AN EMPIRICAL STUDY OF EXTREMELY LOW-BIT QUANTIZATION FOR LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent research on LLM quantization has predominantly focused on post-training quantization (PTQ). While effective at higher bit-widths, PTQ still suffers from severe performance degradation in extremely low-bit settings (*e.g.*, 2-bit), limiting its applicability in resource-constrained environments. In contrast, quantization-aware training (QAT) offers a promising solution to recover the accuracy loss introduced by quantization. However, due to its substantial demands on training data and computational resources, QAT remains largely underexplored for LLMs. In this work, we present a comprehensive empirical study of QAT for extremely low-bit quantized LLMs. We investigate critical factors affecting QAT effectiveness, including quantizer design, quantization granularity, initialization strategies, training data selection, and training hyperparameters. Based on these insights, we propose a general QAT recipe and validate it on LLaMA3 models, achieving state-of-the-art performance under extremely low-bit settings. All code and training details will be released to facilitate reproducibility and foster future research on QAT for LLMs.

1 INTRODUCTION

Large Language Models (LLMs), built upon large-scale transformer architectures(Vaswani et al., 2017), have demonstrated strong generalization ability across diverse tasks, including text generation(Liang et al., 2024), information extraction(Xu et al., 2023), machine translation(Zhu et al., 2024), text classification(Sun et al., 2023), and so on. Despite their impressive performance, the massive parameter size and computational demands result in high latency and energy consumption during inference, which makes deployment challenging, particularly on resource-constrained edge and mobile devices(Zhu et al., 2023).

These challenges highlight the critical need for efficient compression methodologies for LLMs. Quantization, which converts computationally expensive floating-point parameters into low-bit precision, has emerged as an effective technique for reducing memory footprint, computational latency, and energy consumption during inference, thereby facilitating deployment on resource-constrained devices. Extremely low-bit quantization (e.g., 2 bits) could achieve substantial model compression rate, further reducing inference costs and enabling deployment on resource-constrained platforms. However, such aggressive compression often incurs severe accuracy degradation, making the exploration of extremely low-bit quantization methods crucial.

Existing quantization methods can be broadly categorized into post-training quantization (PTQ) and quantization-aware training (QAT). PTQ methods, such as GPTQ(Frantar et al., 2022), AWQ(Lin et al., 2024), and Omniquant(Shao et al., 2023), calibrate a pre-trained full-precision model using a small amount of calibration data, without retraining, and are therefore highly efficient and resource-friendly. However, PTQ methods generally struggle to maintain accuracy under extremely low-bit quantization. In contrast, the QAT approaches utilize training data to to retrain the quantized model, helping to recover accuracy. Among them, some QAT methods(Ma et al., 2024; 2025; Chen et al., 2025) explore training extremely low-bit models from scratch, but they fail to fully leverage the knowledge embedded in pre-trained full-precision models, resulting in higher resource requirements to obtain a well-optimized model. Other approaches(Liu et al., 2025a; Team, 2025) suggest that inheriting knowledge from pre-trained full-precision models improves performance and reduces training cost, highlighting the importance of developing strategies to transfer this knowledge to extremely low-bit quantized models. Despite the recent growing interest in QAT within the research community, a

systematic exploration of training strategies for extremely low-bit quantization remains limited. Key aspects of QAT, such as quantizer design, quantization granularity, initialization strategies, training strategies, and data selection, are still underexplored within a unified framework, leaving substantial room to narrow the performance gap between full-precision and quantized models.

In this work, we build upon a simple quantization baseline to systematically investigate key factors affecting extremely low-bit quantization performance. We begin by revisiting the design of quantizers. Our analysis shows that, using a floating-point offset yields lower quantization error compared to an integer zero-point, and learning the step size is more efficient than learning the upper and lower clipping bounds. Incorporating additional Stretched Elastic levels significantly enhances the representational capacity of the quantizer. Based on these observations, we propose a enhanced LSQ+ quantizer, ELSQ+, which not only exhibits stable training across different learning rates but also achieves substantially faster loss convergence. In addition, we study the effects of quantization granularity and initialization, finding that group-wise quantization consistently outperforms channel-wise quantization. Finer granularities such as group size 64 lead to substantial accuracy gains, while coarser granularities like group sizes 128 and 256 achieve similar performance. Advanced initialization provides clear benefits for coarse-grained quantization, but offers little to no improvement over the naive MinMax method for fine-grained quantization. Furthermore, we investigate the impact of training data on QAT. By simply selecting data that exhibit moderate alignment with the full-precision model, namely the middle-PPL subset, we can effectively improve the accuracy of the quantized model. Finally, we examine the impact of key training hyperparameters, including learning rate, scheduler, weight decay, and training budget, on quantization-aware training, and derive an effective and efficient configuration.

Based on these insights, we design a general and cost-effective solution for extremely low-bit quantization-aware training, termed NashQuant, serving as a stronger baseline for future studies. We validate our proposed NashQuant on model sizes suitable for mobile deployment, such as 1B and 3B. The results show that our solution achieves substantial performance improvements over existing quantization baselines. We strongly believe that extremely low-bit quantization is a key trend for the future development of LLMs, with quantization-aware training being an essential enabling technique. As more optimization techniques are introduced, the accuracy gap between full-precision and quantized models can be further reduced.

2 Design Space of QAT for LLMs

In this section, we systematically investigate several key factors that affect the performance of extremely low-bit quantization for LLMs, including the design of the quantizer, the choice of quantization granularity and initialization, the selection of training data, and the configuration of training hyperparameters. Through rigorous analysis and empirical comparisons, we derive several insights that inform the design of strong baseline. Our analysis primarily focuses on the 2-bit setting.

