OTARo: Once Tuning for All Precisions toward Robust On-Device LLMs

Shaoyuan Chen[1,2,†,△], **Zhixuan Chen**[1,†], **Dawei Yang**[1,*,†], **Zhihang Yuan**[3], **Qiang Wu**[1]

[1]Houmo AI
[2]Sun Yat-sen University
[3]Peking University
chenshy299@mail2.sysu.edu.cn, zhixuan.chen@houmo.ai, dawei.yang@houmo.ai, yuanzhihang@pku.edu.cn, qiang.wu@houmo.ai

Abstract

Large Language Models (LLMs) fine-tuning techniques not only improve the adaptability to diverse downstream tasks, but also mitigate adverse effects of model quantization. Despite this, conventional quantization suffers from its structural limitation that hinders flexibility during the fine-tuning and deployment stages. Practical on-device tasks demand different quantization precisions (i.e. different bit-widths), e.g., understanding tasks tend to exhibit higher tolerance to reduced precision compared to generation tasks. Conventional quantization, typically relying on scaling factors that are incompatible across bit-widths, fails to support the on-device switching of precisions when confronted with complex real-world scenarios. To overcome the dilemma, we propose OTARo, a novel method that enables on-device LLMs to flexibly switch quantization precisions while maintaining performance robustness through once fine-tuning. OTARo introduces Shared Exponent Floating Point (SEFP), a distinct quantization mechanism, to produce different bit-widths through simple mantissa truncations of a single model. Moreover, to achieve bit-width robustness in downstream applications, OTARo performs a learning process toward losses induced by different bit-widths. The method involves two critical strategies: (1) Exploitation-Exploration Bit-Width Path Search (BPS), which iteratively updates the search path via a designed scoring mechanism; (2) Low-Precision Asynchronous Accumulation (LAA), which performs asynchronous gradient accumulations and delayed updates under low bit-widths. Experiments on popular LLMs, e.g., LLaMA3.2-1B, LLaMA3-8B, demonstrate that OTARo achieves consistently strong and robust performance for all precisions.

Introduction

In recent years, Large Language Models (LLMs) have shown outstanding capabilities in Natural Language Processing (NLP) (Achiam et al. 2023; Zubiaga 2024), and on-device deployment of LLMs has emerged as a frontier research direction to enable real-time responsiveness, privacy, and personalization in intelligent terminal services (Qu et al. 2025; Tang et al. 2025). With advances in processing power, memory bandwidth, and storage capacity of edge devices, on-device deployment of compact LLMs becomes feasible.

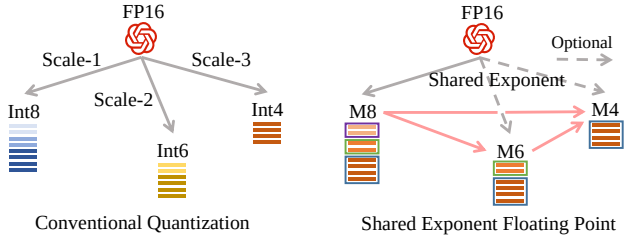


Figure 1: A comparison of conventional quantization and Shared Exponent Floating Point (SEFP) in supporting dynamic precision switching. Gray arrows represent quantization, while red arrows indicate cross-precision conversion. M8, M6, and M4 denote mantissa bit-widths in SEFP.

In parallel, model compression techniques, e.g., quantization (Achiam et al. 2023; Lin et al. 2024), reduce deployment costs while incurring some accuracy degradation.

For a better inference performance of LLMs in devices, fine-tuning techniques play a pivotal role by not only enhancing the adaptability across downstream tasks, but also effectively mitigating the accuracy degradation induced by quantization (Yang et al. 2024b; Lu, Luu, and Buehler 2025). Despite this, conventional quantization still faces challenges in complex real-world scenarios due to its inherent limitation in switching bit-widths.

Practical on-device tasks require different quantization precisions, corresponding to different bit-widths. For instance, generation tasks tolerate increased inference time in exchange for a higher precision, whereas understanding tasks can obtain immediate responses with a lower precision (Manduchi et al. 2024). In addition, different precisions can be employed in the prefill and decoding phases to optimize the inference efficiency (Qiao et al. 2025). Consequently, the practical on-device deployment necessitates support for all precisions, rather than a model with a fixed precision.

In conventional quantization, the original weights are scaled and clipped into different numerical ranges depending on the bit-width, which prevents the reuse of scaling factors (scales) across precisions. Once quantized to a specific bit-width, the model loses the flexibility to switch between precisions. Edge devices can maintain a model zoo containing models of each precision, which substantially increases

*Corresponding author
†Equal contribution
△Work done as an intern at Houmo AI

storage overhead and extends the total fine-tuning time. Alternatively, compressing a full- or half-precision model into various quantization precisions at runtime still requires offline fine-tuning for bit-specific and high-quality scaling factors, thereby incurring similar fine-tuning burdens. It also increases storage demands and introduces non-negligible computational costs during on-device quantization.

To address the challenges, this paper proposes OTARo (**O**nce **T**uning for **A**ll Precisions toward **Ro**bust On-Device LLMs), which obtains one unified model through once fine-tuning to support all quantization precisions. OTARo introduces Shared Exponent Floating Point (SEFP), a distinct quantization mechanism. Notably, compared to conventional quantization, SEFP eliminates scaling factors and allows flexible switching of precisions, as illustrated in fig. 1. In SEFP, an exponent is shared in each group of parameters, and individual mantissas are maintained for each parameter. After aligning exponents, any expected bit-width can be achieved through simple mantissa truncation, and the higher bit-widths can be easily converted to the lower ones.

Furthermore, to enhance performance robustness across bit-widths, OTARo introduces an efficient learning scheme that jointly optimizes for mixed quantization losses induced by different SEFP configurations. The learning process incorporates an iterative bit-width selection, and adopts delayed updates to mitigate the effects of low-precision training. Generally, by unifying all precisions within one fine-tuned model and obviating the need for separate fine-tuning per bit-width, OTARo substantially enhances resource efficiency and deployment flexibility in practical applications.

The contributions of our work are summarized as follows:

- We propose OTARo, a novel fine-tuning paradigm for building robust on-device LLMs. OTARo produces a unified model through once fine-tuning to support all quantization precisions. It introduces Shared Exponent Floating Point (SEFP) to enable lightweight and hardware-friendly multi-precision adaptation, and jointly update weights for quantization errors from different bit-widths.
- We identify the commonality on gradient directions under all bit-width settings, and then propose Exploitation-Exploration Bit-Width Path Search (BPS) strategy to determine the optimal sequence of bit-widths.
- We observe that low-precision training induces severe gradient oscillations. To mitigate the oscillations, we conduct high-dimensional projection analysis on the gradients, and propose Low-Precision Asynchronous Accumulation (LAA) strategy, which performs asynchronous gradient accumulations and delayed updates.
- We conduct the comprehensive evaluations on popular LLMs, e.g., LLaMA3.2-1B and LLaMA3-8B. Experimental results show that OTARo consistently outperforms baselines across all tested bit-widths, in both task-specific and zero-shot settings, highlighting its robustness for multi-precision deployment of LLMs.

Related Work

Quantization

Quantization is a critical technique to reduce the memory footprint and inference costs (Wei et al. 2024). Post-Training Quantization (PTQ) methods quantize a pre-trained model without retraining. GPTQ (Frantar et al. 2022) uses arbitrary order quantization, lazy batch updates, and Cholesky reformulation. CFWS (Yang et al. 2024a) introduces coarse & fine weight splitting and an enhanced KL calibration metric. AWQ (Lin et al. 2024) finds the salient weights based on the activation distribution, and scales the salient weight channels through an equivalent transformation. QuaRot (Ashkboos et al. 2024) uses Hadamard transform and computational invariance to mitigate outliers in activations. SpinQuant (Liu et al. 2024) uses a learnable rotation matrix that is fused into weights. Any-Precision LLM (Park et al. 2024) performs incremental upscaling and specialized hardware design to merge multiple integer formats. QUARK (Zhao et al. 2025) proposes a reordering-based group quantization scheme. Quantization-Aware Training (QAT) methods incorporate fake quantization in training, allowing models to lea quantization noises (Liu et al. 2023; Chen et al. 2024).

Shared Exponent Floating Point

Shared Exponent Floating Point (SEFP) quantization shares an exponent in each parameter group while maintaining individual mantissas for each parameter (Gao et al. 2024; Zhou et al. 2025). SEFP quantization procedure consists of two main steps, illustrated in fig. 2. Firstly, the maximum exponent of each parameter group is selected as the shared one before each mantissa is shifted right according to the difference between the shared exponent and its original one. Secondly, shifted mantissas are truncated to a fixed width. SEFP bit-width is typically donated as EeMm, where e and m denote the bit-width of exponents and mantissas, respectively. Accuracy declines arise almost from the forced truncation in Step 2; the impacts of mantissa overflow in step 1 is negligible. (Kalliojarvi and Astola 2002).

