Quantization Error Propagation: Revisiting Layer-Wise Post-Training Quantization

Yamato Arai
Fujitsu Limited
Department of Basic Science
The University of Tokyo

Yuma Ichikawa
Fujitsu Limited
RIKEN center for AIP

Code: https://github.com/FujitsuResearch/qep

Abstract

Layer-wise PTQ is a promising technique for compressing large language models (LLMs), due to its simplicity and effectiveness without requiring retraining. However, recent progress in this area is saturating, underscoring the need to revisit its core limitations and explore further improvements. We address this challenge by identifying a key limitation of existing layer-wise PTQ methods: the growth of quantization errors across layers significantly degrades performance, particularly in low-bit regimes. To address this fundamental issue, we propose Quantization Error Propagation (QEP), a general, lightweight, and scalable framework that enhances layer-wise PTQ by explicitly propagating quantization errors and compensating for accumulated errors. QEP also offers a tunable propagation mechanism that prevents overfitting and controls computational overhead, enabling the framework to adapt to various architectures and resource budgets. Extensive experiments on several LLMs demonstrate that QEP-enhanced layer-wise PTQ achieves substantially higher accuracy than existing methods. Notably, the gains are most pronounced in the extremely low-bit quantization regime.

1 Introduction

Large Language Models (LLMs) have achieved impressive performance in various natural language processing tasks, including open-ended text generation, multi-step reasoning, and dialogue modeling. Notable examples include ChatGPT [Achiam et al., 2023] and the Llama family [Touvron et al., 2023, Grattafiori et al., 2024]. However, deploying LLMs cost-effectively remains difficult because of their substantial memory usage and computational demands [Chen et al., 2023]. This limitation is especially critical for edge computing and latency-sensitive applications. To address these challenges, a wide range of model compression techniques, such as quantization [Lang et al., 2024, Gong et al., 2024], pruning [Wang et al., 2024, Cheng et al., 2024], low-rank approximation [Yang et al., 2024a, Hu et al., 2022], and knowledge distillation [Xu et al., 2024a, Yang et al., 2024b], have been explored.

Among these methods, layer-wise post-training quantization (PTQ) has emerged as a practical and widely used approach for large-scale LLMs [Frantar et al., 2022, Lin et al., 2024, Yao et al., 2022, Chee et al., 2023]. Unlike block-wise PTQ [Tseng et al., 2024, Shao et al., 2023], global fine-tuning [Egiazarian et al., 2024, Tseng et al., 2024], quantization-aware training (QAT) [Xu et al., 2024b, Wang et al., 2023, Liu et al., 2023], and all of which require heavy retraining and backpropagation, layer-wise PTQ quantizes model parameters layer-by-layer without retraining or backpropagation, resulting in significantly lower computational and memory demands. Despite its simplicity, layer-wise PTQ effectively preserves model quality even at lower bit widths [Frantar et al., 2022, Lin et al., 2024, Chee et al., 2023]. As a result, layer-wise PTQ is increasingly adopted in real-world applications due to its efficient quantization, reduced computational cost, and broader compatibility with large-scale LLMs, varying bit widths, and diverse quantization strategies.

39th Conference on Neural Information Processing Systems (NeurIPS 2025).

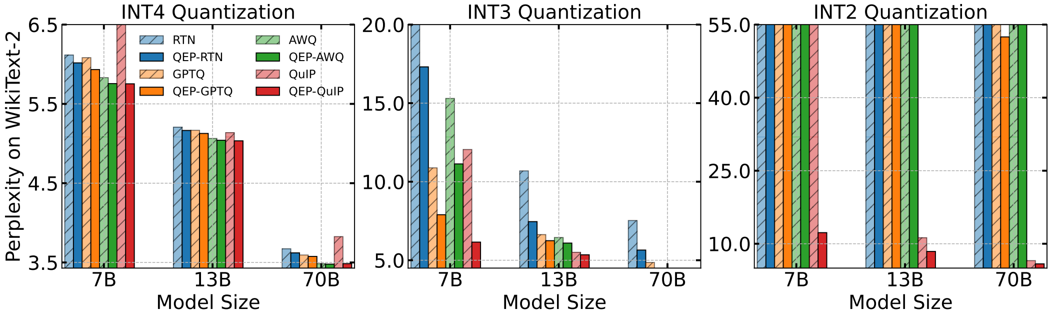

Figure 1: WikiText-2 perplexity comparison across Llama-2 models (7B-70B) quantized to INT-4, INT-3, and INT-2, employing RTN, GPTQ, AWQ, and QuIP methods. Solid bars indicate PTQ with QEP; border bars represent PTQ without QEP. Truncated bars indicate perplexities exceeding axis limits. QEP consistently reduces perplexity, with greater improvements observed at lower bitwidths and smaller model sizes. See Section 6 for detailed settings and results.

Despite significant progress in layer-wise PTQ, advancements in this area are saturating [Malinovskii et al., 2024]. This study aims to push the performance boundaries of layer-wise PTQ by revisiting its core design strategy. This study begins by identifying a fundamental limitation of existing layer-wise PTQ approaches. These approaches do not adequately account for the propagation of quantization errors across layers. Quantization errors accumulate significantly, leading to a degradation in overall model performance, especially in low-bit settings. This represents a key bottleneck for the practical deployment of layer-wise PTQ in large-scale LLMs.

To address this issue, we propose **Q**uantization **E**rror **P**ropagation (**QEP**), a general and computationally efficient framework that enhances the performance of layer-wise PTQ methods. QEP modifies the layer-wise optimization objective to propagate and compensate for accumulated quantization errors, while maintaining computational complexity comparable to existing layer-wise PTQ methods. Furthermore, we introduce a tunable propagation mechanism whose adjustable propagation strength prevents overfitting, a known issue previously observed in GPTQ [Lin et al., 2024]. This mechanism also enables adaptive control over computational overhead, especially in parameter-heavy components such as MLP blocks. Notably, the enhancement of QEP is orthogonal to existing PTQ methods and can be seamlessly integrated with any layer-wise PTQ pipeline.