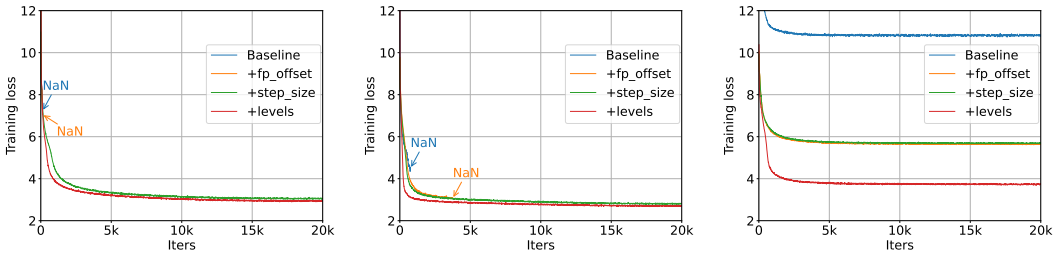
2.1 Preliminaries

Quantization is the process of compressing high-precision tensors into low-precision representations, inherently introducing irreversible information loss. In this work, we primarily focus on extremely low-bit quantization of model weights, specifically the 2-bit settings, which enables substantial compression to reduce storage overhead, resulting in significant inference speedups. Following OmniQuant (Shao et al., 2023) and BitDistiller (Du et al., 2024), given a floating-point weight W, we quantize it to a b-bit tensor W^q with the upper clipping value u and the lower clipping value l.

$$\begin{aligned} W^q &= \mathrm{Clip}(\mathrm{Round}(W/s) + z, 0, 2^b - 1)), \\ W^{deq} &= (W_q - z) \cdot s, \end{aligned} \tag{1}$$

where

$$\begin{aligned} s &= (u - l)/(2^b - 1), \\ z &= \mathrm{Clip}(-\mathrm{Round}(l/s), 0, 2^b - 1). \end{aligned}$$



(a) Learning Rate 1e-3. (b) Learning Rate 1e-4. (c) Learning Rate 1e-5.

Figure 1: Comparison of different quantizers under 2-bit weight quantization. The baseline is the quantizer defined in Equation 1. `+fp_offset`, `+step_size`, and `+levels` denote the three techniques in our Enhanced LSQ+. The experiments are conducted incrementally, adding each technique on top of the previous one, with the final setting corresponding to the full Enhanced LSQ+ quantizer. To further validate effectiveness, all techniques are evaluated under three learning rates.

Here, $\text{Clip}(W, \alpha, \beta) = \text{Min}(\text{Max}(W, \alpha), \beta)$, Round outputs the nearest integer value around the input. Typically, the clipping value u and l are set as learnable parameters, and during training, the corresponding quantization step size and zero-point are computed accordingly. To address the non-differentiability of the rounding operation, we typically employ the Straight-Through Estimator (STE) (Courbariaux et al., 2015) to approximate its gradients.

2.2 Quantizer Design

First, we examine the weight quantizer defined in Equation 1, commonly employed in LLMs. We refer to the resolution as the fineness of its discretization, namely the number of distinct representable values determined by the bit-width. We observe that the resolution of extremely low-bit quantization is highly constrained—for instance, 2-bit allows only four discrete values, corresponding to three levels. Consequently, several design choices that work well at higher bit-widths become suboptimal in this regime. To mitigate this issue, we propose three modifications to enhance the performance of quantizers in extremely low-bit settings.

Floating Offset vs. Integer Zero Point: Typically, the learned clipping values in Equation 1 maps the floating-point range onto the full b-bit integer range, with l mapped to 0 and u to $2^b - 1$. However, we observe that using an offset in the integer domain (i.e. zero-point z) can lead to distortion in the de-quantization of learnable clipping values. As shown in Equation 2, quantizing the clipping values introduces significant rounding errors, with a maximum magnitude of $0.5/(2^b - 1)$ of the floating-point range. In extremely low-bit settings, this error can be substantial. For instance, in 2-bit quantization, the maximum rounding error can reach 16.7% of the full floating-point range.

$$W_{\text{min}}^{deq} = \text{Round}(l/s) \cdot s, \quad W_{\text{max}}^{deq} = \text{Round}(u/s) \cdot s. \tag{2}$$

Although rounding errors are common in the quantization process, clipping values primarily serve to compute the scale. To preserve accuracy during QAT, it is crucial to ensure that updates to these clipping values are as precise as possible. To address this issue, we propose replacing the integer-domain offset z with a floating-point offset β, effectively mitigating the quantization error. The formulation is presented as follows.

$$\begin{aligned} W^q &= \text{Round}(\text{Clip}(\frac{W - \beta}{s}, 0, 2^b - 1)), \\ W^{deq} &= W_q \cdot s + \beta, \end{aligned} \tag{3}$$

where offset $\beta = l$ and the step size s is determined according to the formulation in Equation 1. With this design, the clipping values l and u can be recovered losslessly after quantization and dequantization, thereby enabling more efficient learning of the optimal clipping range.

Step Size vs. Clipping Value: Although clipping values (l, u) and the step size s are algebraically interchangeable via $s = (u - l)/(2^b - 1)$, their gradient behaviors differ substantially in practice. During QAT, the forward computation flows from the clipping values u and l to the derived step size

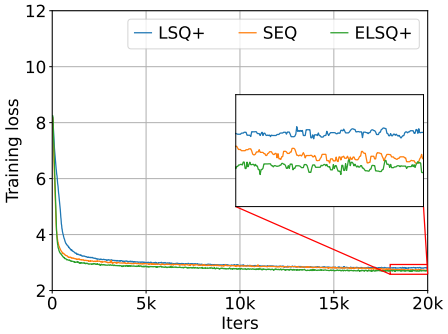


Figure 2: Training loss of four quantizers.

Quantizer	Symmetric	Levels	Clipping/Step
LSQ (Esser et al., 2019)	✓	$2^b - 1$	Step size
LSQ+ (Bhalgat et al., 2020)	✗	$2^b - 1$	Step size
SEQ (Liu et al., 2025a)	✓	2^b	Clipping Value
ELSQ+ (Ours)	✗	2^b	**Step size**

Table 1: Comparison on key quantization properties

s. In the backward pass, however, the gradient of the loss $\mathcal{L}$ is first accumulated with respect to s and only then propagated to u and l through $\partial s/\partial u = 1/(2^b - 1)$ and $\partial s/\partial l = -1/(2^b - 1)$ as shown in Equation 4. The scaling factor $1/(2^b - 1)$ in this chain drastically attenuates the gradients of u and l, leading to significantly smaller update steps. This is particularly problematic because the absolute values of u and l are typically much larger than the step size s in the forward pass, yet their gradients are disproportionately suppressed in the backward pass. Such imbalance renders the optimization of clipping values inefficient and misaligned with their role in the quantization process.

$$\frac{\partial \mathcal{L}}{\partial u} = \frac{\partial \mathcal{L}}{\partial s} \cdot \frac{1}{2^b - 1}, \quad \frac{\partial \mathcal{L}}{\partial l} = \frac{\partial \mathcal{L}}{\partial s} \cdot \Big(- \frac{1}{2^b - 1} \Big). \tag{4}$$

Although gradient scaling techniques (Esser et al., 2019) can be applied to compensate for the suppressed gradients of u and l, we note that large language models are typically trained with the AdamW optimizer (Loshchilov & Hutter, 2017), which normalizes gradients and thereby diminishes the effect of such scaling. This observation naturally motivates directly optimizing the step size s rather than the clipping values, providing a more effective and stable approach for QAT of LLMs, particularly under extremely low-bit settings.