Method

Preliminaries

Due to the non-differentiable operations involved in SEFP quantization, such as mantissa right-shifting and truncation, we incorporate the Straight-Through Estimator (STE) (Yin et al. 2019) to approximate the gradients, and introduce SEFP quantization errors into training losses to facilitate the learning process. The approach is formulated as follows:

$$\frac{\partial Q(\omega, b)}{\partial \omega} = 1 \tag{1}$$

$$\mathcal{L} = \text{Loss}(y, Q(\omega, b)x) \tag{2}$$

$$\frac{\partial \mathcal{L}}{\partial \omega} = \frac{\partial \mathcal{L}}{\partial Q(\omega, b)} \frac{\partial Q(\omega, b)}{\partial \omega} = \frac{\partial \mathcal{L}}{\partial Q(\omega, b)} \tag{3}$$

where, ω is weight, $Q(\cdot, b)$ is SEFP function targeting bit-width b, x is input, y is label, $\text{Loss}(\cdot)$ is training loss function. In the above manner, OTARo enables models to adapt

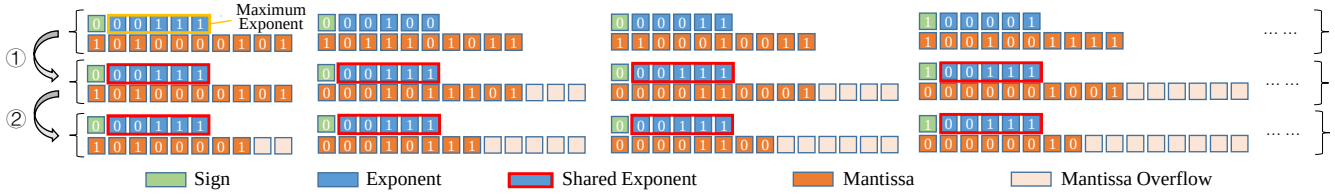


Figure 2: A simple illustration of SEFP quantization (e.g., from FP16 to E5M8) of a group of parameters. Step 1 shows the exponent alignment and mantissa right-shifting for FP16 values. Step 2 shows the forced mantissa truncation.

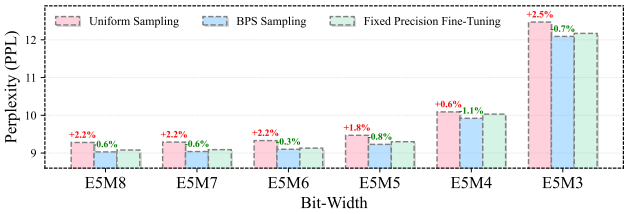


Figure 3: A comparison of uniform sampling, BPS sampling and fixed precision fine-tuning. We report perplexity changes of the first two approaches relative to the last one. In the experiments, LLaMA3.2-1B is fine-tuned and evaluated on the WikiText2 train and test set, respectively.

to quantization errors from different SEFP bit-widths, and fixed precision fine-tuning only learns for quantization errors from a specific bit-width.

Problem Statement

In OTARo, once fine-tuning is performed to obtain a model robust to all bit-widths in SEFP. Each bit-width corresponds to a certain level of precision, with higher bit-widths yielding higher precision. The objective can be formulated as:

$$\omega = \arg\min_{\omega} \frac{1}{|\mathcal{B}|} \sum_{b \in \mathcal{B}} \mathbb{L}(Q(\omega, b)) \tag{4}$$

where, b is the bit-width, $\mathcal{B}$ is the set of bit-widths, $\mathbb{L}(\cdot)$ is test loss function, and $Q(\omega, b)$ is SEFP function under bit-width b. We aim to develop a single model to maintain high performance across all bit-widths, achieving accuracy comparable to the original model before quantization, thus supporting robust and adaptable on-device deployment of LLMs.

Exploitation-Exploration Bit-Width Path Search

To ensure robustness across bit-widths, an intuition is to uniformly sample all target bit-widths, and facilitate models to update based on the gradients induced by quantization errors of different bit-widths. However, our empirical results indicate that this straightforward sampling approach does not align with fixed precision fine-tuning in performance, as shown in fig. 3. This finding prompts us to rethink: whether different bit-widths should be sampled non-uniformly, and scheduled in a deliberate order during fine-tuning.

To assess the feasibility of sampling all bit-widths for robustness, and to inform the design of a practical implementation strategy, we revisit bit-width sampling from a gradient-centric perspective, as illustrated in fig. 4.

It can be observed that the gradients produced at different bit-widths exhibit a certain degree of similarity, which supports the feasibility of enhancing robustness by sampling all bit-widths. Furthermore, for both the higher and lower bit-widths, the gradient direction shows higher consistency with those of the higher ones, but less consistency even with that of the adjacent lower one. This commonality suggests the existence of a larger shared potential subspace between higher bit-widths and others in terms of gradient directions.

Motivated by the observation and analysis, we argue that the bit-width sampling should not only explore all bit-widths to sufficiently learn the loss landscapes across them, but also gradually update toward higher bit-widths that exhibit stronger alignment with the others in gradient directions.

Accordingly, we propose the Exploitation-Exploration Bit-Width Path Search (BPS) strategy, which iteratively selects the bit-width that achieves the highest score in a designed scoring mechanism. The scoring mechanism is formulated as follows:

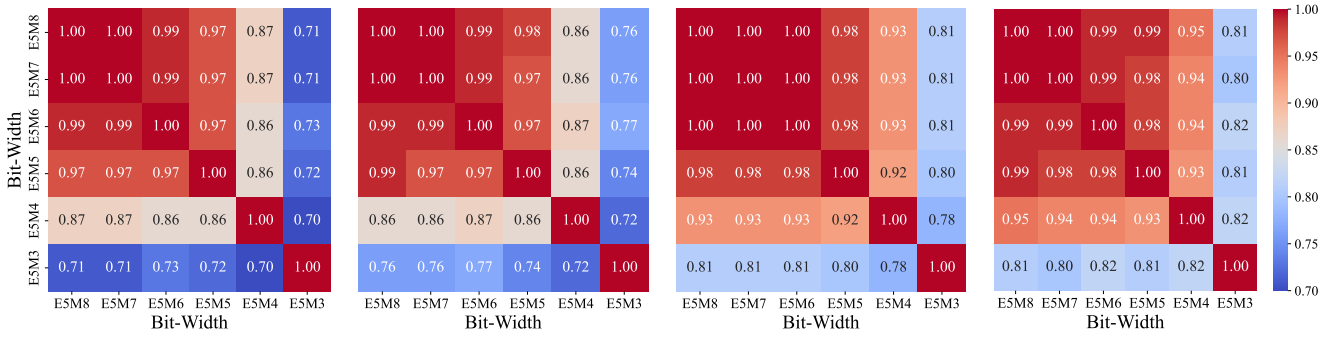
$$\text{Score}(b) = \lambda \sqrt{\frac{\ln t}{t_b}} - \mathcal{L}_b \tag{5}$$

where, the score of bit-width b is denoted as $\text{Score}(b)$, $\mathcal{L}_b$ is the real-time loss for bit-width b, t is the current number of training batches, t_b is the search frequency of b, and exploration coefficient λ is a constant multiplied by the exploration term.

This design encourages the bit-width search path to explore underutilized bit-widths. As fine-tuning progresses, the exploration term of frequently used bit-widths gradually decreases, naturally leading to sampling underutilized ones. The dynamic balance ensures diversity in bit-width selection. This design also ensures that, as the number of batches increases, higher bit-widths are selected more frequently than lower ones. To validate this convergence property, we examine whether the score discrepancy between the higher and lower bit-widths consistently converges toward a positive value as t increases. Suppose h and l denote two distinct bit-widths, and h is the higher one. The score difference is defined as:

$$\Delta = (\lambda \sqrt{\frac{\ln t}{t_h}} - \mathcal{L}_h) - (\lambda \sqrt{\frac{\ln t}{t_l}} - \mathcal{L}_l) \tag{6}$$

where, T is the total number of batches, t_h, t_l are respectively the current counts of h and l being searched (approximately increasing linearly), $\mathcal{L}_h$, $\mathcal{L}_l$ are the corresponding



(a) q proj (b) k proj (c) v proj (d) down proj

Figure 4: Cosine similarities between the gradients produced by different bit-widths of the LLaMA3.2-1B layer-15 q/k/v/down projector. There exists a certain degree of similarity between gradients under different bit-widths. Furthermore, the gradient at each bit-width tends to exhibit stronger similarity with that of its higher bit-width. For example, for q projector, the cosine similarity of gradients between E5M5 and E5M8/E5M7/E5M6 is 0.97, whereas the cosine similarity with E5M4 and E5M3 decreases to 0.86 and 0.72, respectively.

losses at bit-widths h and l, with $\mathcal{L}_l$ generally being larger due to the lower precision of l. We reorganize the terms in the above equation:

$$\Delta = (\mathcal{L}_l - \mathcal{L}_h) + \lambda(\sqrt{\frac{t}{t_h}} - \sqrt{\frac{t}{t_l}})\sqrt{\frac{\ln t}{t}} \tag{7}$$

As t increases to a high value, Δ approaches $\mathcal{L}_l - \mathcal{L}_h$, which is increasingly likely to be positive:

$$\lim_{t \to T} \sqrt{\frac{\ln t}{t}} \to 0 \tag{8}$$

$$\lim_{t \to T} \Delta \to \mathcal{L}_l - \mathcal{L}_h \tag{9}$$

Therefore, the search path in the BPS strategy gradually converges toward the higher bit-widths with smaller losses and more robust gradient directions. As show in fig. 3, the sampling based on the BPS strategy can match or even outperform fixed-precision fine-tuning in each bit-width.