Extensive experiments on several LLMs across various bit-width settings show that QEP significantly enhances layer-wise PTQ methods, including GPTQ [Frantar et al., 2022], AWQ [Lin et al., 2024], QuIP [Chee et al., 2023], as shown in Figure 1. These improvements are particularly pronounced in extreme low-bit regimes, such as 2-bit quantization, where standard layer-wise PTQ methods typically degrade significantly.

2 Related Work

Quantization techniques primarily include data-free PTQ [Dettmers and Zettlemoyer, 2023], layer-wise PTQ [Frantar et al., 2022, Lin et al., 2024, Chee et al., 2023], block-wise PTQ [Tseng et al., 2024, Shao et al., 2023], global fine-tuning PTQ [Egiazarian et al., 2024, Tseng et al., 2024], and QAT [Xu et al., 2024b, Wang et al., 2023, Liu et al., 2023]. Among these methods, weight-only layer-wise PTQ has become especially popular for large-scale models because of its computational efficiency and strong performance [Frantar et al., 2022, Lin et al., 2024, Chee et al., 2023]. Recent benchmarking further highlights that most PTQ advances specifically target layer-wise methods [Zhao et al., 2025]. Following the taxonomy in [Zhao et al., 2025], we outline three distinct approaches and recent developments.

Compensation-based layer-wise PTQ This category, pioneered by GPTQ [Frantar et al., 2022], uses a sequential quantization strategy, in which model weights are quantized based on the Hessian computed from a calibration dataset, while compensating for subsequent unquantized weights. Several studies refined the compensation mechanism by improving update rules [Behdin et al., 2023], integrating nonlinear quantization schemes [Liu et al., 2024a], employing adaptive grid selection [Zhang and Shrivastava, 2024], and using block-wise optimization [Guan et al., 2024].

Rotation-based layer-wise PTQ A second promising direction, advanced by QuIP [Chee et al., 2023], involves preprocessing weights through structured rotation matrices to more uniformly redistribute weight magnitudes. This approach was improved by randomized Hadamard transforms and block-wise and global fine-tuning optimization [Tseng et al., 2024]. Learning-based methods to determine rotation matrices have also been introduced [Liu et al., 2024b]. This rotation-based strategy has also been extended to activation quantization [Ashkboos et al., 2024].

Salience-based layer-wise PTQ Other approaches focus on identifying and preserving *salient weights*, often using mixed-precision quantization frameworks [Dettmers et al., 2022, 2023, Shang et al., 2023]. Although mixed-precision methods usually add complexity due to various data types, AWQ [Lin et al., 2024] mitigates these implementation difficulties. AWQ strategically employs a global scaling mechanism to align salient weights with the quantization grid better, simplifying deployment while maintaining high accuracy.

Recent advances in layer-wise PTQ have mainly focused on nonlinear quantization and block-wise and global fine-tuning extensions; however, the fundamental layer-wise optimization has remained largely unchanged since GPTQ [Frantar et al., 2022]. This study revisits this foundational strategy, identifies its key limitations, and proposes improvements, demonstrating performance gains on the fundamental benchmarks such as GPTQ [Frantar et al., 2022], QuIP [Chee et al., 2023], and AWQ [Lin et al., 2024]. Therefore, our contributions complement and are orthogonal to recent advancements, such as nonlinear quantization and structured extensions.

3 Background

Post-training quantization Post-training quantization (PTQ) is a technique that converts the parameters of pre-trained models into discrete quantized representations. Formally, let $\boldsymbol{W}_l \in \mathbb{R}^{n_l \times d_l}$ denote the pre-trained weight matrix associated with the l-th linear operation. Note that the index l specifically refers to individual linear transformations rather than entire transformer blocks. The objective of PTQ is to find a quantized approximation $\widehat{\boldsymbol{W}}_l \in \mathbb{Q}^{n_l \times d_l}$ that closely approximates the behavior of the original model, preserving performance while reducing computational costs and memory usage. The set $\mathbb{Q} \subset \mathbb{R}$ denotes the discrete quantization domain, which is represented as a finite set of 2^b distinct quantization levels, referred to as a b-bit quantization scheme. To achieve accurate quantization, many approaches leverage a small calibration dataset. Specifically, given a calibration dataset $\boldsymbol{X} \in \mathbb{R}^{d_1 \times m}$ consisting of m samples, these methods aim to find optimal quantized parameters $\widehat{\boldsymbol{W}}_l$ that minimizes the deviation from the performance of the original model.

Layer-wise PTQ Layer-wise PTQ has emerged as a promising framework [Frantar et al., 2022, Frantar and Alistarh, 2022] for compressing large-scale LLMs. Recent advancements in this area have significantly reduced the computational overhead and memory requirements of deploying LLMs. Despite methodological differences, existing layer-wise PTQ approaches typically follow a shared sequential quantization scheme, processing each layer independently and sequentially from the input layer toward the output layer.

Formally, these techniques quantize the model parameters $\{\boldsymbol{W}_l\}_{l=1}^{L}$ by solving the following layer-wise *independent* optimization problem:

$$\min_{\widehat{\boldsymbol{W}}_l \in \mathbb{Q}^{n_l \times d_l}} \left\| \boldsymbol{W}_l \mathsf{X}_l - \widehat{\boldsymbol{W}}_l \mathsf{X}_l \right\|_F^2 , \tag{1}$$

where X_l denotes the input activations to the l-th layer. This quantization proceeds sequentially from $l = 1$ toward the output layers. Due to the quadratic form of the reconstruction objective, the associated Hessian, $\mathsf{H}_l := \mathsf{X}_l \mathsf{X}_l^\top$, can be efficiently precomputed and cached for reuse in subsequent optimization steps, improving computational efficiency in practice.

Existing PTQ methods typically use one of two possible forms for activation inputs X_l: Either quantized activations $\boldsymbol{X}_l$, obtained by forward propagating the calibration dataset through previously quantized weights $\{\widehat{\boldsymbol{W}}_1, \dots, \widehat{\boldsymbol{W}}_{l-1}\}$, or full-precision activations $\boldsymbol{X}_l$, resulting from forward propagation through the original, unquantized weights $\{\boldsymbol{W}_1, \dots, \boldsymbol{W}_{l-1}\}$. There is no consensus among existing PTQ methods [Frantar et al., 2022, Lin et al., 2024, Chee et al., 2023] regarding whether quantized or full-precision activations produce better quantization outcomes.