Stretched Elastic vs. Vanilla Elastic: We further study the representational capacity of the quantizer. In prior approaches, which we refer to as *vanilla elastic*, b-bit quantization is typically assumed to produce 2^b discrete nodes, with $2^b - 1$ quantization levels between them. Recent work, ParetoQ (Liu et al., 2025a), introduces a quantizer based on *Stretched Elastic Quantization* (SEQ). In 2-bit quantization, this approach maps to the discrete nodes $\{-1.5, -0.5, 0.5, 1.5\}$, allocating two levels to positive values and two levels to negative values. Consequently, the total number of quantization intervals increases from $2^b - 1$ to 2^b, effectively enhancing the representational capacity with extra level. To this end, building upon the two techniques introduced above, we further incorporate the *Stretched Elastic* approach to extend the quantization levels, resulting in the *Enhanced LSQ+* quantizer, as formulated in the following equation.

$$\begin{aligned} W^q &= \text{Round}(\text{Clip}(\frac{W - \beta}{s}, -2^{b-1} + \epsilon, 2^{b-1} - \epsilon) - 0.5) + 0.5, \\ W^{deq} &= W_q \cdot s + \beta, \end{aligned} \tag{5}$$

where ϵ denotes a very small positive constant, typically set to 0.01. ELSQ+ can be regarded as a rethinking and extension of existing approaches for extremely low-bit quantization, incorporating three key techniques: floating-point offsets, learnable step size, and adding stretched elastic level.

To validate the effectiveness of ELSQ+, we conduct incremental ablation experiments across three different learning rates, as shown in Figure 1. When training with a high learning rate of $1e-4$, the baseline quickly diverges, while a low learning rate of $1e-5$ leads to high-loss collapse. Using a floating-point offset stabilizes training, and replacing clipping values with a learnable step size improves performance, especially at high learning rates. Adding extra levels with stretched elastic quantization further reduces loss across all learning rates, showing that each strategy consistently enhances ELSQ+ performance. In addition, we conduct a comprehensive comparison with LSQ (Esser et al., 2019), LSQ+ (Bhalgat et al., 2020), and SEQ (Liu et al., 2025a) quantizers, as shown in Table 1. ELSQ+ differs from these methods in that it adopts an asymmetric design, learns the step size, and further incorporates stretched elastic quantization, making it particularly well-suited for extremely

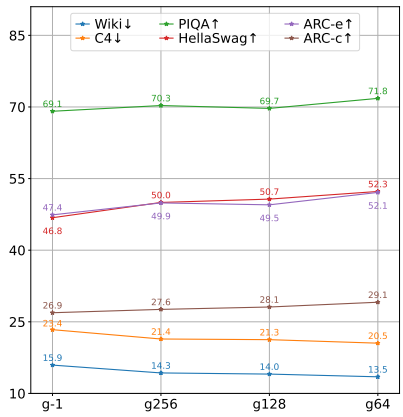


Figure 3: Quantization granularity.

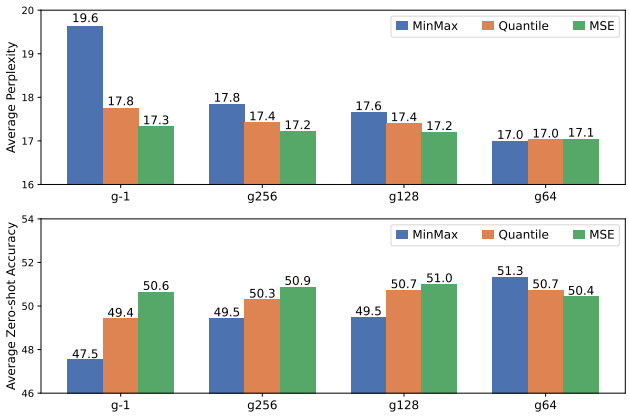


Figure 4: Quantization Initialization.

low-bit settings. The results, summarized in Figure 2, show that ELSQ+ consistently achieves significantly lower loss than LSQ+ and SEQ, underscoring its advantage in the low-bit regime.

2.3 Quantization Granularity and Initialization

Quantization Granularity. In general, finer quantization granularity leads to better accuracy. This observation is particularly salient in LLMs, where parameter distributions often exhibit a substantial number of outliers. In coarse-grained quantization schemes, for example channel-wise quantization, an entire set of parameters within a large scope is constrained to share a single quantization interval. However, the presence of outliers in LLM parameters forces the quantization levels to expand in order to cover a much larger floating-point range under a fixed bit-width. Consequently, even significantly different parameters are mapped to the same quantization nodes, leading to severe information loss and accuracy degradation. By contrast, fine-grained quantization more faithfully approximates parameter distributions. By locally adjusting step sizes and offsets, it captures statistical variations and isolates outliers within smaller subgroups. This prevents diverse parameters from collapsing into the same quantization level, alleviating the representational limits of extremely low-bit quantization. It is worth noting that the effectiveness of fine-grained quantization has been extensively validated in PTQ (Shao et al., 2023). However, its efficacy in QAT, especially under extremely low-bit regimes, remains largely unexplored.

Our work extends the study of quantization granularity to this more challenging scenario, demonstrating its practical benefits in training 2-bit LLMs. we conduct experiments comparing channel-wise quantization with group-wise quantization under group sizes of $\{64, 128, 256\}$, employing the Enhanced LSQ+ quantizer. As shown in Figure 3, all group-wise quantization strategies outperform the channel-wise baseline in both perplexity and zero-shot accuracy, confirming that finer granularity mitigates outlier effects and improves downstream generalization. Performance at group sizes 128 and 256 is similar, whereas group size 64 yields substantially better results. At these larger group sizes, the granularity remains too coarse to effectively separate outliers, resulting in comparable quantization errors. In contrast, group size 64 provides finer isolation of outliers, reducing quantization errors and enabling more accurate parameter representation, which translates into improved model performance across all evaluation metrics.

Quantization Initialization. Previous studies have shown that better initialization can lead to significant improvements in the performance of quantized models, as exemplified by LSQ(Esser et al., 2019), which introduced mean-squared-error (MSE)–based initialization. In contrast, researches on QAT for LLMs have paid little attention to this factor, with simple MinMax initialization remaining the prevalent choice(Liu et al., 2023; Chen et al., 2024; Liu et al., 2025a). However, under extremely low-bit quantization, the representational range of quantized models is severely compressed, which can exacerbate the impact of initialization on both training stability and model accuracy. Furthermore, under the same bit-width, how quantization granularity interacts with initialization remains largely unexplored. To investigate this, we systematically evaluate multiple common initialization schemes, including MinMax, Quantile and MSE initialization, across different quantization granularity.

