98	8.55
W2A16†	OmniQuant	8.91	11.79	7.35	9.75	6.60	8.66	5.65	7.60
W2A16†	PB-LLM	20.61	47.09	10.73	25.40	9.65	16.28	6.50	11.13
W2A16†	DB-LLM	**7.59**	**9.74**	**6.35**	**8.42**	**5.52**	**7.46**	**4.84**	**6.83**

Table 1: **Performance comparisons of different methods for weight-only quantization on LLaMA-1** for language generation tasks. † represents the group size is 64.

#Bits	Method	2-7B	2-13B	2-70B
W16A16	-	5.47	4.88	3.31
W2A16†	RTN	431.97	26.22	10.31
W3A16	RTN	539.48	10.68	7.52
W2A16†	AWQ	2.1e5	1.2e5	-
W3A16	AWQ	24.00	10.45	-
W2A16†	GPTQ	20.85	22.44	NAN
W2A16†	OmniQuant	9.64	7.55	6.11
W2A16†	PB-LLM	20.37	43.38	NAN
W2A16†	DB-LLM	**7.23**	**6.19**	**4.64**

Table 2: **Weight-only quantization method comparisons** on LLaMA-2 with WikiText2 perplexity results.

Method	WikiText2	C4	Ppl Avg.	Acc Avg.
W16A16	5.68	7.08	6.38	62.22
2-bit Baseline	18.32	30.42	24.37	40.14
Baseline + Fine-tuning	8.42	10.10	9.26	53.45
Baseline + FDB	7.77	9.84	8.81	54.09
Baseline + FDB + DAD	**7.59**	**9.74**	**8.67**	**54.44**

Table 3: **Effect of DAD and FDB components**.

γ	0	0.1	0.3	0.5	0.7	0.9	1.0
WikiText2	7.61	**7.59**	7.62	7.71	7.86	8.00	8.09

Table 4: **Ablation study** of key hyper-parameter γ.

Moreover, our method is also demonstrated advantages in zero-shot tasks in Table 5. DB-LLM still outperforms other state-of-the-art strategies by a large margin. For instance, our approach improves the accuracy of LLaMA-1-7B by 6.39% and 5.45% on HellaSwag and Winogrande, respectively. Meanwhile, for LLaMA-1-65B, DB-LLM is close to FP results (less than 4% accuracy degradation).

4.2 Ablation Studies

To better understand the effectiveness of our method, we provide detailed ablation studies to show the effect of each component and the proposed deviation-aware loss.

Ablation for each component: Table 3 shows the effect of each component. When removing the DAD component, the perplexity slightly increases by about 0.1%-0.2% since the quantized model struggles to predict ambiguous samples. Furthermore, the FDB component is critical as the performance decreases significantly (7.77 to 18.32) without a fine-tuning procedure. Our well-designed FDB flexibly enhances the representation capability and promotes efficient computation.

Ablation for Deviation-Aware Loss: To investigate the impact of key hyper-parameter γ, we conduct the ablation experiments in Table 4. We find that simply introducing the student entropy ($\gamma = 0$) or teacher entropy ($\gamma = 1$) adversely affects the performance, and the teacher model is more convincing. Hence, by validation, we set γ to 0.1 in all our experiments, which is a sweet spot that unites the teacher-student entropy to guide the quantized model.

Ablation for dataset sizes: We conduct experiments on different dataset sizes and compare the training overhead with OmniQuant in Table 7. For a 7B model, OmniQuant (Shao et al., 2023) trained for 20 epochs using 128 samples, which, according to the paper, takes about 1.1 hours on a single A100 GPU. Our method only requires training for a single epoch. By reducing the amount of data