Low-Precision Asynchronous Accumulation

Through the BPS strategy, we obtain a search path that sufficiently explores multiple bit-widths while gradually favoring higher ones. However, the robustness of fine-tuned models suffers from the impacts of low bit-widths. To address this issue, we propose the Low-Precision Asynchronous Accumulation (LAA) strategy, which utilizes a distinctive weight update scheme under continuous low-bit width sampling conditions. To demonstrate the problem, we perform a detailed analysis of errors introduced by SEFP quantization:

$$\mathcal{L}_q = \frac{1}{2}||xQ(\omega, m) - x\omega||^2 \tag{10}$$

$$= \frac{1}{2}||\frac{x[\omega 2^m]}{2^m} - x\omega||^2 \tag{11}$$

$$\frac{\partial \mathcal{L}_q}{\partial \omega} = \frac{x\left(\omega 2^m - [\omega 2^m]\right)}{2^m} \cdot \frac{\partial Q(\omega, m)}{\partial \omega} \tag{12}$$

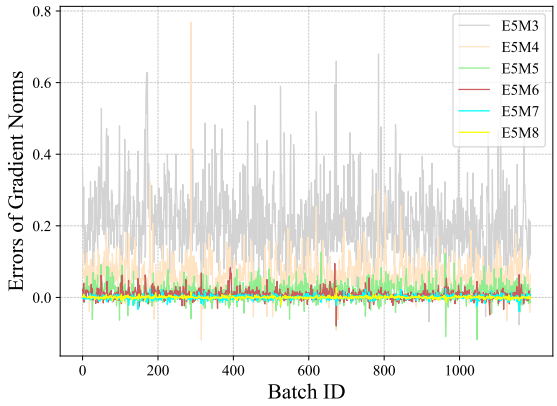


Figure 5: The errors of gradient norms $||\nabla_{\text{sefp}}|| - ||\nabla_{\text{fp}}||$ under different SEFP bit-widths. Gradients are calculated on LLaMA3.2-1B layer-15 down projector.

where, $\mathcal{L}_q$ denotes an approximation of the SEFP error, $\partial \mathcal{L}_q / \partial \omega$ is the derivative of the error with respect to the weight., m denotes the number of mantissa bits in SEFP, and $[\cdot]$ is rounding function. Suppose ω_0 is an arbitrary parameter in ω, we define the function $\epsilon(\omega_0)$ as:

$$\epsilon(\omega_0) = \frac{\omega_0 2^m - [\omega_0 2^m]}{2^m} \tag{13}$$

Function $\epsilon(\omega_0)$ describes a sawtooth wave with both period and amplitude equal to $1/2^m$ (see Appendix A for visualization). Due to the discontinuities introduced by the rounding operation, the function linearly increases from 0 within each period, then abruptly drops to a negative value before jumping back to 0, forming a periodic oscillatory pattern. When m is small, the wave exhibits larger amplitudes, leading to more pronounced oscillations. In low bit-width settings, changes in parameters can induce significant and periodic variations in SEFP quantization errors, resulting in

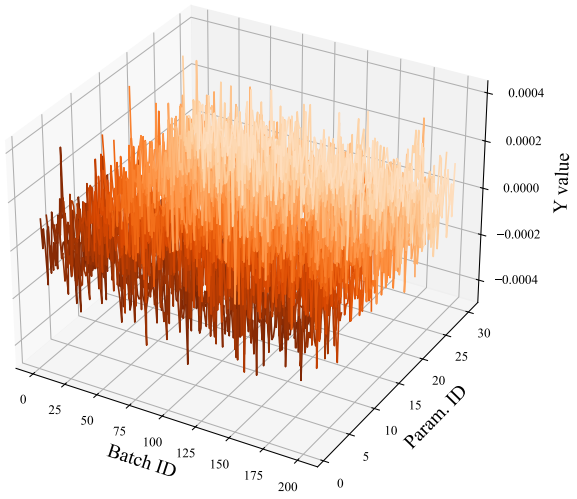


Figure 6: Y values. For model LLaMA3.2-1B and dataset WikiText2, the figure shows the Y values of 30 gradient values in 200 batches, and indicates that $\mathbb{E}[Y] \approx 0$.

periodic and intense oscillations in both loss and gradient during fine-tuning.

Suppose that the gradient in FP16, SEFP fine-tuning are respectively ∇_{fp} and ∇_{sefp}, the error of gradient norms induced by SEFP is $||\nabla_{\text{sefp}}|| - ||\nabla_{\text{fp}}||$. As shown in fig. 5, we observe that the oscillations in $||\nabla_{\text{sefp}}|| - ||\nabla_{\text{fp}}||$ become more intense as the bit-width decreases, and the oscillation exhibits an overall periodic pattern, providing indirect support for the above inference.

Based on these analysis, the error introduced by SEFP quantization is a critical issue to be tackled under low bit-widths. To measure the space distance between gradients with and without introducing SEFP quantization errors (i.e., ∇_{fp} and ∇_{sefp}, respectively), we model the relationship of them through a high-dimensional linear mapping:

$$\nabla_{\text{sefp}} = X \cdot \nabla_{\text{fp}} + Y \tag{14}$$

where X denotes the global linear mapping matrix from ∇_{fp} to ∇_{sefp}, and Y represents the residual perturbation term introduced by quantization, which corresponds to the non-linear and unexplained quantization noise. Estimation of X,Y is described in Appendix B, using Least Squares Method (LSM).

To empirically validate the statistical properties of the perturbation Y, we conduct experiments under the low bit-width setting (e.g., E5M3). As shown in fig. 6, the results indicate that while Y exhibits considerable fluctuation between batches, its mean remains close to zero, that is,

$$\mathbb{E}[Y] \approx 0 \tag{15}$$

In the LAA strategy, gradients are asynchronously accumulated over N batches to generate a gradient ∇^N_{sefp} to update models. ∇^N_{sefp} can be expressed as:

$$\nabla^N_{\text{sefp}} = \sum_{i=1}^{N} \nabla_{\text{sefp},i} = X \sum_{i=1}^{N} \nabla_{\text{fp},i} + \sum_{i=1}^{N} Y_i. \tag{16}$$

Algorithm 1: Overall Pipeline of OTARo.

Require: Batch id t, total number of batches T, bit-width b, training loss function Loss(), weight ω, quantized ω under a bit-width b $Q(\omega, b)$, exploration coefficient λ, search count of a bit-width b t_b, inputs x, labels y, batch id and also a flag for gradient accumulations i (starts from 0), delayed interval N, learning rate η.

for each batch $t = 1$ to T **do**
  Calculate scores: $\text{Score}(b) = \lambda\sqrt{(\ln t)/t_b} - \mathcal{L}_b$
  Select optimal bit-width $b^* = \arg\max_b \text{Score}(b)$
  Calculate the training loss: $\mathcal{L} = \text{Loss}(y, Q(\omega, b^*)x)$
  Calculate the gradient: $\nabla_{\text{sefp}} = \partial\mathcal{L}/\partial Q(\omega, b)$
  if bit-width is ultra-low **then**
    if i is 0 **then**
      Initialize the general gradient: $\nabla^N_{\text{sefp}} \leftarrow \nabla_{\text{sefp}}$
    else
      Accumulate gradients: $\nabla^N_{\text{sefp}} \leftarrow \nabla^N_{\text{sefp}} + \nabla_{\text{sefp}}$
    end if
    $i \leftarrow i + 1$
    if i is N **then**
      Update weights: $\omega \leftarrow \omega - \eta\nabla^N_{\text{sefp}}$
      $i \leftarrow 0$
    end if
  else
    Standard gradient updates: $\omega \leftarrow \omega - \eta\nabla_{\text{sefp}}$
  end if
end for

Suppose independence or weak correlation among the Y_i, the relative influence of the perturbation with N rising diminishes as:

$$\frac{\left\|\sum_{i=1}^{N} Y_i\right\|}{\left\|\sum_{i=1}^{N} \nabla_{\text{fp},i}\right\|} \propto \frac{1}{\sqrt{N}} \to 0 \tag{17}$$

Accordingly, the parameter update rule under the LAA strategy becomes:

$$\omega \leftarrow \omega - \eta \sum_{i=1}^{N} \nabla_{\text{sefp},i} = \omega - \eta X \sum_{i=1}^{N} \nabla_{\text{fp},i} - \eta \sum_{i=1}^{N} Y_i \tag{18}$$

where η is the learning rate. For the cumulative perturbation tends toward zero expectation, the high-frequency oscillations introduced by low bit-widths are effectively mitigated. This stabilizes the training process and enhances convergence. In addition, asynchronous accumulation avoids the memory growth in the computing device.