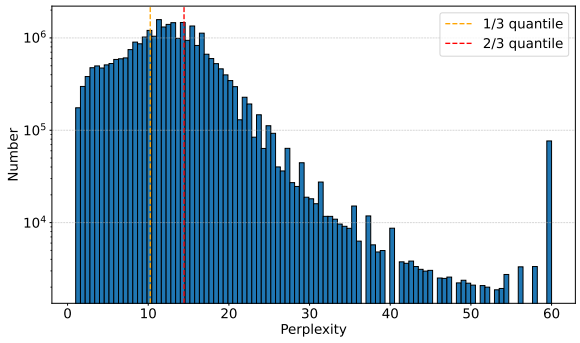


Figure 5: Distribution of perplexity in SlimPajama.

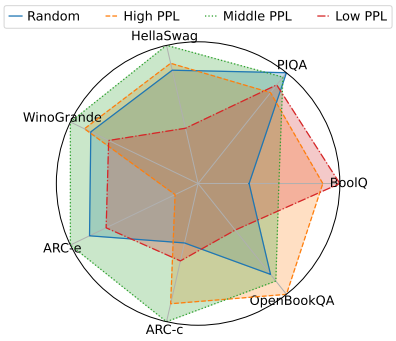


Figure 6: Zero-shot accuracy comparison.

As shown in Figure4, we find that MSE achieves best performance in perplexity and zero-shot under channel-wise quantization. And, as the quantization granularity becomes finer, the difference between the best and worst results across quantization initialization methods decreases significantly. As shown in Figure 4, MSE achieves the best performance in both perplexity and zero-shot accuracy under channel-wise quantization. Moreover, as quantization granularity becomes finer, the gap between the best and worst initialization methods decreases substantially. For instance, with channel-wise quantization, the difference in PPL between MSE (best) and MinMax (worst) is 2.3 points, with an average zero-shot performance gap of 3.1 points. In contrast, using group-wise quantization with a group size of 64 reduces the PPL difference to just 0.1 points.

2.4 Training Data Selection

Data quality is widely acknowledged as a decisive factor in the success of LLM training, and its role has been thoroughly examined in the full-precision models. However, in LLM quantization, this aspect has received relatively little attention. Most existing methods focus on PTQ, where only a small calibration set is required, often consisting of 128 samples randomly drawn from datasets such as Wikipedia (Merity et al., 2016) or C4 (Raffel et al., 2020). Since the model weights remain fixed in PTQ, the impact of data quality on the final performance is minimal, leading to the perception that data is not a decisive factor for quantization. In contrast, quantization-aware training (QAT) requires updating the model weights, making data quality substantially more critical. Modern LLMs are typically pretrained on trillions of tokens from large-scale general-purpose corpora, followed by fine-tuning with hundreds of billions of carefully curated, high-quality data. Unfortunately, such high-quality datasets used in the final training stages are often not publicly available.

In this section, we study the impact of data selection on QAT using open-source datasets. Specifically, we consider SlimPajama (Soboleva et al., 2023), a cleaned and deduplicated version of RedPajama. To measure alignment with a pretrained model, we evaluate SlimPajama using LLaMA3-1B by computing the model's perplexity (PPL) on the dataset. Formally, given a dataset $\mathcal{D} = x^{(1)}, x^{(2)}, \ldots, x^{(N)}$ and a language model with parameters θ, perplexity is defined as

$$\mathrm{PPL}(\mathcal{D};\theta) = \exp\left(-\frac{1}{\sum_{i=1}^{N} T_i}\sum_{i=1}^{N}\sum_{t=1}^{T_i}\ln p_\theta\left(x_t^{(i)} \mid x_{<t}^{(i)}\right)\right), \tag{6}$$

where T_i is the length of the i-th sequence, and $p_\theta(x_t^{(i)} \mid x_{<t}^{(i)})$ denotes the probability assigned by the model to token $x_t^{(i)}$ conditioned on its preceding context.

As shown in Figure 5, we present the perplexity distribution of the SlimPajama dataset. It can be observed that the dataset exhibits a highly uneven alignment with the LLaMA3 model: the majority of data points are concentrated below a perplexity of 60, while the maximum perplexity reaches several million. We further partitioned the dataset into tertiles based on the 1/3 and 2/3 quantiles, resulting in three subsets: high-PPL, middle-PPL, and low-PPL. Typically, low-PPL data are considered easier for the model to learn, representing well-learned or simple examples, whereas high-PPL data correspond to more difficult or even noisy examples. Under extremely low-bit quantization, quantization-aware training (QAT) can be interpreted as a reconstruction of the model weights (Liu et al., 2025a). Motivated by this, we investigate the impact of high-, middle-, and low-PPL data on

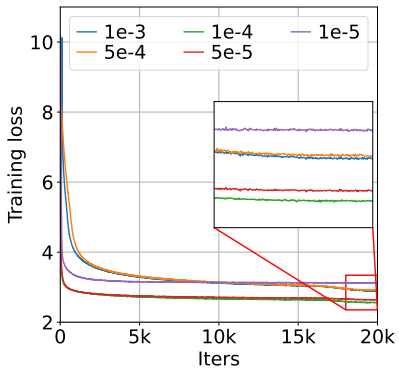


Figure 7: Learning Rate.

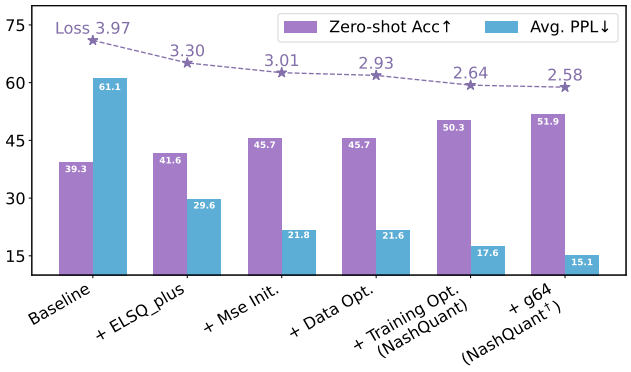


Figure 8: A strong baseline of 2-bit quantized LLMs.

LLM performance under extremely low-bit training. Figure 6 illustrates the zero-shot accuracy across seven downstream tasks. Compared with randomly sampled data from the dataset, high-PPL data generally result in slightly worse performance across most metrics, with particularly poor results on the ARC-e task. Middle-PPL data consistently achieve the best performance across the majority of tasks, whereas low-PPL data lead to highly uneven metrics, with most being significantly degraded.