Model	#Bits	Method	Accuracy (%) ↑					
			PIQA	ARC-e	ARC-c	HellaSwag	Winogrande	Avg.
LLaMA-1-7B	W16A16	-	77.37	52.53	41.38	72.99	66.85	62.22
	W2A16	GPTQ	59.36	32.11	25.09	35.14	49.01	40.14
	W2A16	AWQ	50.05	25.76	29.44	25.93	49.96	36.23
	W2A16	QuIP	62.57	38.26	28.33	43.41	53.99	45.31
	W2A16	OmniQuant	68.66	44.49	29.69	54.32	55.56	50.54
	W2A16	PB-LLM	55.39	34.22	24.23	31.99	52.88	39.74
	W2A16	DB-LLM	**72.14**	**44.70**	**33.62**	**60.71**	**61.01**	**54.44**
LLaMA-1-13B	W16A16	-	79.05	59.85	44.62	76.22	70.09	65.97
	W2A16	GPTQ	71.44	49.58	36.01	63.34	62.43	56.56
	W2A16	AWQ	50.76	27.19	28.92	26.29	47.91	36.21
	W2A16	QuIP	69.97	40.40	31.06	54.60	56.91	50.59
	W2A16	OmniQuant	73.01	49.54	33.70	62.10	61.96	56.06
	W2A16	PB-LLM	62.89	40.99	28.33	40.77	58.09	46.21
	W2A16	DB-LLM	**74.16**	**51.18**	**37.54**	**68.29**	**64.72**	**59.18**
LLaMA-1-30B	W16A16	-	80.09	58.92	45.39	79.21	72.77	67.28
	W2A16	GPTQ	72.91	49.49	36.69	66.89	65.27	58.25
	W2A16	AWQ	48.91	26.22	29.44	25.91	47.12	35.52
	W2A16	QuIP	70.67	44.95	33.96	61.39	61.17	54.43
	W2A16	OmniQuant	75.57	52.06	38.48	68.34	65.11	59.91
	W2A16	PB-LLM	66.87	43.06	30.97	50.47	62.75	50.82
	W2A16	DB-LLM	**77.58**	**52.57**	**40.53**	**72.75**	**69.46**	**62.58**
LLaMA-1-65B	W16A16	-	80.85	58.75	46.25	80.73	77.11	68.73
	W2A16	GPTQ	77.58	52.61	40.19	72.05	71.82	62.85
	W2A16	QuIP	74.92	50.25	39.08	68.44	65.98	59.73
	W2A16	OmniQuant	78.51	52.65	40.36	72.37	68.82	62.54
	W2A16	PB-LLM	74.16	52.15	37.80	65.00	71.27	60.08
	W2A16	DB-LLM	**79.87**	**53.66**	**42.58**	**76.13**	**71.82**	**64.81**

Table 5: **Performance comparisons of different methods for weight-only quantization** for zero-shot tasks.

Method	Model Size	Sparsity	FLOPs
FP-16	12.6G	-	423.4G
3-bit quantization	2.8G	-	88.2G
2-bit quantization	2.2G	48.3%	37.3G
binarization	1.4G	0%*	36.4G
Ours	2.3G	62.8%	29.8G

Table 6: **Model size, sparsity, and computational complexity** of LLaMA-1-7B with different compression methods, where the model processes a 32-token sentence. *Binarization does not map the weights to 0, we treat its sparsity as 0.

used by our DB-LLM, such as adopting a dataset of only 2.5K in size, it only takes approximately 1.1 GPU hours as well. However, our approach still outperforms OmniQuant, with a perplexity reduction of about 0.4 and 1.6% accuracy improvement on downstream tasks. Furthermore, our DB-LLM's accuracy continues to improve as the size of the dataset increases.

Method	Dataset Size	Ppl Avg.	Acc Avg.	GPU hours
FP	-	6.38	62.22	-
OmniQuant	128	10.35	50.54	1.1
DB-LLM	2.5k	**9.92**	**52.18**	1.1
	5k	9.54	52.75	2.3
	10k	9.12	53.37	4.4
	20k	**8.67**	**54.44**	**8.2**

Table 7: **Effect of different dataset sizes.**

Ablation for group sizes: We add experimental results for our DB-LLM under different group sizes (64, 128) in Table 8, which are widely used. As

Method	Group Size	WikiText2	C4	Ppl Avg.	Acc Avg.
W16A16	-	5.68	7.08	6.38	62.22
DB-LLM	64	7.59	9.74	8.67	54.44
DB-LLM	128	8.63	10.81	9.72	51.76
OmniQuant	64	8.90	11.79	10.35	50.54

Table 8: **Effect of different group sizes.**

the group size increases, we observe a decline in model performance. Notably, our DB-LLM at a group size of 128 surpasses OmniQuant at a group size of 64 (8.63 vs 8.90 on WikiText2 and 10.81 vs 11.79 on C4).

4.3 Storage Saving and Speedup

As shown in Table 6, we specifically calculate the model size, sparsity, and computational complexity of several compression methods of the LLaMA-1-7B model. In terms of model size, we have introduced a negligible amount of quantization parameters. However, as analyzed in the previous Section 3.2, higher sparsity can lead to a lower average number of bits per weight. This suggests that there is further potential for model size reduction. Despite this, the overall model size is almost identical to that of 2-bit quantization. More importantly, our model exhibits significantly higher sparsity, which substantially reduces the computational complexity during model inference. We measure computational complexity by the number of floating-point operations (FLOPs) required for a single inference (Sun et al., 2023; Ma et al., 2023; Liu et al., 2018). The FLOPs decreases from 423.4 billion to 29.8 billion, indicating a reduction of approximately 14.2 times.