Finally, the pseudo-code (algorithm 1) is presented to more vividly demonstrate the overall pipeline.

Experiments

In this section, we present a comprehensive evaluation of the proposed OTARo method. We report the experimental setup, i.e., models, datasets, methods and implementation details, and show the overall results. In addition, we perform ablation studies, and calculate the memory reduction and speedup of SEFP quantization.

Models	Methods	E5M8	E5M7	E5M6	E5M5	E5M4	E5M3
LLaMA2-7B	Before Fine-Tuning	57.64%	57.65%	57.64%	57.42%	57.79%	56.99%
	FP16 Fine-Tuning	58.24%	58.27%	58.14%	57.96%	57.82%	56.46%
	Fixed Precision Fine-Tuning	58.31%	58.32%	58.27%	58.20%	58.14%	57.09%
	Ours	**59.15%**	**59.09%**	**58.93%**	**58.75%**	**58.70%**	**57.72%**
LLaMA2-13B	Before Fine-Tuning	60.87%	60.75%	60.76%	60.87%	60.40%	60.41%
	FP16 Fine-Tuning	61.43%	61.42%	61.43%	61.13%	60.84%	60.01%
	Fixed Precision Fine-Tuning	61.54%	61.43%	61.50%	61.17%	61.23%	60.63%
	Ours	**62.33%**	**62.05%**	**62.11%**	**61.83%**	**61.89%**	**61.10%**
LLaMA3-8B	Before Fine-Tuning	62.44%	62.35%	62.49%	62.31%	62.01%	59.81%
	FP16 Fine-Tuning	63.42%	63.50%	63.44%	63.29%	61.23%	59.62%
	Fixed Precision Fine-Tuning	63.89%	63.64%	63.55%	63.34%	62.30%	59.84%
	Ours	**64.09%**	**64.12%**	**63.99%**	**63.84%**	**63.34%**	**60.67%**
LLaMA3.2-3B	Before Fine-Tuning	57.21%	57.10%	57.39%	57.29%	56.68%	54.52%
	FP16 Fine-Tuning	57.80%	57.75%	57.92%	57.74%	56.78%	54.57%
	Fixed Precision Fine-Tuning	57.89%	58.08%	58.14%	57.85%	57.00%	54.78%
	Ours	**58.48%**	**58.46%**	**58.64%**	**58.21%**	**57.82%**	**56.03%**
Qwen3-8B	Before Fine-Tuning	64.45%	64.38%	64.72%	63.29%	63.13%	58.65%
	FP16 Fine-Tuning	65.82%	65.62%	65.80%	64.95%	63.62%	58.60%
	Fixed Precision Fine-Tuning	65.93%	66.10%	65.91%	65.14%	64.82%	61.05%
	Ours	**66.74%**	**66.64%**	**66.58%**	**66.19%**	**66.13%**	**61.51%**

Table 1: Zero-shot results of LLaMA2-7B, LLaMA2-13B, LLaMA3-8B, LLaMA3.2-3B, Qwen3-8B. We report average accuracies of all zero-shot tasks, i.e., Arc-Challenge, Arc-Easy, BoolQ, HellaSwag, MATHQA, OpenBookQA, PIQA, WinoGrande.

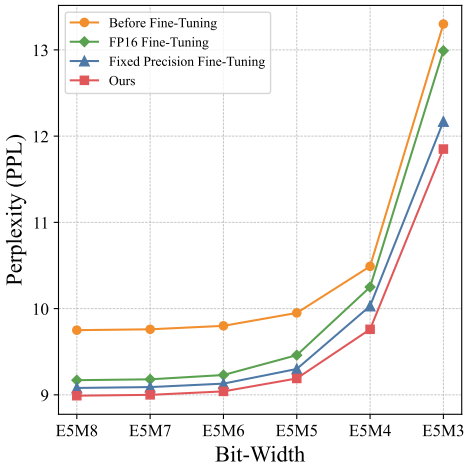


Figure 7: Task-specific fine-tuning results of LLaMA3.2-1B.

Setup

Models In the current development of LLMs, LLaMA (Touvron et al. 2023; Grattafiori et al. 2024) and Qwen(Yang et al. 2025) families have gained popularity due to strong performance and broad community support. Accordingly, in zero-shot experiments, we use LLaMA2-7B, LLaMA2-13B, LLaMA3-8B, LLaMA3.2-3B and Qwen3-8B, and in task-specific fine-tuning experiments, we use LLaMA3.2-1B.

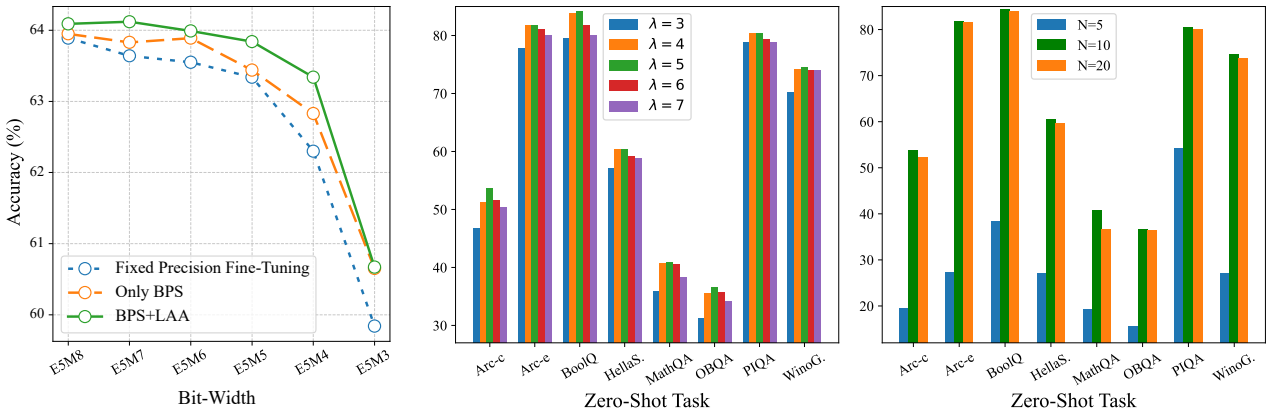
Datasets For zero-shot experiments, we adopt the Alpaca (Taori et al. 2023) dataset. Each model is fine-tuned on Alpaca, and evaluated on a suite of benchmarks spanning different reasoning and understanding skills: ARC-Easy, ARC-Challenge (Clark et al. 2018), BoolQ (Clark et al. 2019), HellaSwag (Zellers et al. 2019), MATHQA (Amini et al. 2019), OpenBookQA (Mihaylov et al. 2018), PIQA (Bisk et al. 2020), WinoGrande (Sakaguchi et al. 2021). For task-specific fine-tuning experiments, we adopt the WikiText2 (Merity et al. 2016) dataset. The model is fine-tuned and evaluated using its train and test set, respectively.

Methods We evaluate pre-trained models to show the optimizations from fine-tuning. FP16 fine-tuning and fixed precision fine-tuning are also included as baselines to highlight the bit-width robustness achieved by OTARo. In fixed precision fine-tuning, the backpropagation-based weight update mechanism aligns with that employed in OTARo. Fixed precision fine-tuning make the models adapt to quantization errors from each fixed bit-width, while multiplying the total fine-tuning time.

Implementation Details In the experiments, we set the learning rate to 1e-5, and use SGD optimizer. SEFP group size is set to 64, and the bit-widths include {E5M8, E5M7, E5M6, E5M5, E5M4, E5M3}. In the BPS strategy, exploration coefficient λ is set to 5, and in the LAA strategy, delayed updates are performed with a delay step N=10. For zero-shot experiments, each model is fine-tuned in a single epoch to avoid excessive adaptation, and evaluated with accuracy (%). For task-specific fine-tuning experiments, the model is fine-tuned in 20 epochs for a well-converged state, and evaluated with perplexity, i.e. PPL. In ablation studies, strategies comparison, λ, and N are included. All experiments are conducted on NVIDIA A6000 GPUs.