2.5 Training Hyperparameters

In addition to quantizer design, quantization granularity, initialization strategies, and the selection of training data, the choice of training hyperparameters also plays a crucial role in determining the final accuracy of quantized models, such as learning rate, scheduler, training steps, and weight decay. However, this aspect has been largely overlooked in prior QAT studies. In this work, we conduct a systematic ablation study on training hyperparameters to investigate their impact on the effectiveness of QAT under extremely low-bit quantization.

Learning Rate. As shown in Figure 1, learning rate has a substantial impact on the performance of a given quantizer. A small learning rate (e.g., $1e{-}5$) often leads certain quantizers to converge to excessively high loss values, whereas a large learning rate (e.g., $1e{-}3$) can cause training instability, with the loss diverging to NaN. Through careful design, we find that ELSQ+ consistently converges stably across a wide range of learning rates. To further identify the optimal setting, we conduct a finer-grained ablation study on the learning rate. As illustrated in Figure 7, we visualize the loss curves under five learning rates ($1e{-}3$, $5e{-}4$, $1e{-}4$, $5e{-}5$, $1e{-}5$). The smallest learning rate, $1e{-}5$, exhibits the slowest convergence: although its loss decreases rapidly at the beginning, it soon stagnates without further reduction. In contrast, $5e{-}4$ and $1e{-}3$ achieve comparable convergence speeds, while $1e{-}4$ and $5e{-}5$ consistently yield the fastest loss reductions throughout training. Among them, $1e{-}4$ demonstrates the most stable and effective convergence, enabling the model to reach the best overall performance.

Scheduler. Cosine annealing is widely used in full-precision training of LLMs, while the Warmup-Stable-Decay(WSD) scheduler(Team, 2025) has recently been proposed for 1.58-bit quantization, adopting a warm-up and stable learning rate for the first 80% of steps and decay for the final 20%. Our ablations show that the two schedulers yield highly similar results: WSD improves average zero-shot accuracy over cosine annealing by only 0.1, with nearly overlapping training curves. However, WSD achieves slightly faster early convergence and lower final loss, suggesting it as a preferable choice for quantization-aware training.

Training Steps. Training cost is a critical consideration in QAT. ParetoQ (Liu et al., 2025a) recommends 120k steps for 2-bit quantization, where weights are heavily distorted and require reconstruction. With our improved techniques described above, we revisit this question under our framework, training with 10k, 20k, 40k, 80k, and 120k steps. We find that increasing the step substantially improves quantized model accuracy, with the most pronounced gains observed between 10k and 80k. Beyond 80k, additional training yields minimal improvement. Therefore, 80k steps offer the most balanced trade-off between accuracy and training cost for extremely low-bit quantization.

Table 2: Main results of 2-bit quantization on different evaluation benchmarks. We report the perplexity on Wiki2 and C4, and zero-shot accuracy on 7 downstream tasks. * denotes that the results are reproduced with open-source training data.

Model	Method	Perplexity ↓			Zero-shot Accuracy ↑							
		Wiki2	C4	Avg	BoolQ	PIQA	HS	WG	ARC-e	ARC-c	OBQA	Avg
LLaMA3-1B	FP	9.75	14.01	11.9	63.7	74.2	63.8	60.7	60.3	36.5	37.2	56.6
	GPTQ-g64	283	1061	672	39.7	50.9	27.4	47.8	27.2	25.9	28.0	35.3
	AWQ-g64	2.1e5	2.5e5	2.3e5	45.4	53.4	26.7	49.0	25.7	27.2	28.2	36.5
	OmniQuant-g64	104	202	153	47.0	56.3	28.3	50.0	32.7	21.9	25.6	37.4
	LLM-QAT	16.87	22.84	19.85	58.3	68.8	47.5	52.6	47.9	28.4	**32.6**	48.0
	EfficientQAT	85.52	68.21	76.87	60.6	60.7	33.8	50.1	34.4	24.2	29.2	41.9
	EfficientQAT-g64	17.98	24.09	21.03	56.4	65.4	41.2	52.3	46.3	26.7	30.4	45.5
	ParetoQ*	19.63	25.68	22.65	59.9	66.5	40.1	53.0	40.9	25.2	28.4	44.8
	NashQuant	**13.86**	**21.32**	**17.59**	**62.0**	**71.3**	**51.8**	**56.0**	**50.5**	**29.4**	31.4	**50.3**
LLaMA3-3B	FP	7.81	11.33	9.6	73.0	77.4	73.7	69.2	71.7	46.0	43.2	64.9
	GPTQ-g64	106	578	342	41.3	53.2	29.6	50.9	27.1	23.9	24.6	35.8
	AWQ-g64	3.0e4	2.1e4	2.6e4	41.8	50.3	26.1	48.9	25.7	24.1	25.2	34.6
	OmniQuant-g64	44.75	75.79	60.27	40.9	58.7	35.5	50.7	35.8	24.3	28.4	39.2
	LLM-QAT	12.60	17.52	15.06	**65.8**	73.1	59.6	55.1	58.0	34.0	32.6	54.0
	EfficientQAT	29.46	34.31	31.89	64.9	64.7	46.6	54.0	48.1	27.0	28.2	47.6
	EfficientQAT-g64	11.86	17.16	14.51	64.1	72.1	61.1	61.9	64.2	36.4	36.0	56.5
	ParetoQ*	13.10	18.47	15.78	63.8	71.1	55.9	58.3	53.2	31.1	32.8	52.3
	NashQuant	**10.19**	**16.93**	**13.56**	63.8	**74.2**	**64.8**	**62.6**	**64.3**	**37.2**	**36.4**	**57.6**

Weight Decay. Finally, we examine the effect of weight decay in quantization-aware training. In full-precision LLM training, weight decay is typically set to 0.1, while some quantization methods (Liu et al., 2023; 2025a) adopt 0 without justification. We conduct ablations with coefficients of 0, 0.01, and 0.1, and observe nearly identical loss trajectories across all settings, suggesting that weight decay has little impact on quantized training. Since extremely low-bit quantization primarily focuses on weight reconstruction, a reasonable choice is to set weight decay to 0.