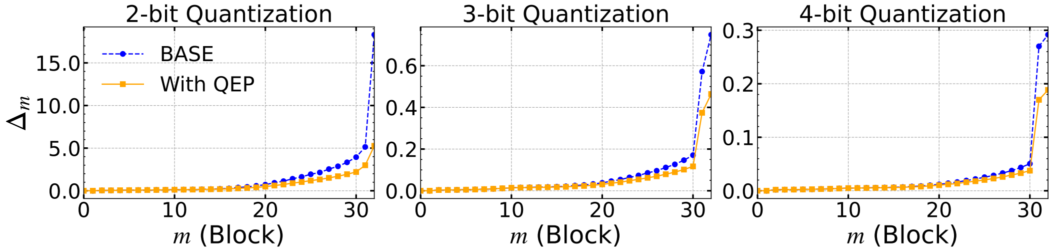

Figure 2: Accumulation and growth of quantization errors across layers in a partially quantized Llama2-7B model [Touvron et al., 2023]. The first 10 Transformer blocks are quantized using standard RTN (BASE) and QEP-enhanced RTN (With QEP), while the remaining Transformer blocks after the 10th remain at full precision. The plot shows the squared Frobenius norm Δ_m, defined in Eq. (2), between the original and partially quantized outputs at each Transformer block m.

Leading layer-wise PTQ methods use distinct optimization strategies to approximate the behavior of the original model while adhering to the foundational sequential layer-wise framework in Eq. (1). GPTQ [Frantar et al., 2022], for example, uses quantized activations, $\mathsf{X}_l = \widehat{\boldsymbol{X}}_l$, and quantizes parameters row-wise by sequentially minimizing reconstruction error and correcting residuals in the remaining unquantized entries until each row is fully quantized. AWQ [Lin et al., 2024] uses original activations, $\mathsf{X}_l = \boldsymbol{X}_l$, and identifies a small subset of *salient weights* whose magnitudes significantly influence the layer outputs, subsequently rescaling these weights before quantization.

4 Bottleneck: Quantization Error Accumulation and Growth

To motivate our proposed approach, this section first revisits the core layer-wise optimization formulation given by Eq. (1), emphasizing its key limitation: The *accumulation* and *growth* of quantization errors across layers significantly degrade the performance. We investigate this phenomenon using experiments conducted on the pre-trained Llama-2-7B model [Touvron et al., 2023]. Specifically, we quantize only the first 10 Transformer blocks [Vaswani et al., 2017], while keeping all subsequent blocks in full precision. To quantify the propagation and accumulation of the errors, we measure the discrepancy between fully precise and partially quantized outputs at each block using a calibration dataset. Let $\mathtt{TransBlock}_m(\cdot)$ denote the original full-precision m-th Transformer block, and $\widehat{\mathtt{TransBlock}}_m(\cdot)$ denote its quantized counterpart. We evaluate the following metric at the m-th block:

$$\Delta_m = \left\| f_m(\boldsymbol{X}) - \widehat{f}_m(\boldsymbol{X}) \right\|_F^2 , \tag{2}$$

$$f_m(\boldsymbol{X}) \coloneqq \mathtt{TransBlock}_m \circ \cdots \circ \mathtt{TransBlock}_{n+1} \circ \mathtt{TransBlock}_n \circ \cdots \circ \mathtt{TransBlock}_1(\boldsymbol{X}),$$
$$\widehat{f}_m(\boldsymbol{X}) \coloneqq \mathtt{TransBlock}_m \circ \cdots \circ \mathtt{TransBlock}_{n+1} \circ \widehat{\mathtt{TransBlock}}_n \circ \cdots \circ \widehat{\mathtt{TransBlock}}_1(\boldsymbol{X}).$$

This experiment sets $n = 10$. Figure 2 shows an approximately exponential *accumulation* or errors within the quantized layer, as well as an error *growth* that persists in the unquantized layers. This *growth* occurs due to the layer-wise *independent* quantization approach described in Eq. (1), which neither accounts for quantization error propagated from previous layers nor corrects previously accumulated errors, thus exacerbating error growth in subsequent unquantized layers. The exponential accumulation of quantization errors observed empirically can also be theoretically explained under mild conditions, as detailed in Appendix B.2. Therefore, instead of treating layer-wise quantization as a series of independent optimization problems, it is essential to reformulate the original layer-wise optimization presented in Eq. (1) to mitigate error accumulation and growth.

5 QEP: Quantization Error Propagation

Existing layer-wise independent PTQ has inherent limitations, particularly the *accumulation* and *growth* of quantization errors discussed in Section4. To address these limitations, we introduce **Q**uantization **E**rror **P**ropagation (**QEP**), a general, lightweight, and scalable framework that improves layer-wise PTQ by propagating quantization errors. In subsequent sections, we provide theoretical evidence showing that QEP effectively reduces quantization errors.

5.1 Problem Reformulation

We reformulate the layer-wise *independent* optimization strategy presented in Eq. (1) to propagate quantization errors across layers effectively. Instead of minimizing output differences based on shared input activations X_l, our reformulation directly minimizes the discrepancy between full-precision and quantized outputs, each computed using their respective upstream inputs. Formally, for each layer l, we optimize the discrete quantized weight matrix $\widehat{\boldsymbol{W}}_l$ as follows:

$$\min_{\widehat{\boldsymbol{W}}_l \in \mathbb{Q}^{n_l \times d_l}} \left\| \boldsymbol{W}_l \boldsymbol{X}_l - \widehat{\boldsymbol{W}}_l \widehat{\boldsymbol{X}}_l \right\|_F^2 . \tag{3}$$

This objective ensures that the quantized weights $\widehat{\boldsymbol{W}}_l$ are optimized not only to independently approximate the full-precision weights $\boldsymbol{W}_l$ but also to counteract and compensate for the cumulative quantization errors introduced by previous layers. In contrast to the existing objective in Eq. (1), where the trivial optimal solution is $\widehat{\boldsymbol{W}}_l = \boldsymbol{W}_l$ if $\boldsymbol{W}_l \in \mathbb{Q}^{n_l \times d_l}$, the optimal solution under the formulation in Eq. (3) is generally $\widehat{\boldsymbol{W}}_l \neq \boldsymbol{W}_l$, explicitly enabling error correction and accounting for accumulated quantization errors.