Zero-Shot Results

We report overall results of zero-shot experiments in table 1, where OTARo consistently achieves the highest average ac-



(a) strategies comparison (b) exploration coefficient λ (c) delay step N

Figure 8: Ablation results. We show the ablation results for strategies, exploration coefficient λ, delay step N. For strategies comparison, we report average accuracies of all zero-shot tasks, and for the others, we report E5M8 accuracies in each task.

curacy across all bit-widths (E5M8 to E5M3) for all models.

For challenging low-bit settings (E5M4, E5M3), OTARo also obtains high accuracies and outperforms the baselines. For example, in LLaMA3-8B, OTARo achieves 63.34% at E5M4 and 60.67% at E5M3, compared with 62.30% and 59.84% from fixed precision fine-tuning, and 61.23% and 59.62% from FP16 fine-tuning. And, in Qwen3-8B, OTARo reaches 66.13%, 61.51% respectively at E5M4, E5M3, exceeding both fixed precision fine-tuning results (64.82%, 61.05%) and FP16 fine-tuning results (63.62%, 58.60%).

These experimental results highlight the ability of OTARo to maintain robust performance in zero-shot tasks. Detailed results are provided in Appendix C.

Task-Specific Fine-Tuning Performances

As shown in fig. 7, in the task-specific fine-tuning task, OTARo outperforms all baselines, achieving the lowest perplexity in all bit-widths. Detailed data is in Appendix D, where we can see, compared to fixed precision fine-tuning, OTARo obtains an average reduction of 0.16 PPL, and at lower bit-widths, the benefits are more pronounced, with reductions of 0.27 and 0.32 PPL respectively at E5M4 and E5M3. The experimental results confirm the effectiveness of OTARo in task-specific fine-tuning.

Ablation Studies

We perform ablation studies to evaluate the impacts of the BPS and LAA strategies. BPS alone can achieve robustness by exploring multiple bit-widths rather than relying on a fixed one. However, BPS-only fine-tuned models still suffer from low-precision gradient oscillations. By incorporating LAA, the model obtains further performance improvements.

Moreover, we ablate exploration coefficient λ in BPS. A value too large causes an under-use of high precisions, whereas a value too small leads to an insufficient exploration of low precisions. Experiments with $\lambda \in \{3, 4, 5, 6, 7\}$ confirm that λ=5 offers the best balance.

Precisions	Mem. (GB)	Dec. Thpt. (token/s)
FP16	15.20	39.00
SEFP-E5M4	4.77 (69% ↓)	95.65 (×2.45)

Table 2: The memory consumption (Mem.) and decoding throughput (Dec. Thpt.) of FP16 and SEFP LLaMA-8B (E5M4 taken as an example for SEFP).

We also explore different values of delay step N in LAA. N=10 strikes a balance between smoothing gradient oscillations and ensuring sufficient updates, achieving the best performance compared to N=5 and N=20.

Ablation results are shown in fig. 8.

Memory and Speed

We benchmark the memory consumption (including storage spaces for weights and KV cache) and decoding throughput of LLaMA3-8B respectively under FP16 and SEFP data formats, suppose an input of 2000 tokens, to show the efficiency gains from SEFP quantization. As shown in table 2, from FP16 to SEFP, the memory consumption is reduced by 69%, and the speedup of decoding throughput reaches ×2.45. These results highlight the practical advantages of SEFP in enabling light-weight and high-throughput on-device LLMs.

Conclusion

In this paper, we have proposed OTARo, a novel fine-tuning method that enables robust, flexible, and hardware-friendly LLMs at edge. OTARo can obtain one unified model to support all precisions after once fine-tuning. It introduces SEFP quantization to switch precisions flexibly, and performs a complete end-to-end learning workflow that stabilizes the performances across precisions. Experiments demonstrate its effectiveness in multiple LLMs and domains. In summary, this work advances the multi-precision adaptation of on-device LLMs, and provides a crucial methodological support for the real-world edge intelligence.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Amini, A.; Gabriel, S.; Lin, P.; Koncel-Kedziorski, R.; Choi, Y.; and Hajishirzi, H. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*.

Ashkboos, S.; Mohtashami, A.; Croci, M. L.; Li, B.; Cameron, P.; Jaggi, M.; Alistarh, D.; Hoefler, T.; and Hensman, J. 2024. Quarot: Outlier-free 4-bit inference in rotated llms. *Advances in Neural Information Processing Systems*, 37: 100213–100240.

Bisk, Y.; Zellers, R.; Gao, J.; Choi, Y.; et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 7432–7439.

Chen, M.; Shao, W.; Xu, P.; Wang, J.; Gao, P.; Zhang, K.; and Luo, P. 2024. Efficientqat: Efficient quantization-aware training for large language models. *arXiv preprint arXiv:2407.11062*.

Clark, C.; Lee, K.; Chang, M.-W.; Kwiatkowski, T.; Collins, M.; and Toutanova, K. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Clark, P.; Cowhey, I.; Etzioni, O.; Khot, T.; Sabharwal, A.; Schoenick, C.; and Tafjord, O. 2018. Think you have solved question answering? try arc, the ai2 reasoning challenge. *arXiv preprint arXiv:1803.05457*.

Frantar, E.; Ashkboos, S.; Hoefler, T.; and Alistarh, D. 2022. Gptq: Accurate post-training quantization for generative pre-trained transformers. *arXiv preprint arXiv:2210.17323*.

Gao, J.; Shen, J.; Zhang, Y.; Ji, W.; and Huang, H. 2024. Precision-Aware Iterative Algorithms Based on Group-Shared Exponents of Floating-Point Numbers. *arXiv preprint arXiv:2411.04686*.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Kalliojarvi, K.; and Astola, J. 2002. Roundoff errors in block-floating-point systems. *IEEE transactions on signal processing*, 44(4): 783–790.

Lin, J.; Tang, J.; Tang, H.; Yang, S.; Chen, W.-M.; Wang, W.-C.; Xiao, G.; Dang, X.; Gan, C.; and Han, S. 2024. Awq: Activation-aware weight quantization for on-device llm compression and acceleration. *Proceedings of Machine Learning and Systems*, 6: 87–100.

Liu, Z.; Oguz, B.; Zhao, C.; Chang, E.; Stock, P.; Mehdad, Y.; Shi, Y.; Krishnamoorthi, R.; and Chandra, V. 2023. Llm-qat: Data-free quantization aware training for large language models. *arXiv preprint arXiv:2305.17888*.

Liu, Z.; Zhao, C.; Fedorov, I.; Soran, B.; Choudhary, D.; Krishnamoorthi, R.; Chandra, V.; Tian, Y.; and Blankevoort, T. 2024. Spinquant: Llm quantization with learned rotations. *arXiv preprint arXiv:2405.16406*.

Lu, W.; Luu, R. K.; and Buehler, M. J. 2025. Fine-tuning large language models for domain adaptation: Exploration of training strategies, scaling, model merging and synergistic capabilities. *npj Computational Materials*, 11(1): 84.

Manduchi, L.; Pandey, K.; Meister, C.; Bamler, R.; Cotterell, R.; Däubener, S.; Fellenz, S.; Fischer, A.; Gärtner, T.; Kirchler, M.; et al. 2024. On the challenges and opportunities in generative ai. *arXiv preprint arXiv:2403.00025*.

Merity, S.; Xiong, C.; Bradbury, J.; and Socher, R. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Mihaylov, T.; Clark, P.; Khot, T.; and Sabharwal, A. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. *arXiv preprint arXiv:1809.02789*.

Park, Y.; Hyun, J.; Cho, S.; Sim, B.; and Lee, J. W. 2024. Any-precision llm: Low-cost deployment of multiple, different-sized llms. *arXiv preprint arXiv:2402.10517*.

Qiao, Y.; Chen, Z.; Zhang, Y.; Wang, Y.; and Huang, S. 2025. TeLLMe: An Energy-Efficient Ternary LLM Accelerator for Prefilling and Decoding on Edge FPGAs. *arXiv preprint arXiv:2504.16266*.

Qu, G.; Chen, Q.; Wei, W.; Lin, Z.; Chen, X.; and Huang, K. 2025. Mobile edge intelligence for large language models: A contemporary survey. *IEEE Communications Surveys & Tutorials*.

Sakaguchi, K.; Bras, R. L.; Bhagavatula, C.; and Choi, Y. 2021. Winogrande: An adversarial winograd schema challenge at scale. *Communications of the ACM*, 64(9): 99–106.

Tang, J.; Sorokin, R.; Ignasheva, E.; Jensen, G.; Chen, L.; Lee, J.; Kulik, A.; and Grundman, M. 2025. Scaling On-Device GPU Inference for Large Generative Models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 6355–6364.

Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wei, L.; Ma, Z.; Yang, C.; and Yao, Q. 2024. Advances in the neural network quantization: A comprehensive review. *Applied Sciences*, 14(17): 7445.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; et al. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Yang, D.; He, N.; Hu, X.; Yuan, Z.; Yu, J.; Xu, C.; and Jiang, Z. 2024a. Post-training quantization for re-parameterization via coarse & fine weight splitting. *Journal of Systems Architecture*, 147: 103065.

Yang, H.; Zhang, Y.; Xu, J.; Lu, H.; Heng, P. A.; and Lam, W. 2024b. Unveiling the generalization power of fine-tuned large language models. *arXiv preprint arXiv:2403.09162*.

Yin, P.; Lyu, J.; Zhang, S.; Osher, S.; Qi, Y.; and Xin, J. 2019. Understanding straight-through estimator in training activation quantized neural nets. *arXiv preprint arXiv:1903.05662*.

Zellers, R.; Holtzman, A.; Bisk, Y.; Farhadi, A.; and Choi, Y. 2019. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*.

Zhao, Z.; Li, H.; Liu, F.; Lu, Y.; Wang, Z.; Yang, T.; Jiang, L.; and Guan, H. 2025. QUARK: Quantization-Enabled Circuit Sharing for Transformer Acceleration by Exploiting Common Patterns in Nonlinear Operations. arXiv:2511.06767.

Zhou, S.; Wang, S.; Yuan, Z.; Shi, M.; Shang, Y.; and Yang, D. 2025. GSQ-Tuning: Group-Shared Exponents Integer in Fully Quantized Training for LLMs On-Device Fine-tuning. *arXiv preprint arXiv:2502.12913*.

Zubiaga, A. 2024. Natural language processing in the era of large language models. *Front Artif Intell*.

A. Visual $\epsilon(\omega_0)$ of Different Mantissa Bit-Widths

In this section, we illustrate the $\epsilon(\omega_0)$ of different mantissa bit-widths ($m = \{8, 7, 6, 5, 4, 3\}$), as shown in fig. 9.

B. High-Dimensional Linear Mapping

To further investigate the gradient oscillations introduced by low bit-width quantization, we adopt a high-dimensional linear mapping approach. Specifically, we model the relationship between the gradient under SEFP fine-tuning, ∇_{sefp}, and the gradient under FP16 fine-tuning, ∇_{fp}, as the following linear relationship:

$$\nabla_{\text{sefp}} = X \cdot \nabla_{\text{fp}} + Y, \tag{19}$$

where X denotes a linear mapping matrix, and Y represents the residual term that captures additional perturbations introduced by quantization.

To solve for X and Y in the above formulation, we collect the gradient samples from N consecutive batches, which are denoted as:

$$G_{\text{fp}} = \begin{bmatrix} \nabla_{\text{fp},1} \\ \nabla_{\text{fp},2} \\ \vdots \\ \nabla_{\text{fp},N} \end{bmatrix} \in \mathbb{R}^{N \times d}, \quad G = \begin{bmatrix} \nabla_{\text{sefp},1} \\ \nabla_{\text{sefp},2} \\ \vdots \\ \nabla_{\text{sefp},N} \end{bmatrix} \in \mathbb{R}^{N \times d}, \tag{20}$$

where d denotes the dimensionality of the model parameters.

According to the principle of Least Squares Method (LSM), the optimal solution for X minimizes the reconstruction error $\|G - G_{\text{fp}}X\|_F^2$, and its analytical solution is given by:

$$X = (G_{\text{fp}}^\top G_{\text{fp}})^{-1} G_{\text{fp}}^\top G, \tag{21}$$

where $(\cdot)^\top$ denotes matrix transpose, and $\|\cdot\|_F$ represents the Frobenius norm.

After obtaining X, the residual perturbation corresponding to each batch is calculated as:

$$Y = G - G_{\text{fp}}X, \tag{22}$$

that is,

$$Y_i = \nabla_{\text{sefp},i} - X \cdot \nabla_{\text{fp},i}, \quad i = 1, \ldots, N. \tag{23}$$

According to this formulation, Y completely captures the residual stochastic perturbations in the gradients under SEFP quantization that cannot be explained by linear mapping from ∇_{fp}. This enables a more accurate characterization of the non-linear errors introduced by low bit-width quantization.

Compared to the conventional direct differencing method (i.e., $\nabla_{\text{sefp}} - \nabla_{\text{fp}}$), this approach introduces a fitted X term, which effectively eliminates the linear scaling effect caused by the variation in gradient magnitude across batches. Therefore, Y is able to more purely reflect the stochastic and non-stationary nature of the SEFP-induced perturbations.

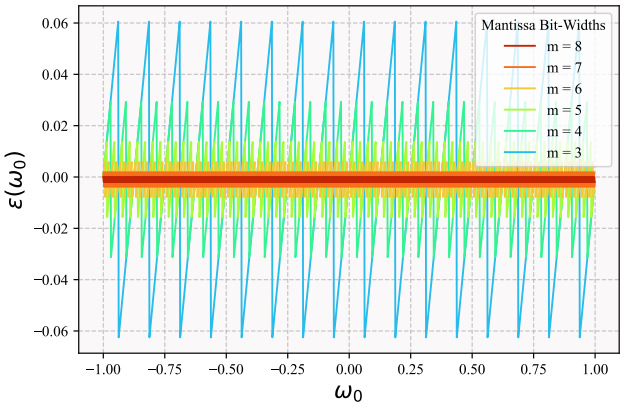


Figure 9: Visualization of $\epsilon(\omega_0)$ function under the mantissa bit-width $m = \{8, 7, 6, 5, 4, 3\}$.

C. Detailed Zero-Shot Results

In zero-shot experiments, we include 5 popular LLMs, i.e., LLaMA2-7B, LLaMA2-13B, LLaMA3-8B, LLaMA3.2-3B, Qwen3-8B. Each model is fine-tuned with Alpaca dataset and tested with ARC-Easy, ARC-Challenge, BoolQ, HellaSwag, MATHQA, OpenBookQA, PIQA, WinoGrande. The detailed zero-shot results of these LLMs are demonstrated in tables 3 to 7.

D. Detailed Results of Task-Specific Fine-Tuning

In task-specific fine-tuning experiments, we fine-tune and evaluate LLaMA3.2-1B using WikiText2 train set and test set, respectively, and report the detailed results before/after fine-tuning in table 8.

Methods	Tasks	E5M8	E5M7	E5M6	E5M5	E5M4	E5M3
Before Fine-Tuning	Arc-c	43.34%	43.34%	43.09%	42.41%	42.75%	43.77%
	Arc-e	76.35%	76.39%	76.18%	76.47%	76.30%	75.25%
	BoolQ	77.74%	77.74%	77.40%	77.06%	78.47%	76.67%
	HellaS.	57.14%	57.20%	57.16%	57.08%	56.90%	56.42%
	MathQA	28.14%	27.77%	28.17%	28.11%	27.97%	27.47%
	OBQA	31.40%	31.40%	32.00%	30.40%	32.80%	30.40%
	PIQA	77.97%	78.29%	77.97%	77.97%	78.02%	77.20%
	WinoG.	69.06%	69.06%	69.14%	69.85%	69.14%	68.75%
	AVG.	**57.64%**	**57.65%**	**57.64%**	**57.42%**	**57.79%**	**56.99%**
FP16 Fine-Tuning	Arc-c	45.31%	45.22%	44.62%	45.14%	44.20%	43.34%
	Arc-e	77.53%	77.36%	77.27%	76.94%	78.16%	74.54%
	BoolQ	77.49%	77.65%	77.46%	76.79%	78.01%	75.87%
	HellaS.	55.77%	55.76%	55.74%	55.43%	55.08%	54.53%
	MathQA	28.71%	28.41%	28.64%	28.84%	27.60%	27.20%
	OBQA	33.20%	33.80%	33.00%	32.40%	31.60%	31.80%
	PIQA	78.02%	78.13%	78.07%	78.13%	78.07%	76.61%
	WinoG.	69.85%	69.85%	70.32%	70.01%	69.85%	67.80%
	AVG.	**58.24%**	**58.27%**	**58.14%**	**57.96%**	**57.82%**	**56.46%**
Fixed Precision Fine-Tuning	Arc-c	45.22%	45.14%	45.14%	44.28%	43.94%	43.07%
	Arc-e	77.57%	77.48%	77.10%	76.52%	76.81%	75.57%
	BoolQ	78.07%	77.40%	77.28%	77.83%	78.59%	75.42%
	HellaS.	55.80%	55.71%	55.77%	55.41%	55.02%	54.60%
	MathQA	28.61%	28.84%	29.15%	29.21%	28.24%	28.21%
	OBQA	33.00%	33.60%	33.00%	34.40%	34.00%	31.30%
	PIQA	78.07%	78.07%	78.24%	77.64%	77.80%	78.26%
	WinoG.	70.17%	70.32%	70.48%	70.32%	70.72%	70.25%
	AVG.	**58.31%**	**58.32%**	**58.27%**	**58.20%**	**58.14%**	**57.09%**
Ours	Arc-c	45.56%	45.95%	46.07%	44.96%	44.03%	44.69%
	Arc-e	77.72%	76.99%	76.85%	76.82%	76.45%	75.80%
	BoolQ	78.93%	78.88%	78.59%	78.59%	78.74%	76.13%
	HellaS.	57.99%	57.70%	57.78%	57.63%	57.61%	56.92%
	MathQA	29.68%	29.88%	29.68%	29.73%	28.59%	28.77%
	OBQA	34.20%	33.90%	33.20%	33.70%	34.80%	31.70%
	PIQA	78.40%	78.56%	78.43%	78.31%	78.13%	77.63%
	WinoG.	70.72%	70.85%	70.80%	70.27%	71.25%	70.15%
	AVG.	**59.15%**	**59.09%**	**58.93%**	**58.75%**	**58.70%**	**57.72%**