3 Experiments

3.1 Experimental Setup

Baselines. We conduct experiments on LLMs of different scales, including LLaMA3-1B, LLaMA3-3B, and LLaMA3-8B (Dubey et al., 2024). To evaluate the effectiveness of our proposed NashQuant, we compare it with state-of-the-art LLM quantization approaches, covering both PTQ and QAT methods. For PTQ, which calibrates quantization parameters without retraining, we consider OmniQuant (Shao et al., 2023), AWQ (Lin et al., 2024), and GPTQ (Frantar et al., 2022). For QAT, which jointly optimizes quantization parameters during fine-tuning, we include LLM-QAT (Liu et al., 2023), BitDistiller (Du et al., 2024), EfficientQAT (Chen et al., 2024), and ParetoQ (Liu et al., 2025a).

Benchmarks. We evaluate the performance of our proposed NashQuant against existing LLM quantization methods using two metrics: perplexity and zero-shot accuracy, with evaluation conducted via the *lm-evaluation-harness* (EleutherAI, 2021). For perplexity, we report results on WikiText-2 (Merity et al., 2016) and C4 (Raffel et al., 2020), where lower values indicate better language modeling. For zero-shot accuracy, we evaluate on seven downstream tasks: BoolQ (Clark et al., 2019), PIQA (Bisk et al., 2020), HellaSwag (Zellers et al., 2019), Winogrande (Sakaguchi et al., 2021), ARC (Clark et al., 2018), and OpenBookQA (Mihaylov et al., 2018), where higher accuracy indicates stronger performance.

Implementation Details. In this paper, all training is conducted using the SlimPajama dataset (Soboleva et al., 2023) with a sequence length of 2048. For our proposed NashQuant, we select examples within the middle-PPL range. The model is trained with our ELSQ+ quantizer for a total of 80k steps, using a batch size of 128 following ParetoQ (Liu et al., 2025a). We employ the AdamW optimizer , a learning rate of $1e{-}4$, weight decay set to 0, and the WSD scheduler with final decay to 0. Our main experiments adopt channel-wise quantization with MSE-based initialization.

3.2 Main Results

In Table 2, we present the 2-bit results of several commonly used baselines on perplexity and zero-shot tasks. We compare our proposed NashQuant with a range of state-of-the-art PTQ and QAT techniques. It is important to note that the Llama3 model used in our experiments is the publicly available open-source version, rather than a reproduced full-precision model. Given that PTQ methods generally perform suboptimally in 2-bit quantization settings, we report all PTQ results using a group size of 64 to ensure consistency. Our NashQuant outperforms the existing PTQ and QAT methods, setting a new state-of-the-art (SOTA) benchmark.

For the Llama3-1B model, most PTQ methods fail to achieve acceptable performance under 64-group quantization. In contrast, our channel-wise NashQuant demonstrates notable improvements over existing PTQ and QAT baselines. The results shows that our NashQuant outperform EfficientQAT and EfficientQAT-g64 both on perplexity and zero-shot accuracy. Meanwhile, the average perplexity of channel-wise NashQuant is 2.26 lower than that of LLM-QAT, while achieving 5% higher accuracy on average across seven zero-shot tasks. Furthermore, when compared to the recent work ParetoQ, NashQuant achieves nearly a 30% improvement on the Wiki and 17% on the C4, along with a 5.5 points increases in zero-shot accuracy. Notably, ParetoQ requires an additional 40k iterations (10B) to train, underscoring the efficiency of our approach.

For the larger Llama3-3B model, NashQuant also achieves best performance among PTQ and QAT methods. In particular, NashQuant achieves a 22% improvement over ParetoQ on the Wiki2 dataset and a 10% improvement on C4, accompanied by a 5.2-point gain in average accuracy across zero-shot tasks. Moreover, NashQuant achieves a 10% reduction in average perplexity together with a 1.1-point increase in zero-shot task performance when compare to EfficientQAT-g64. While against LLM-QAT, NashQuant attains an additional 1.5-point decrease in average perplexity and a 3.6-point improvement on zero-shot benchmarks. Compared to the full-precision Llama3-3B model, our 2-bit channel-wise NashQuant only drops 1.04 and 2.9 on Wiki2 and C4, respectively, thereby significantly reducing the gap between full-precision and quantized model.

3.3 A Strong QAT baseline for LLMs

In the previous sections, we systematically explored various aspects of extremely low-bit quantization for LLMs, including the design of quantizer, quantization granularity and initialization, selection of QAT training data, as well as the tuning of key hyperparameters such as learning rate, weight decay, and scheduler choices, together with the trade-offs between training cost and accuracy. Building upon these observation, in this section we present a strong baseline for extremely low-bit LLM quantization. As shown in Figure 8, we illustrate the evolution of our model. The baseline employs the LSQ+ quantizer and training strategy following ParetoQ (Liu et al., 2025a). Replacing LSQ+ with our proposed ELSQ+ quantizer significantly reduces the converged loss and improves both perplexity and zero-shot performance. Further applying MSE-based initialization enhances channel-wise quantization and accelerates convergence. Building upon this, optimizing the training data and hyperparameters yields the standard NashQuant presented in the previous subsection. Finally, introducing group-wise quantization produces our strongest baseline NashQuant†.

4 Conclusion

In this study, we present a comprehensive and systematic empirical investigation of extremely low-bit quantization-aware training (QAT) for large language models (LLMs) within a unified framework. Our analysis examines how quantizer design, quantization granularity and initialization, training hyperparameters, and data selection affect both the stability and effectiveness of training, thereby providing deeper insights into extremely low-bit quantization. Based on these findings, we introduce NashQuant, a robust and practical QAT baseline that outperforms existing PTQ and QAT methods under extensive evaluations on LLaMA-3-1B and LLaMA-3-3B. By demonstrating superior performance across diverse settings, NashQuant establishes itself as a reliable benchmark, offering the research community a solid reference for future studies and a fair foundation for comparing subsequent methods in the pursuit of extremely low-bit LLM quantization.

Ethics Statement. This work fully complies with the ICLR Code of Ethics. Our study does not involve human subjects, sensitive personal information, or data that may compromise privacy, safety, or security. All datasets used are publicly available and widely adopted in the research community, and their use aligns with ethical and legal standards. Our contributions primarily advance methodological understanding in quantization research without introducing ethical concerns. The authors have no conflicts of interest, and research integrity has been strictly maintained.

Reproducibility Statement. To ensure reproducibility, we provide detailed descriptions of our methods, training settings, and evaluation protocols in the main paper and appendix. Hyperparameters, training settings, and training procedures are clearly documented, while implementation details and results are included in the supplementary materials. We will also release anonymized source code and processing scripts to facilitate independent verification and replication of our results. All datasets employed are publicly accessible, therefore we ensure that the reported results can be faithfully reproduced and extended by other researchers.