Although the modification from Eq. (1) seems straightforward, Eq. (3) inherently breaks the key structural simplification that facilitates efficient quantization in existing PTQ frameworks. Specifically, the optimization in Eq. (3) no longer solely depends on the Hessian matrix H_l, thereby preventing the direct use of existing Hessian-based acceleration methods for quantization. In the following section, we address this challenge by proposing a practical and efficient weight correction scheme that overcomes this limitation while retaining the advantages of our error-propagation approach.

5.2 Weight Correction

To efficiently perform quantization by the objective in Eq. (3) as in existing layer-wise PTQ methods, we relax the discrete feasible set to a continuous domain, leading to the following proposition.

Proposition 5.1. *Assume that the matrix $\widehat{\boldsymbol{H}}_l$ is invertible. Then, after relaxing the discrete feasible set $\mathbb{Q}^{n_l \times d_l}$ into the continuous domain $\mathbb{R}^{n_l \times d_l}$, the optimal solution $\boldsymbol{W}_l^*$ is given by the following closed-form expression:*

$$\boldsymbol{W}_l^* \coloneqq \boldsymbol{W}_l + \boldsymbol{W}_l \boldsymbol{\delta}_l \widehat{\boldsymbol{X}}_l^\top \widehat{\boldsymbol{H}}_l^{-1} = \operatorname*{argmin}_{\widehat{\boldsymbol{W}}_l \in \mathbb{R}^{n_l \times d_l}} \left\| \boldsymbol{W}_l \boldsymbol{X}_l - \widehat{\boldsymbol{W}}_l \widehat{\boldsymbol{X}}_l \right\|_F^2 , \tag{4}$$

where $\boldsymbol{\delta}_l \coloneqq \boldsymbol{X}_l - \widehat{\boldsymbol{X}}_l$ represents the accumulated quantization error from proceeding layers, $\widehat{\boldsymbol{H}}_l \coloneqq \widehat{\boldsymbol{X}}_l \widehat{\boldsymbol{X}}_l^\top$ denotes the empirical Hessian constructed from quantized activations.

The proof of Proposition 5.1 is provided in Appendix B.1. Proposition 5.1 highlights an important distinction from the existing formulation given by Eq. (1). Specifically, when upstream quantization introduces non-negligible errors, i.e., $\boldsymbol{\delta}_{l-1} \neq 0$, the optimal quantized weights differ from straightforward approximations of the original weights $\boldsymbol{W}_l$. Instead, the optimal solution explicitly includes a correction term that compensating for accumulated quantization errors.

This corrected weight enables us to reformulate the equivalent optimization objective within the original discrete set $\widehat{\boldsymbol{W}}_l \in \mathbb{Q}^{n_l \times d_l}$ as follows:

$$\min_{\widehat{\boldsymbol{W}}_l \in \mathbb{Q}^{n_l \times d_l}} \left\| \boldsymbol{W}_l^* \widehat{\boldsymbol{X}}_l - \widehat{\boldsymbol{W}}_l \widehat{\boldsymbol{X}}_l \right\|_F^2 . \tag{5}$$

This objective shares the same structure as Eq. (1), with $\boldsymbol{W}_l$ replaced by its corrected counterpart $\boldsymbol{W}_l^*$. This reformulation restores the quadratic structure found in Eq. (1), facilitating efficient optimization through the Hessian matrix $\mathsf{H}_l = \widehat{\boldsymbol{H}}_l$. The structure of Eq. (5) allows for seamless integration with various existing layer-wise PTQ methods, as discussed in Section2. Furthermore, the proposed layer-wise quantization formulation in Eq. (3) formally guarantees improved quantization accuracy compared to the existing layer-wise *independent* PTQ defined in Eq. (1). Specifically, we establish the following theoretical result:

Theorem 5.2 (Informal)**.** *Consider an L-layer neural network defined by:*

$$f_{\boldsymbol{\theta}}(X) = \sigma_L(\boldsymbol{W}_L \sigma_{L-1}(\boldsymbol{W}_{L-1} \cdots \sigma_2(\boldsymbol{W}_2 \sigma_1(\boldsymbol{W}_1 \boldsymbol{X})) \cdots)),$$

where each activation function σ_l is Lipschitz continuous and $\boldsymbol{\theta}$ denotes the set of all full-precision parameters $\{\boldsymbol{W}_l\}_{l=1}^{L}$. The output quantization error of the proposed quantization method defined in Eq. (3) *is bounded by that of the existing layer-wise PTQ defined in Eq.* (1)*:*

$$\left\| f_{\theta}(\boldsymbol{X}) - f_{\widehat{\boldsymbol{\theta}}_{\mathrm{QEP}}}(\boldsymbol{X}) \right\|_F \leq \left\| f_{\boldsymbol{\theta}}(\boldsymbol{X}) - f_{\widehat{\boldsymbol{\theta}}_{\mathrm{BASE}}}(\boldsymbol{X}) \right\|_F .$$

where $\widehat{\boldsymbol{\theta}}_{\mathrm{QEP}}$ and $\widehat{\boldsymbol{\theta}}_{\mathrm{BASE}}$ denote the sets of parameters quantized by the objective in Eq. (3) *and the base PTQ method by the objective in Eq.* (1)*, respectively.*

Explicit conditions and detailed proof are provided in Appendix B.3. The additional computational overhead arises solely from computing the correction term $\boldsymbol{\delta}_l \widehat{\boldsymbol{X}_l}^\top$, since computing the Hessian inverse $\widehat{\boldsymbol{H}_l}^{-1}$ remains unchanged from existing layer-wise *independent* PTQ. As empirically demonstrated in Section 6.1, this additional computation requires significantly less runtime compared to the quantization processes of layer-wise PTQ methods, even for large-scale LLMs, due to the tunable mechanism described in the next section.