1-bit matrix operations lack some underlying support, but our method is also a weight-only quantization method, which can benefit from deploying packed weights without introducing additional operations, compared to QuIP (Chee et al., 2023).

5 Conclusion

In this paper, we present DB-LLM, an accurate Dual-Binarization approach for efficient Large Language Models (LLMs). Through a detailed analysis of extremely low-bit quantization and binarization, we've outlined the advantages and disadvantages of each method. Capitalizing on these insights, we meticulously develop the Flexible Dual Binarization to represent weights efficiently. This method transcends the constraints imposed by data formats. Additionally, we examine the macro-level prediction deviations in low-bit quantization and introduce Deviation-Aware Distillation, which directs the model to focus more on ambiguous samples. Our experiments show that our method surpasses the current state-of-the-art (SOTA) in 2-bit quantization accuracy and also greatly reduces computational demands compared to traditional techniques.

Limitations

While our DB-LLM demonstrates considerable advancements in ultra-low bit quantization, there are still avenues for further exploration and improvement. Firstly, the potential of full binarization for even more extreme bit-width compression presents an area that warrants additional investigation. This approach could further reduce computational demands but needs careful consideration to maintain model accuracy. Secondly, our current methodology primarily focuses on weight quantization, leaving the quantization of activation and scale values as a promising area for future research. Delving into these aspects could yield additional gains in efficiency and model performance, making LLMs even more accessible for practical applications.

Ethics Statements

We take ethical considerations seriously and strictly adhere to the ACL Ethics Policy. This paper proposes a flexible dual binarization algorithm and a deviation-aware distillation method to improve the computational efficiency of LLMs. All employed models and datasets in this paper are publicly available and have been widely adopted by researchers. All experimental results upon these open models and datasets are reported accurately and objectively. Thus, we believe that this research will not pose any ethical issues.

Acknowledgments

We are grateful to the anonymous reviewers and the area chair for their insightful comments and suggestions. This work was supported by the National Science and Technology Major Project (2021ZD0110503), the National Natural Science Foundation of China (No. 62306025, No. 92367204), and Beijing Municipal Science and Technology Project (Nos. Z231100010323002). Xuebo Liu was sponsored by CCF-Tencent Rhino-Bird Open Research Fund.

References

Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jin Jin, Xin Jiang, Qun Liu, Michael Lyu, and Irwin King. 2021. Binarybert: Pushing the limit of bert quantization. In *ACL*.

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

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

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

Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. 2015. Binaryconnect: Training deep neural networks with binary weights during propagations. *NeurIPS*.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint*.

Philip Gage. 1994. A new algorithm for data compression. *C Users J.*, page 23¨C38.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Knowledge distillation of large language models. *arXiv preprint*.

Song Han, Huizi Mao, and William J. Dally. 2016. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. In *ICLR*.

Shwai He, Liang Ding, Daize Dong, Jeremy Zhang, and Dacheng Tao. 2022. SparseAdapter: An easy approach for improving the parameter-efficiency of adapters. In *Findings of EMNLP*.

Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio. 2016. Binarized neural networks. *NeurIPS*.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. 2023. Awq: Activation-aware weight quantization for llm compression and acceleration. *arXiv preprint*.

Jing Liu, Ruihao Gong, Xiuying Wei, Zhiwei Dong, Jianfei Cai, and Bohan Zhuang. 2023a. Qllm: Accurate and efficient low-bitwidth quantization for large language models. *arXiv preprint*.

Zechun Liu, Barlas Oguz, Aasish Pappu, Lin Xiao, Scott Yih, Meng Li, Raghuraman Krishnamoorthi, and Yashar Mehdad. 2022. Bit: Robustly binarized multi-distilled transformer. *NeurIPS*.

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

Zechun Liu, Zhiqiang Shen, Marios Savvides, and Kwang-Ting Cheng. 2020. Reactnet: Towards precise binary neural network with generalized activation functions. In *ECCV*.

Zechun Liu, Baoyuan Wu, Wenhan Luo, Xin Yang, Wei Liu, and Kwang-Ting Cheng. 2018. Bi-real net: Enhancing the performance of 1-bit cnns with improved representational capability and advanced training algorithm. In *ECCV*.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *ICLR*.

Qingyu Lu, Baopu Qiu, Liang Ding, Kanjian Zhang, Tom Kocmi, and Dacheng Tao. 2023. Error analysis prompting enables human-like translation evaluation in large language models: A case study on chatgpt. *arXiv preprint*.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *arXiv preprint*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. In *ICLR*.

Keqin Peng, Liang Ding, Qihuang Zhong, Li Shen, Xuebo Liu, Min Zhang, Yuanxin Ouyang, and Dacheng Tao. 2023. Towards making the most of chatgpt for machine translation. In *Findings of EMNLP*.