REFERENCES

Yash Bhalgat, Jinwon Lee, Markus Nagel, Tijmen Blankevoort, and Nojun Kwak. Lsq+: Improving low-bit quantization through learnable offsets and better initialization. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 696–697, 2020.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pp. 7432–7439, 2020.

Mengzhao Chen, Wenqi Shao, Peng Xu, Jiahao Wang, Peng Gao, Kaipeng Zhang, and Ping Luo. Efficientqat: Efficient quantization-aware training for large language models. *arXiv preprint arXiv:2407.11062*, 2024.

Mengzhao Chen, Chaoyi Zhang, Jing Liu, Yutao Zeng, Zeyue Xue, Zhiheng Liu, Yunshui Li, Jin Ma, Jie Huang, Xun Zhou, and Ping Luo. Scaling law for quantization-aware training, 2025.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions. *ArXiv*, abs/1905.10044, 2019. URL https://api.semanticscholar.org/CorpusID:165163607.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457, 2018. URL https://api.semanticscholar.org/CorpusID:3922816.

Matthieu Courbariaux, Yoshua Bengio, and Jean-Pierre David. Binaryconnect: Training deep neural networks with binary weights during propagations. *Advances in neural information processing systems*, 28, 2015.

Dayou Du, Yijia Zhang, Shijie Cao, Jiaqi Guo, Ting Cao, Xiaowen Chu, and Ningyi Xu. Bitdistiller: Unleashing the potential of sub-4-bit llms via self-distillation. *arXiv preprint arXiv:2402.10631*, 2024.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. Extreme compression of large language models via additive quantization, 2024.

EleutherAI. lm-evaluation-harness, 2021. URL https://github.com/EleutherAI/lm-evaluation-harness. Accessed: 2025-01-14.

Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019.

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

Xun Liang, Hanyu Wang, Yezhaohui Wang, Shichao Song, Jiawei Yang, Simin Niu, Jie Hu, Dan Liu, Shunyu Yao, Feiyu Xiong, and Zhiyu Li. Controllable text generation for large language models: A survey, 2024.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Wei-Ming Chen, Wei-Chen Wang, Guangxuan Xiao, Xingyu Dang, Chuang Gan, and Song Han. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of machine learning and systems*, 6: 87–100, 2024.

Yifei Liu, Jicheng Wen, Yang Wang, Shengyu Ye, Li Lyna Zhang, Ting Cao, Cheng Li, and Mao Yang. Vptq: Extreme low-bit vector post-training quantization for large language models. In *The 2024 Conference on Empirical Methods in Natural Language Processing*, 2024.

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

Zechun Liu, Changsheng Zhao, Hanxian Huang, Sijia Chen, Jing Zhang, Jiawei Zhao, Scott Roy, Lisa Jin, Yunyang Xiong, Yangyang Shi, et al. Paretoq: Scaling laws in extremely low-bit llm quantization. *arXiv preprint arXiv:2502.02631*, 2025a.

Zihan Liu, Xinhao Luo, Junxian Guo, Wentao Ni, Yangjie Zhou, Yue Guan, Cong Guo, Weihao Cui, Yu Feng, Minyi Guo, Yuhao Zhu, Minjia Zhang, Chen Jin, and Jingwen Leng. Vq-llm: High-performance code generation for vector quantization augmented llm inference. In *2025 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 1496–1509. IEEE, March 2025b. doi: 10.1109/hpca61900.2025.00112. URL http://dx.doi.org/10.1109/HPCA61900.2025.00112.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

Shuming Ma, Hongyu Wang, Lingxiao Ma, Lei Wang, Wenhui Wang, Shaohan Huang, Li Dong, Ruiping Wang, Jilong Xue, and Furu Wei. The era of 1-bit llms: All large language models are in 1.58 bits, 2024. URL https://arxiv.org/abs/2402.17764.

Shuming Ma, Hongyu Wang, Shaohan Huang, Xingxing Zhang, Ying Hu, Ting Song, Yan Xia, and Furu Wei. Bitnet b1.58 2b4t technical report, 2025. URL https://arxiv.org/abs/2504.12285.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *Conference on Empirical Methods in Natural Language Processing*, 2018. URL https://api.semanticscholar.org/CorpusID:52183757.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9):99–106, 2021.

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

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama. https://www.cerebras.net/blog/slimpajama-a-627b-token-cleaned-and-deduplicated-version-of-redpajama, 2023. URL https://huggingface.co/datasets/cerebras/SlimPajama-627B.

Xiaofei Sun, Xiaoya Li, Jiwei Li, Fei Wu, Shangwei Guo, Tianwei Zhang, and Guoyin Wang. Text classification via large language models, 2023.

MiniCPM Team. Minicpm4: Ultra-efficient llms on end devices. 2025.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. QuIP$\#$: Even better LLM quantization with hadamard incoherence and lattice codebooks. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=9BrydUVcoe.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, Yang Wang, and Enhong Chen. Large language models for generative information extraction: A survey, 2023.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? In *Annual Meeting of the Association for Computational Linguistics*, 2019. URL https://api.semanticscholar.org/CorpusID:159041722.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Shujian Huang, Lingpeng Kong, Jiajun Chen, and Lei Li. Multilingual machine translation with large language models: Empirical results and analysis, 2024. URL https://arxiv.org/abs/2304.04675.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models, 2023.

Table 3: Main results of 2-bit quantization on CosineAnealling and WSD Scheduler.

Model	Scheduler	Perplexity ↓			Zero-shot Accuracy ↑							
		Wiki2	C4	Avg	BoolQ	PIQA	HS	WG	ARC-e	ARC-c	OBQA	Avg
LLaMA3-1B	Cosine	14.37	21.55	17.96	57.6	70.8	53.2	56.2	54.3	29.7	34.0	50.8
	WSD	14.61	21.76	18.18	60.4	71.1	52.8	55.8	53.7	29.8	33.0	50.9

A Appendix

A.1 Related works

Post-Training Quantization (PTQ) for LLMs. PTQ is an efficient approach for large language model compression, as it is directly applied to a pretrained model by calibrating with a small amount of data, without necessitating retraining. This makes PTQ particularly attractive when data and computational resources are limited. PTQ approaches can be broadly categorized into scalar quantization and vector quantization. Scalar methods, such as GTPQ(Frantar et al., 2022), perform layer-wise weight quantization utilizing second-order information from the Hessian matrix. AWQ(Lin et al., 2024) reduces outlier effects by introducing channel-wise smooth scaling, which transfers the impact of activation outliers to the salient weights. While, OmniQuant(Shao et al., 2023), in contrast, mitigates the extreme weight outliers by learning optimal upper and lower truncation boundaries. More recently, vector quantization (VQ) techniques have also shown great progress in extremely low-bit quantization. QuIP#(Tseng et al., 2024) leverages Hadamard matrices to compress weights into compact codebooks, whereas AQLM(Egiazarian et al., 2024) incorporates a classic learned additive quantization(AQ) strategy into LLMs by jointly optimizing the codebook with transformer blocks. Additionally, VPTQ(Liu et al., 2024) combines second-order optimization with codebook initialization to further improve vector quantization efficiency. VQ-LLM(Liu et al., 2025b) further alleviates memory and access inefficiencies in traditional vector quantization schemes by introducing codebook caching. Despite these advances, PTQ techniques still face substantial accuracy degradation, particularly in the case of extremely low-bit quantization.