5.3 Controlling Propagation Strength

Although solving Eq. (5) effectively reduces the accumulation of quantization error, it can lead to overfitting. This issue is particularly pronounced when the calibration dataset is small and insufficiently representative of the target task, or when the model includes blocks with a large number of parameters such as the MLP blocks commonly found in transformer architectures, causing the correction to overfit the calibration dataset.

To address this issue, we introduce a tunable propagation mechanism that generalizes the correction term using a scaling parameter $\alpha_l \in [0, 1]$:

$$\boldsymbol{W}_l^*(\alpha_l) = \boldsymbol{W}_l + \alpha_l \boldsymbol{W}_l \boldsymbol{\delta}_l \widehat{\boldsymbol{X}_l}^\top \widehat{\boldsymbol{H}_l}^{-1}. \tag{6}$$

Here, setting $\alpha_l = 1$ recovers original fully-corrected case presented in Eq. (4), whereas setting $\alpha_l = 0$ corresponds to the existing approach in Eq. (1) under the setting that $\mathsf{X}_l = \widehat{\boldsymbol{X}}_l$. This tunable correction mechanism relates to the following regularization optimization:

Proposition 5.3. *The parameter α_l corresponds to the regularization parameter λ in the following optimization problem:*

$$\min_{\widehat{\boldsymbol{W}}_l \in \mathbb{Q}^{n_l \times d_l}} \|\boldsymbol{W}_l \boldsymbol{X}_l - \widehat{\boldsymbol{W}}_l \widehat{\boldsymbol{X}}_l\|_F^2 + \lambda_l \|\boldsymbol{W}_l - \widehat{\boldsymbol{W}}_l\|_F^2, \ \ \lambda_l \in \mathbb{R}_+.$$

Specifically, as α_l increases from 0 to 1, the corresponding parameter λ_l decreases from $+\infty$ to 0.

The derivation is provided in Appendix B.4. Additionally, the following proposition is established.

Proposition 5.4. *Under the same assumptions in Theorem 5.2, the output quantization error of the method employing QEP with parameter $\{\alpha_l\}_{l=1}^{L}$ decreases monotonically as each α_l approaches 1.*

Explicit conditions and comprehensive proofs of this proposition are provided in Appendix B.3. Consequently, the parameter α_l effectively controls overfitting, analogous to regularization techniques, and importantly provides a systematic way to balance overfitting and underfitting in layer-wise PTQ methods. Indeed, this parameter is crucial for preventing overfitting, especially in MLP blocks, which contain more parameters than other blocks.

Furthermore, in large-scale LLMs, the high-dimensional activations in MLP layers often result in computationally expensive correction terms. In these cases, selectively setting $\alpha_l = 0$ for specific layers eliminates the computational cost of the correction term and acts as implicit regularization, potentially improving generalization. Therefore, appropriately setting $\alpha_l = 0$ can reduce the correction time by approximately one-third and one-half. Developing adaptive strategies for layer-wise, data-aware, or resource-efficient tuning of α_l is a promising direction for future research. In the following, we refer to the overall approach, including the tunable mechanism controlled by $\{\alpha_l\}_{l=1}^{L}$, as **Q**uantization **E**rror **P**ropagation (**QEP**).

6 Experiments

We conduct experiments to validate the effectiveness of QEP in improving the performance of layer-wise PTQ relative to existing methods.

Baselines We use representative layer-wise PTQ methods based on linear quantization such as round-to-nearest (RTN) [Frantar et al., 2022, Dettmers and Zettlemoyer, 2023], GPTQ [Frantar et al., 2022], AWQ [Lin et al., 2024], and QuIP [Chee et al., 2023]. Although previous studies have explored extensions, such as non-linear and block-wise quantization, as discussed in Section 2, these techniques are orthogonal to the core improvement introduced by QEP. Therefore, to isolate and emphasize the impact of QEP, we focus on these representative layer-wise PTQ methods.

Quantization This study focuses on weight-only quantization schemes, specifically per-channel and group-wise quantization, which have recently shown superior trade-offs between efficiency and accuracy [Dettmers and Zettlemoyer, 2023, Frantar et al., 2022, Lin et al., 2024]. The main text evaluates per-channel quantization under INT4, INT3, and INT2 precision settings. Due to space constraints, detailed results for group-wise quantization are presented in Appendix D. For the propagation strength parameter α_l, we adopt a representative default value of $\alpha_l = 1/2$ for all layers, except for the MLP layers in the Llama-2 70B model, for which we set $\alpha_l = 0$. Tuning α_l can further improve performance but is beyond the scope of this study and is left for future work.

Datasets Following previous studies, we evaluate the Hessian matrix using the same default calibration datasets used in their original implementations. Specifically, GPTQ and QuIP use the C4 dataset [Frantar et al., 2022] for calibration, while AWQ uses the Pile dataset [Gao et al., 2020]. Following Frantar et al. [2022], we evaluate the correction term in Eq. (4) using 128 randomly sampled segments of 2048 tokens each from the C4 dataset[Raffel et al., 2020], which consists of web-crawled text excerpts.

Models Following Lin et al. [2024], Frantar et al. [2022], we evaluate our method on recent popular LLMs, namely the Llama-2 and Llama-3 model families [Touvron et al., 2023], with size ranging from 7B to 70B parameters, as well as Mistral-7 B [Jiang, 2024]. These models demonstrate superior performance compared to other open-source LLMs [Zhang et al., 2022, Workshop et al., 2022] and have become widely adopted as foundational models for numerous derivative open-source models [Taori et al., 2023, Chiang et al., 2023].

Evaluations Following established evaluation protocols from prior studies [Dettmers et al., 2022, Xiao et al., 2023, Frantar et al., 2022, Dettmers and Zettlemoyer, 2023, Yao et al., 2022], we evaluate the quantized LLMs using the perplexity (PPL) on WikiText2 [Merity et al., 2016], Penn Treebank (PTB) [Marcus et al., 1994], and C4 [Raffel et al., 2020], and zero-shot accuracy on benchmarks including ARC Easy (ArcE) [Boratko et al., 2018], PiQA [Bisk et al., 2020], and StoryCloze (SC) [Mostafazadeh et al., 2016]. Due to space limitations, detailed results for each dataset are provided in Appendix D. All experiments are conducted using a single NVIDIA V100 GPU.