Table 3: Detailed zero-shot results of LLaMA2-7B. We report accuracies in Arc-Challenge, Arc-Easy, BoolQ, HellaSwag, MATHQA, OpenBookQA, PIQA, WinoGrande, and the average values of all zero-shot tasks.

Methods	Tasks	E5M8	E5M7	E5M6	E5M5	E5M4	E5M3
Before Fine-Tuning	Arc-c	48.38%	47.95%	47.87%	48.46%	47.78%	47.10%
	Arc-e	79.38%	79.34%	78.96%	79.55%	79.29%	79.00%
	BoolQ	80.58%	80.34%	80.40%	80.40%	79.51%	81.04%
	HellaS.	60.05%	60.06%	60.12%	59.98%	59.62%	58.84%
	MathQA	32.09%	32.06%	32.26%	32.03%	31.49%	31.26%
	OBQA	35.00%	35.00%	35.20%	35.20%	34.60%	34.40%
	PIQA	79.27%	79.00%	79.11%	79.16%	78.94%	79.27%
	WinoG.	72.22%	72.22%	72.14%	72.14%	71.98%	72.38%
	AVG.	**60.87%**	**60.75%**	**60.76%**	**60.87%**	**60.40%**	**60.41%**
FP16 Fine-Tuning	Arc-c	50.68%	50.51%	50.43%	49.57%	49.49%	48.63%
	Arc-e	80.26%	80.09%	80.01%	79.71%	80.05%	78.66%
	BoolQ	81.53%	81.77%	81.62%	81.16%	80.73%	79.54%
	HellaS.	59.58%	59.61%	59.56%	59.04%	59.28%	57.20%
	MathQA	33.53%	33.47%	33.47%	33.17%	32.66%	31.59%
	OBQA	35.60%	35.40%	35.60%	35.80%	35.40%	33.80%
	PIQA	78.73%	78.78%	78.84%	79.11%	78.45%	78.84%
	WinoG.	71.51%	71.74%	71.90%	71.51%	70.64%	71.82%
	AVG.	**61.43%**	**61.42%**	**61.43%**	**61.13%**	**60.84%**	**60.01%**
Fixed Precision Fine-Tuning	Arc-c	50.77%	50.85%	50.51%	49.49%	50.26%	49.91%
	Arc-e	80.18%	79.92%	79.67%	79.38%	79.80%	79.83%
	BoolQ	81.74%	81.38%	81.90%	81.10%	81.35%	78.53%
	HellaS.	59.56%	59.65%	59.52%	59.10%	58.77%	57.58%
	MathQA	33.63%	33.43%	33.63%	33.60%	33.27%	32.90%
	OBQA	36.00%	35.80%	35.80%	35.20%	36.00%	35.62%
	PIQA	78.78%	78.67%	79.05%	79.00%	78.94%	79.16%
	WinoG.	71.67%	71.74%	71.90%	72.45%	71.43%	71.51%
	AVG.	**61.54%**	**61.43%**	**61.50%**	**61.17%**	**61.23%**	**60.63%**
Ours	Arc-c	52.21%	51.93%	51.72%	50.58%	50.16%	50.01%
	Arc-e	81.75%	81.58%	81.55%	80.99%	80.75%	80.66%
	BoolQ	82.95%	82.65%	82.73%	82.00%	82.38%	81.11%
	HellaS.	60.35%	60.16%	59.93%	59.71%	59.71%	58.10%
	MathQA	34.72%	34.49%	34.78%	34.67%	34.51%	33.78%
	OBQA	36.70%	35.90%	35.40%	36.00%	36.40%	36.30%
	PIQA	78.38%	78.38%	78.52%	78.33%	78.66%	78.20%
	WinoG.	71.60%	71.34%	72.24%	72.37%	72.53%	70.64%
	AVG.	**62.33%**	**62.05%**	**62.11%**	**61.83%**	**61.89%**	**61.10%**

Table 4: Detailed zero-shot results of LLaMA2-13B. We report accuracies in Arc-Challenge, Arc-Easy, BoolQ, HellaSwag, MATHQA, OpenBookQA, PIQA, WinoGrande, and the average values of all zero-shot tasks.

Methods	Tasks	E5M8	E5M7	E5M6	E5M5	E5M4	E5M3
Before Fine-Tuning	Arc-c	50.43%	50.34%	50.60%	50.26%	49.15%	45.48%
	Arc-e	80.18%	79.80%	80.35%	80.01%	79.42%	76.73%
	BoolQ	81.22%	80.95%	81.25%	80.46%	80.89%	79.82%
	HellaS.	60.16%	60.13%	60.08%	59.99%	59.89%	57.97%
	MathQA	40.40%	40.74%	40.07%	40.74%	39.33%	36.78%
	OBQA	34.80%	34.60%	34.60%	34.60%	35.60%	32.20%
	PIQA	79.60%	79.54%	79.71%	79.22%	79.11%	77.37%
	WinoG.	72.69%	72.69%	73.24%	73.16%	72.69%	72.14%
	AVG.	**62.44%**	**62.35%**	**62.49%**	**62.31%**	**62.01%**	**59.81%**
FP16 Fine-Tuning	Arc-c	52.22%	52.65%	52.30%	51.54%	49.40%	45.05%
	Arc-e	81.65%	81.90%	81.69%	82.32%	81.31%	76.56%
	BoolQ	82.02%	81.68%	81.87%	81.31%	80.76%	81.19%
	HellaS.	59.07%	58.96%	58.91%	58.72%	57.66%	57.05%
	MathQA	40.94%	40.84%	40.60%	40.70%	37.05%	36.45%
	OBQA	36.80%	37.20%	36.80%	37.40%	32.80%	31.40%
	PIQA	79.60%	79.49%	79.87%	79.54%	78.29%	77.53%
	WinoG.	75.06%	75.30%	75.45%	74.82%	72.53%	71.74%
	AVG.	**63.42%**	**63.50%**	**63.44%**	**63.29%**	**61.23%**	**59.62%**
Fixed Precison Fine-Tuning	Arc-c	52.65%	51.88%	52.13%	51.02%	49.49%	45.31%
	Arc-e	82.11%	82.07%	81.99%	81.52%	80.18%	76.98%
	BoolQ	83.94%	82.78%	83.43%	83.94%	81.62%	80.98%
	HellaS.	59.87%	59.52%	59.51%	59.89%	58.84%	56.69%
	MathQA	40.84%	40.74%	40.13%	40.07%	38.29%	35.91%
	OBQA	36.80%	36.40%	36.00%	36.00%	35.60%	33.20%
	PIQA	80.41%	80.58%	80.25%	80.14%	79.98%	77.91%
	WinoG.	74.51%	75.14%	74.98%	74.11%	74.43%	71.74%
	AVG.	**63.89%**	**63.64%**	**63.55%**	**63.34%**	**62.30%**	**59.84%**
Ours	Arc-c	53.75%	53.84%	53.07%	53.50%	52.56%	48.12%
	Arc-e	81.86%	81.78%	81.94%	82.15%	81.31%	78.07%
	BoolQ	84.28%	84.59%	84.25%	84.07%	83.88%	82.63%
	HellaS.	60.43%	60.51%	60.36%	60.44%	60.34%	57.48%
	MathQA	40.84%	40.30%	40.57%	40.40%	39.87%	36.68%
	OBQA	36.60%	37.40%	37.40%	36.00%	35.40%	32.60%
	PIQA	80.47%	80.30%	80.03%	79.76%	80.47%	78.02%
	WinoG.	74.51%	74.27%	74.27%	74.43%	72.85%	71.74%
	AVG.	**64.09%**	**64.12%**	**63.99%**	**63.84%**	**63.34%**	**60.67%**

Table 5: Detailed zero-shot results of LLaMA3-8B. We report accuracies in Arc-Challenge, Arc-Easy, BoolQ, HellaSwag, MATHQA, OpenBookQA, PIQA, WinoGrande, and the average values of all zero-shot tasks.