Quantization-Aware Training (QAT) for LLMs. Although PTQ methods are widely used for extremely low-bit LLM quantization, their fine-grained granularity is suboptimal for hardware deployment and accuracy degradation still remains unavoidable. To address these limitations, quantization-aware training (QAT) has been introduced, which alleviates accuracy loss by leveraging additional training data and extra learnable parameters. LLM-QAT(Liu et al., 2023) and Bitdistiller(Du et al., 2024) integrate knowledge distillation to preserve performance, while EfficientQAT(Chen et al., 2024) introduces a block-wise training scheme that reduces the memory and computational burden. More recently, ParetoQ(Liu et al., 2025a) employs the stretched elastic quantization(SEQ) and achieves state-of-the-art results in 2-bit weight-only QAT. Additionally, ternary quantization, such as BitNet b1.58(Ma et al., 2024; 2025) and BitCPM(Team, 2025) either train models from scratch or inheriting pre-trained model weights, offering valuable insights into training strategies and optimization for extremely low-bit QAT. Despite these advances, a systematic understanding of how quantizer design and training strategies jointly affect extremely low-bit QAT remains unexplored.

A.2 Training Hyperparameters

To supplement the discussion in the main paper, we provide additional figures and tables related to experiments on training hyperparameters. Table 3 compares the results using the cosine annealing and WSD schedulers, while Figure 9 presents their corresponding loss curves. Table 4 reports the outcomes across five different learning rates, and Table 10 illustrates the effect of varying weight decay settings on quantization-aware training.

A.3 Implementation Details for Baselines

We reproduce the baseline results reported in our main experiments using the publicly available code. The specific training details are summarized in Table 5 and most settings are aligned with those described in the original papers. For a fair comparison, LLM-QAT is not trained with any self-generated data and all reproductions use the SlimPajama dataset. It is important to note that

Table 4: Main results of 2-bit quantization on different training iterations.

Model	Iterations	Perplexity ↓			Zero-shot Accuracy ↑							
		Wiki2	C4	Avg	BoolQ	PIQA	HS	WG	ARC-e	ARC-c	OBQA	Avg
LLaMA3-1B	10k	13.89	20.64	17.26	60.4	69.7	51.4	56.2	53.8	29.0	32.0	50.4
	20k	13.68	20.38	17.03	59.8	70.4	52.0	56.7	52.1	29.4	33.0	50.5
	40k	13.30	20.36	16.83	58.6	71.7	52.5	55.6	52.4	29.8	32.6	50.4
	80k	13.21	20.49	16.85	59.6	70.9	**53.0**	**56.8**	54.2	**30.5**	**35.0**	**51.4**
	120k	**13.03**	**20.41**	**16.72**	**61.8**	**71.4**	53.0	55.0	**55.1**	29.9	32.8	51.3

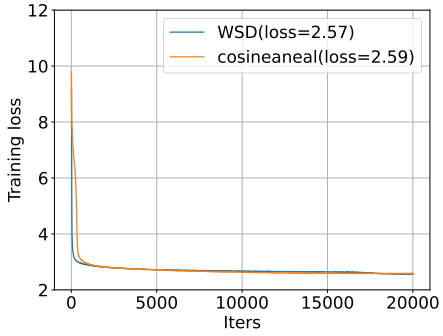


Figure 9: Comparison of different schedulers.

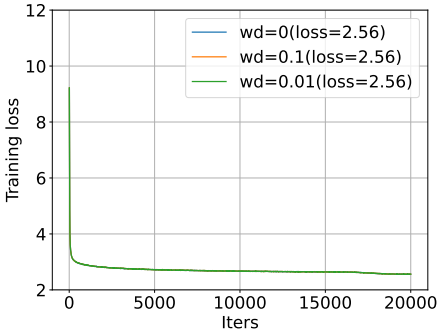


Figure 10: Comparison of different wd.

EfficientQAT is originally a two-stage training method. However, during our reproduction of channel-wise quantization, we encountered an unresolved bug, so the reported EfficientQAT results in Table 2 correspond only to the first stage. In contrast, EfficientQAT-g64 results are obtained following the full two-stage training as intended.

A.4 A General QAT Recipe

Based on our preceding analyses, we propose a simple and efficient training approach for low-bit quantization-aware training. As shown in Table 5, we recommend an initial learning rate of $1e-4$ with the WSD scheduler, a batch size of 128, and 80k training iterations. The optimizer remains AdamW, and training is performed in a single end-to-end stage with all parameters updated. Most importantly, we strongly recommend training with our proposed ELSQ+ quantizer. Data selection is also important, but using data from the same source as that for our large model training could further improve results. Since our evaluation does not involve long sequences, a sequence length of 2k is considered sufficient.

A.5 LLM Declaration

In this study, Large Language Models (LLMs) are only utilized for polishing and refining the academic writing, ensuring the paper with clarity, coherence, and academic tone. Its role is limited to assisting in improving the structure and fluency of the text, with all scientific content and findings being solely the result of the authors' work.

Table 5: Implementation Details of QAT baselines and proposed general QAT recipe.

	LLM-QAT	**EfficientQAT**	**ParetoQ**	**A General QAT recipe**
Training Data	Slimpajama	Slimpajama	Slimpajama	Slimpajama
Sequence Length	2048	4096	2048	2048
Batch Size	128	2, 32	128	128
Optimizer	AdamW	AdamW	AdamW	AdamW
Traing Steps	120k	{4k,128}	120k	80k
Training stages	1	2	1	1
Scheduler	Cosine Aneal	Cosine Aneal	Cosine Aneal	WSD
Initial LR	$2e-5$	$\{1e-4, 2e-5\}$	$2e-5$	$1e-4$
Initialization	MinMax	MinMax	MinMax	MSE
Weight Decay	0	0	0	0