6.1 Results

Perplexity Table 1 summarizes PPL results of various quantized models evaluated on WikiText2, comparing several bit-widths and different layer-wise PTQ methods, both with and without QEP. Additional C4 and PTB dataset results are provided in Appendix D.1, demonstrating consistent trends in the following. Our results indicate that incorporating QEP significantly enhances the performance of layer-wise PTQ, substantially reducing perplexity across nearly all tested methods and quantization levels. In medium-bit scenarios such as INT4 and INT3, where AWQ already exhibits strong performance, applying QEP yields further improvements. At 2-bit quantization, existing layer-wise PTQ methods based on linear quantization typically suffer severe PPL degradation, rendering deployment infeasible. However, QEP effectively mitigates this issue, making INT2 quantization achievable with practical perplexity levels. Notably, QEP-enhanced QuIP achieves state-of-the-art perplexity results among all tested layer-wise PTQ methods. Similar significant improvements are observed for RTN, GPTQ, and AWQ at INT2g32, INT2g64, and INT2g128 quantization levels; see Appendix D.1 for details.

Table 1: Evaluation of perplexities (↓) for Llama models on WikiText-2 under various layer-wise PTQ methods and bitwidths.

Bits	Method	QEP	Llama-2-7B	Llama-2-13B	Llama-2-70B	Llama-3-8B	Mistral-7B
FP16	-	-	5.472	4.883	3.319	6.137	5.255
INT4	RTN	✗	6.116	5.206	3.672	8.540	5.997
		✓	**6.017**	**5.165**	**3.621**	**8.021**	**5.877**
	GPTQ	✗	6.083	5.167	3.594	147.912	5.643
		✓	**5.933**	**5.127**	**3.576**	**9.509**	**5.528**
	AWQ	✗	5.831	5.064	3.484	7.108	5.716
		✓	**5.756**	**5.041**	**3.479**	**6.981**	**5.636**
	QuIP	✗	8.434	5.137	3.826	6.998	11.109
		✓	**5.753**	**5.034**	**3.485**	**6.650**	**5.479**
INT3	RTN	✗	539.866	10.688	7.530	2276.227	29.390
		✓	**17.309**	**7.458**	**5.648**	**86.430**	**10.241**
	GPTQ	✗	10.881	6.632	4.860	64.457	8.247
		✓	**7.898**	**6.245**	**4.102**	**18.845**	**7.347**
	AWQ	✗	15.299	6.448	4.362	11.802	7.902
		✓	**11.131**	**6.092**	**4.103**	**10.713**	**7.169**
	QuIP	✗	12.048	5.503	4.135	8.288	7.108
		✓	**6.154**	**5.347**	**3.813**	**7.703**	**5.842**
INT2	RTN	✗	**17783.918**	**51152.832**	26077.172	1437176.750	78488.328
		✓	97153.266	61158.555	**26063.672**	**554142.313**	**50540.059**
	GPTQ	✗	13051.469	**1301.395**	107.458	**236596.891**	3543.708
		✓	**7214.328**	2782.353	**52.472**	282245.188	**1665.287**
	AWQ	✗	**199448.797**	93036.517	**81834.344**	1044956.250	**31391.543**
		✓	229888.406	**74735.836**	88684.156	**639158.313**	32668.666
	QuIP	✗	65.593	11.232	6.536	70.518	26.632
		✓	**11.972**	**8.417**	**5.869**	**27.326**	**9.586**

Zero-shot tasks We evaluate the zero-shot accuracy of quantized models on several tasks. Table 2 summarizes the average accuracy for the ArcE, PiQA, and SC datasets. Detailed results for each dataset are provided in Appendix D.2. Consistent with the perplexity results, QEP effectively improves existing layer-wise PTQ methods. Notably, the performance gains from QEP are especially pronounced with INT2 quantization. For Llama-2-70B, the QEP-enhanced QuIP at INT2 achieves performance comparable to RTN and GPTQ at INT3 quantization.

Runtime We examine the impact of computation time required for the correction term. Table 3 shows the processing time of each layer-wise PTQ. Since the quantization processing time for RTN is only a few seconds and thus negligible, the measured time for QEP+RTN is primarily due to computing the preprocessing of the correction term. This result indicates that calculating the QEP correction term requires significantly less computation time than other existing layer-wise PTQ quantization processes. Moreover, using the same calibration dataset for weight correction and quantization reduces preprocessing overhead by approximately one-half to one-third by reusing computational steps.

Table 3: Runtime comparison of the quantization process.

Runtime	Llama-2		
	7B	13B	70B
GPTQ	14.9m	26.4m	2.9h
AWQ	13.6m	25.4m	2.4h
QEP + RTN	10.9m	19.6m	1.7h

Table 4: Perplexity relative to RTN on WikiText2, comparing GPTQ and QEP+RTN calibrated with C4, PTB, and WikiText2.

PPL to RTN (↓)	Calibration Dataset		
	C4	PTB	WikiText2
GPTQ	-0.25	+0.07	-0.46
QEP + RTN	**-0.33**	**-0.30**	**-0.49**

Robustness As discussed in Section 5.3, our method adaptively controls propagation strength in Eq. (6) to mitigate overfitting to the calibration dataset. In this section, we empirically validate this approach. Table 4 compares the perplexity difference among QEP-enhanced RTN, GPTQ, and RTN when quantizing Llama-2-7B, evaluated on Wiki-

Table 2: Zero-shot average accuracy (↑) on ARC-Easy, PIQA, and StoryCloze for Llama models across three quantization settings.