Haotong Qin, Yifu Ding, Mingyuan Zhang, YAN Qinghua, Aishan Liu, Qingqing Dang, Ziwei Liu, and Xianglong Liu. 2021. Bibert: Accurate fully binarized bert. In *ICLR*.

Haotong Qin, Ruihao Gong, Xianglong Liu, Mingzhu Shen, Ziran Wei, Fengwei Yu, and Jingkuan Song. 2020. Forward and backward information retention for accurate binary neural networks. In *CVPR*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *JMLR*.

Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. 2016. Xnor-net: Imagenet classification using binary convolutional neural networks. In *ECCV*.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2021. Winogrande: An adversarial winograd schema challenge at scale. *ACM*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *ACL*.

Yuzhang Shang, Zhihang Yuan, Qiang Wu, and Zhen Dong. 2023. Pb-llm: Partially binarized large language models. *arXiv preprint*.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *IEEE*.

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*.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2023. A simple and effective pruning approach for large language models. *arXiv preprint*.

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

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

Jan Van Leeuwen. 1976. On the construction of huffman trees. In *ICALP*.

Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. 2023. Outlier suppression+: Accurate quantization of large language models by equivalent and effective shifting and scaling. In *ACL*.

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

Mingxue Xu, Yao Lei Xu, and Danilo P Mandic. 2023. Tensorgpt: Efficient compression of the embedding layer in llms based on the tensor-train decomposition. *arXiv preprint*.

Zhihang Yuan, Yuzhang Shang, Yue Song, Qiang Wu, Yan Yan, and Guangyu Sun. 2023. Asvd: Activation-aware singular value decomposition for compressing large language models. *arXiv preprint*.

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

Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, and Qun Liu. 2020. Ternarybert: Distillation-aware ultra-low bit bert. In *EMNLP*.

Qihuang Zhong, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2023. Can chatgpt understand too? a comparative study on chatgpt and fine-tuned bert. *arXiv preprint*.

Qihuang Zhong, Liang Ding, Li Shen, Juhua Liu, Bo Du, and Dacheng Tao. 2024. Revisiting knowledge distillation for autoregressive language models. *arXiv preprint*.

Miaoxi Zhu, Qihuang Zhong, Li Shen, Liang Ding, Juhua Liu, Bo Du, and Dacheng Tao. 2023. Zero-shot sharpness-aware quantization for pre-trained language models. In *EMNLP*.

A Example Appendix

A.1 Details about FDB

In the initialization phase, we use the GPTQ (Frantar et al., 2022) to quickly obtain the 2-bit quantized weights and integrate the zero points of asymmetric quantization into the scales. The positive and negative values of the scales correspond to two different zero points.

A.2 More experimental results

See Table 9 for more results. The Results show a similar trend to that of LLaMA-1 in Table 1, demonstrating the effectiveness and universality of our proposed method.

Model	#Bits	Method	PPL ↓		Accuracy (%) ↑				
			WikiText2	C4	PIQA	ARC-e	ARC-c	HellaSwag	Winogrande
LLaMA-2-7B	W16A16	-	5.47	6.97	76.99	53.58	40.53	72.96	67.25
	W2A16	AWQ	2.06e5	1.54e5	50.00	26.52	26.79	26.14	49.64
	W2A16	OmniQuant	9.64	12.73	68.72	39.77	30.89	53.44	56.12
	W2A16	PB-LLM	20.37	44.88	55.22	29.88	22.01	30.49	50.36
	W2A16	DB-LLM	**7.23**	**9.62**	**73.18**	**45.20**	**33.53**	**61.98**	**61.72**
LLaMA-2-13B	W16A16	-	4.88	6.47	79.05	57.95	44.28	76.62	69.61
	W2A16	AWQ	1.25e5	9.74e4	50.49	26.73	29.61	25.74	51.07
	W2A16	OmniQuant	7.55	10.05	71.06	47.69	34.73	61.15	57.77
	W2A16	PB-LLM	43.38	68.59	55.01	31.27	23.12	30.23	52.33
	W2A16	DB-LLM	**6.19**	**8.38**	**75.14**	**51.64**	**38.14**	**68.04**	**64.09**
LLaMA-2-70B	W16A16	-	3.32	5.52	80.85	59.72	47.95	80.85	76.95
	W2A16	OmniQuant	6.11	7.89	76.28	55.18	41.04	71.74	67.09
	W2A16	DB-LLM	**4.64**	**6.77**	**79.27**	**55.93**	**44.45**	**76.16**	**73.32**

Table 9: **Performance comparisons of different methods on LLaMA-2** model family.