Methods	Tasks	E5M8	E5M7	E5M6	E5M5	E5M4	E5M3
Before Fine-Tuning	Arc-c	41.89%	42.15%	42.32%	42.32%	42.66%	37.46%
	Arc-e	74.82%	74.58%	74.58%	74.45%	74.28%	69.07%
	BoolQ	73.46%	73.18%	73.76%	74.04%	70.83%	72.87%
	HellaS.	55.31%	55.29%	55.36%	55.37%	54.55%	52.42%
	MathQA	34.87%	34.47%	34.97%	35.28%	33.80%	33.00%
	OBQA	30.80%	31.00%	31.40%	30.80%	32.20%	28.60%
	PIQA	76.71%	76.66%	76.88%	76.82%	76.71%	76.06%
	WinoG.	69.85%	69.46%	69.85%	69.22%	68.43%	66.69%
	AVG.	**57.21%**	**57.10%**	**57.39%**	**57.29%**	**56.68%**	**54.52%**
FP16 Fine-Tuning	Arc-c	43.09%	43.26%	42.66%	43.60%	42.83%	38.23%
	Arc-e	74.83%	74.49%	74.66%	74.62%	74.54%	69.61%
	BoolQ	74.01%	73.98%	74.71%	74.98%	70.98%	74.59%
	HellaS.	54.42%	54.50%	54.53%	53.40%	53.89%	51.76%
	MathQA	35.58%	35.08%	35.14%	34.37%	34.64%	32.70%
	OBQA	33.60%	34.00%	34.20%	34.20%	31.80%	28.80%
	PIQA	77.15%	76.88%	77.31%	76.61%	77.09%	75.84%
	WinoG.	69.69%	69.77%	70.17%	70.17%	68.43%	65.04%
	AVG.	**57.80%**	**57.75%**	**57.92%**	**57.74%**	**56.78%**	**54.57%**
Fixed Precision Fine-Tuning	Arc-c	43.00%	43.52%	43.00%	42.83%	43.00%	38.31%
	Arc-e	75.29%	75.17%	75.17%	74.83%	74.41%	71.42%
	BoolQ	75.84%	76.21%	76.15%	75.44%	75.78%	74.40%
	HellaS.	54.29%	54.33%	54.30%	54.34%	53.07%	50.07%
	MathQA	35.04%	35.11%	35.58%	35.18%	33.37%	30.92%
	OBQA	32.80%	33.40%	33.60%	33.00%	30.80%	31.40%
	PIQA	77.20%	77.26%	76.99%	77.37%	76.66%	75.19%
	WinoG.	69.69%	69.61%	70.32%	69.77%	68.90%	66.54%
	AVG.	**57.89%**	**58.08%**	**58.14%**	**57.85%**	**57.00%**	**54.78%**
Ours	Arc-c	44.62%	44.62%	44.80%	44.80%	43.77%	42.58%
	Arc-e	75.72%	75.51%	75.93%	75.76%	74.75%	71.72%
	BoolQ	76.61%	76.48%	77.00%	75.69%	76.45%	73.49%
	HellaS.	54.85%	54.86%	54.86%	54.68%	53.55%	51.10%
	MathQA	35.51%	35.01%	35.11%	34.74%	34.30%	33.70%
	OBQA	33.40%	33.20%	34.20%	32.60%	32.80%	30.80%
	PIQA	77.80%	77.97%	77.75%	78.07%	76.61%	75.90%
	WinoG.	69.30%	70.01%	69.46%	69.30%	70.32%	68.98%
	AVG.	**58.48%**	**58.46%**	**58.64%**	**58.21%**	**57.82%**	**56.03%**

Table 6: Detailed zero-shot results of LLaMA3.2-3B. We report accuracies in Arc-Challenge, Arc-Easy, BoolQ, HellaSwag, MATHQA, OpenBookQA, PIQA, WinoGrande, and the average values of all zero-shot tasks.

Methods	Tasks	E5M8	E5M7	E5M6	E5M5	E5M4	E5M3
Before Fine-Tuning	Arc-c	52.90%	52.47%	53.58%	51.11%	53.92%	47.70%
	Arc-e	81.65%	81.69%	82.11%	81.02%	81.52%	76.98%
	BoolQ	83.00%	83.09%	83.82%	80.24%	80.15%	77.86%
	HellaS.	58.91%	58.83%	59.05%	57.88%	56.91%	52.37%
	MathQA	55.54%	55.18%	55.68%	54.61%	54.00%	46.50%
	OBQA	32.40%	32.20%	32.20%	30.80%	31.60%	29.20%
	PIQA	79.00%	78.89%	78.89%	78.94%	77.42%	74.92%
	WinoG.	72.22%	72.69%	72.45%	71.74%	69.53%	63.69%
	AVG.	**64.45%**	**64.38%**	**64.72%**	**63.29%**	**63.13%**	**58.65%**
FP16 Fine-Tuning	Arc-c	57.59%	56.91%	57.68%	54.69%	54.78%	48.46%
	Arc-e	84.05%	84.22%	83.84%	83.33%	81.69%	77.44%
	BoolQ	84.50%	84.34%	84.77%	82.14%	82.94%	76.79%
	HellaS.	58.19%	57.96%	58.19%	57.03%	56.25%	52.44%
	MathQA	57.96%	57.99%	58.89%	57.02%	54.41%	46.23%
	OBQA	32.40%	31.60%	32.00%	32.80%	31.20%	28.20%
	PIQA	79.38%	79.49%	79.11%	79.65%	78.18%	75.19%
	WinoG.	72.45%	72.45%	71.90%	72.93%	69.53%	64.01%
	AVG.	**65.82%**	**65.62%**	**65.80%**	**64.95%**	**63.62%**	**58.60%**
Fixed Precision Fine-Tuning	Arc-c	57.85%	58.36%	57.51%	56.48%	56.31%	54.78%
	Arc-e	84.60%	84.81%	83.96%	83.54%	83.54%	82.28%
	BoolQ	84.25%	84.16%	84.89%	82.02%	84.34%	75.32%
	HellaS.	58.19%	58.23%	58.07%	57.65%	56.94%	54.69%
	MathQA	58.69%	58.73%	58.29%	57.59%	56.52%	47.97%
	OBQA	32.20%	32.00%	32.80%	31.80%	31.40%	29.60%
	PIQA	79.43%	79.76%	79.54%	79.27%	77.80%	77.09%
	WinoG.	72.22%	72.77%	72.22%	72.77%	71.74%	66.69%
	AVG.	**65.93%**	**66.10%**	**65.91%**	**65.14%**	**64.82%**	**61.05%**
Ours	Arc-c	59.73%	59.73%	59.98%	59.04%	59.30%	55.30%
	Arc-e	85.56%	85.61%	85.14%	84.97%	84.64%	83.12%
	BoolQ	86.36%	86.09%	86.12%	84.83%	84.77%	81.35%
	HellaS.	57.84%	57.69%	57.57%	57.28%	57.36%	53.80%
	MathQA	58.59%	58.53%	58.69%	57.79%	57.29%	46.33%
	OBQA	34.00%	33.40%	33.20%	33.00%	33.00%	29.60%
	PIQA	78.94%	78.94%	78.62%	79.43%	79.65%	76.44%
	WinoG.	72.93%	73.16%	73.32%	73.16%	73.01%	66.14%
	AVG.	**66.74%**	**66.64%**	**66.58%**	**66.19%**	**66.13%**	**61.51%**

Table 7: Detailed zero-shot results of Qwen3-8B. We report accuracies in Arc-Challenge, Arc-Easy, BoolQ, HellaSwag, MATHQA, OpenBookQA, PIQA, WinoGrande, and the average values of all zero-shot tasks.

Methods	E5M8	E5M7	E5M6	E5M5	E5M4	E5M3	AVG.	STD.
Before Fine-Tuning	9.75	9.76	9.80	9.95	10.49	13.30	10.51	1.40
FP16 Fine-Tuning	9.17	9.18	9.23	9.46	10.25	12.99	10.05	1.50
Fixed Precision Fine-Tuning	9.08	9.09	9.13	9.30	10.03	12.17	9.80	1.26
Ours	**8.99**	**9.00**	**9.04**	**9.19**	**9.76**	**11.85**	**9.64**	**1.12**

Table 8: Detailed results of LLaMA3.2-1B in task-specific fine-tuning experiments. We report the perplexity (PPL) values in WikiText2 test set, and the average, standard deviation values of all bit-widths.