Bits	Method	QEP	Llama-2-7B	Llama-2-13B	Llama-2-70B	Llama-3-8B	Mistral-7B
FP16	-	-	0.7601	0.7840	0.8014	0.7920	0.8056
INT4	RTN	✗	0.6802	**0.7160**	0.7325	0.7643	0.7831
		✓	**0.6844**	0.7131	**0.7343**	**0.7686**	**0.7921**
	GPTQ	✗	**0.6817**	**0.7134**	0.7306	0.4812	**0.7906**
		✓	0.6795	0.7104	**0.7308**	**0.7531**	0.7904
	AWQ	✗	0.6832	0.7120	0.7257	0.7821	0.7956
		✓	**0.6870**	**0.7126**	**0.7331**	**0.7879**	**0.7967**
	QuIP	✗	0.6500	**0.7248**	0.7285	**0.7872**	0.7204
		✓	**0.6920**	0.7167	**0.7311**	0.7800	**0.8012**
INT3	RTN	✗	0.4770	0.6082	0.6402	0.4560	0.6448
		✓	**0.5802**	**0.6550**	**0.6939**	**0.5388**	**0.6963**
	GPTQ	✗	0.6367	0.6747	0.7043	0.4891	0.7305
		✓	**0.6549**	**0.6853**	**0.7078**	**0.5901**	**0.7422**
	AWQ	✗	0.5840	0.6886	0.7209	0.7074	0.7534
		✓	**0.6264**	**0.6916**	**0.7283**	**0.7216**	**0.7675**
	QuIP	✗	0.6232	0.7034	0.7246	0.7433	0.7422
		✓	**0.6804**	**0.7128**	**0.7273**	**0.7549**	**0.7933**
INT2	RTN	✗	0.4139	**0.4283**	0.4147	**0.4183**	**0.4130**
		✓	**0.4199**	0.4191	**0.4145**	0.4108	0.4084
	GPTQ	✗	0.4162	0.4222	0.4356	0.4116	**0.4159**
		✓	**0.4263**	**0.4283**	**0.4714**	**0.4228**	0.4148
	AWQ	✗	**0.4213**	**0.4176**	0.4129	**0.4164**	0.4177
		✓	0.4162	0.4165	**0.4140**	0.4150	**0.4181**
	QuIP	✗	0.4667	0.5945	0.6628	0.4600	0.5422
		✓	**0.5926**	**0.6404**	**0.6998**	**0.5121**	**0.6858**

Text2 across various calibration datasets. Consistent with prior findings [Lin et al., 2024], GPTQ exhibits significant sensitivity to the calibration dataset: it outperforms RTN on C4 and WikiText2 but experiences notable performance degradation on PTB. In contrast, QEP-enhanced RTN consistently improves performance across all calibration datasets, demonstrating robustness to distributional shifts. This highlights the effectiveness of propagation control in preventing overfitting to the calibration dataset.

7 Conclusion

We revisit the core design of layer-wise PTQ and identify a critical limitation: the exponential accumulation and growth of quantization errors across network layers. To address this issue, we propose QEP, a general framework that explicitly propagates and compensates for accumulated quantization errors. Extensive experiments demonstrate that QEP substantially improves performance, especially in low-bit quantization scenarios. These findings underscore that meaningful progress in layer-wise PTQ can still be made by revisiting fundamental strategies, complementing recent trends primarily centered around non-linear and block-wise quantization techniques. Integrating QEP with these advanced quantization methods in the future presents a promising approach toward achieving extreme compression, potentially exceeding QAT performance.

Limitations QEP relies on a small calibration set, as in other layer-wise PTQ approaches, which makes performance sensitive to data quality; however, it overfits less than comparable methods such as GPTQ and AWQ. The method also introduces a per-layer propagation-strength parameter α_l; Although this parameter is tunable, a fixed value, e.g., $\alpha_l = {}^1/_2$, works well in most cases, and automatic learning of α_l is left for future work.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Alemán, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. LLaMA: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The LLaMA 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Lingjiao Chen, Matei Zaharia, and James Zou. Frugalgpt: How to use large language models while reducing cost and improving performance. *arXiv preprint arXiv:2305.05176*, 2023.

Jiedong Lang, Zhehao Guo, and Shuyu Huang. A comprehensive study on quantization techniques for large language models. In *Proceedings of the 4th International Conference on Artificial Intelligence, Robotics, and Communication (ICAIRC)*, pages 224–231. IEEE, 2024.

Ruihao Gong, Yifu Ding, Zining Wang, Chengtao Lv, Xingyu Zheng, Jinyang Du, Haotong Qin, Jinyang Guo, Michele Magno, and Xianglong Liu. A survey of low-bit large language models: Basics, systems, and algorithms. *arXiv preprint arXiv:2409.16694*, 2024.

Wenxiao Wang, Wei Chen, Yicong Luo, Yongliu Long, Zhengkai Lin, Liye Zhang, Binbin Lin, Deng Cai, and Xiaofei He. Model compression and efficient inference for large language models: A survey. *arXiv preprint arXiv:2402.09748*, 2024.

Hongrong Cheng, Miao Zhang, and Javen Qinfeng Shi. A survey on deep neural network pruning: Taxonomy, comparison, analysis, and recommendations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

Menglin Yang, Jialin Chen, Yifei Zhang, Jiahong Liu, Jiasheng Zhang, Qiyao Ma, Harshit Verma, Qianru Zhang, Min Zhou, Irwin King, et al. Low-rank adaptation for foundation models: A comprehensive review. *arXiv preprint arXiv:2501.00365*, 2024a.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. LoRA: Low-rank adaptation of large language models. *International Conference on Learning Representations (ICLR)*, 2022.

Xiaohan Xu, Ming Li, Chongyang Tao, Tao Shen, Reynold Cheng, Jinyang Li, Can Xu, Dacheng Tao, and Tianyi Zhou. A survey on knowledge distillation of large language models. *arXiv preprint arXiv:2402.13116*, 2024a.

Chuanpeng Yang, Yao Zhu, Wang Lu, Yidong Wang, Qian Chen, Chenlong Gao, Bingjie Yan, and Yiqiang Chen. Survey on knowledge distillation for large language models: Methods, evaluation, and application. *ACM Transactions on Intelligent Systems and Technology*, 2024b.

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.

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.

Zhewei Yao, Reza Yazdani Aminabadi, Minjia Zhang, Xiaoxia Wu, Conglong Li, and Yuxiong He. Zeroquant: Efficient and affordable post-training quantization for large-scale transformers. *Advances in Neural Information Processing Systems*, 35:27168–27183, 2022.

Jerry Chee, Yaohui Cai, Volodymyr Kuleshov, and Christopher M. De Sa. Quip: 2-bit quantization of large language models with guarantees. *Advances in Neural Information Processing Systems*, 36: 4396–4429, 2023.

Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. Quip#: Even better LLM quantization with hadamard incoherence and lattice codebooks. *arXiv preprint arXiv:2402.04396*, 2024.

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.

Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. Extreme compression of large language models via additive quantization. *arXiv preprint arXiv:2401.06118*, 2024.

Yuzhuang Xu, Xu Han, Zonghan Yang, Shuo Wang, Qingfu Zhu, Zhiyuan Liu, Weidong Liu, and Wanxiang Che. Onebit: Towards extremely low-bit large language models. *arXiv preprint arXiv:2402.11295*, 2024b.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Huaijie Wang, Lingxiao Ma, Fan Yang, Ruiping Wang, Yi Wu, and Furu Wei. Bitnet: Scaling 1-bit transformers for large language models. *arXiv preprint arXiv:2310.11453*, 2023.

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.

Vladimir Malinovskii, Denis Mazur, Ivan Ilin, Denis Kuznedelev, Konstantin Burlachenko, Kai Yi, Dan Alistarh, and Peter Richtárik. PV-Tuning: Beyond straight-through estimation for extreme LLM compression. *Advances in Neural Information Processing Systems*, 37:5074–5121, 2024.

Tim Dettmers and Luke Zettlemoyer. The case for 4-bit precision: k-bit inference scaling laws. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 7750–7774, 2023.

Jiaqi Zhao, Ming Wang, Miao Zhang, Yuzhang Shang, Xuebo Liu, Yaowei Wang, Min Zhang, and Liqiang Nie. Benchmarking post-training quantization in LLMs: Comprehensive taxonomy, unified evaluation, and comparative analysis. *arXiv preprint arXiv:2502.13178*, 2025.

Kayhan Behdin, Ayan Acharya, Sathiya Keerthi Aman Gupta, and Rahul Mazumder. Quantease: Optimization-based quantization for language models—an efficient and intuitive algorithm. *stat*, 1050:5, 2023.

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. *arXiv preprint arXiv:2409.17066*, 2024a.

Tianyi Zhang and Anshumali Shrivastava. Leanquant: Accurate and scalable large language model quantization with loss-error-aware grid. *arXiv preprint arXiv:2407.10032*, 2024.

Ziyi Guan, Hantao Huang, Yupeng Su, Hong Huang, Ngai Wong, and Hao Yu. APTQ: Attention-aware post-training mixed-precision quantization for large language models. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*, pages 1–6, 2024.

Zechun Liu, Changsheng Zhao, Igor Fedorov, Bilge Soran, Dhruv Choudhary, Raghuraman Krishnamoorthi, Vikas Chandra, Yuandong Tian, and Tijmen Blankevoort. Spinquant: LLM quantization with learned rotations. *arXiv preprint arXiv:2405.16406*, 2024b.

Saleh Ashkboos, Amirkeivan Mohtashami, Maximilian Croci, Bo Li, Pashmina Cameron, Martin Jaggi, Dan Alistarh, Torsten Hoefler, and James Hensman. Quarot: Outlier-free 4-bit inference in rotated LLMs. *Advances in Neural Information Processing Systems*, 37:100213–100240, 2024.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. Gpt3.INT8(): 8-bit matrix multiplication for transformers at scale. *Advances in Neural Information Processing Systems*, 35: 30318–30332, 2022.

Tim Dettmers, Ruslan Svirschevski, Vage Egiazarian, Denis Kuznedelev, Elias Frantar, Saleh Ashkboos, Alexander Borzunov, Torsten Hoefler, and Dan Alistarh. Spqr: A sparse-quantized representation for near-lossless LLM weight compression. *arXiv preprint arXiv:2306.03078*, 2023.

Yuzhang Shang, Zhihang Yuan, Qiang Wu, and Zhen Dong. Pb-llm: Partially binarized large language models. *arXiv preprint arXiv:2310.00034*, 2023.

Elias Frantar and Dan Alistarh. Optimal brain compression: A framework for accurate post-training quantization and pruning. *Advances in Neural Information Processing Systems*, 35:4475–4488, 2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. The pile: An 800 gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*, 2020.

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.

Fengqing Jiang. Identifying and mitigating vulnerabilities in llm-integrated applications. Master's thesis, University of Washington, 2024.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. OPT: Open pre-trained transformer language models. https://arxiv.org/abs/2205.01068, 2022.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, Jonathan Tow, Alexander M. Rush, Stella Biderman, et al. BLOOM: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*, 2022. v4.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following LLaMA model. GitHub repository, 2023. URL https://github.com/tatsu-lab/stanford_alpaca.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, et al. Vicuna: An open-source chatbot impressing GPT-4 with 90%* chatgpt quality. https://lmsys.org/blog/2023-03-30-vicuna/, March 2023.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 38087–38099, 2023.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. https://arxiv.org/abs/1609.07843, 2016.

Mitchell Marcus, Grace Kim, Mary Ann Marcinkiewicz, Robert MacIntyre, Ann Bies, Mark Ferguson, Karen Katz, and Britta Schasberger. The Penn Treebank: Annotating predicate argument structure. In *Human Language Technology Workshop (HLT)*, 1994.

Michael Boratko, Harshit Padigela, Divyendra Mikkilineni, Pritish Yuvraj, Rajarshi Das, Andrew McCallum, Maria Chang, Achille Fokoue-Nkoutche, Pavan Kapanipathi, Nicholas Mattei, et al. A systematic classification of knowledge, reasoning, and context within the ARC dataset. https://arxiv.org/abs/1806.00358, 2018.

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, pages 7432–7439, 2020.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of NAACL-HLT*, pages 839–849, 2016.

Ji Lin, Chuang Gan, and Song Han. Defensive quantization: When efficiency meets robustness. *arXiv preprint arXiv:1904.08444*, 2019.

Yuhang Li, Ruokai Yin, Donghyun Lee, Shiting Xiao, and Priyadarshini Panda. Gptaq: Efficient finetuning-free quantization for asymmetric calibration. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=QdELyl0FST.

Dongwon Jo, Taesu Kim, Yulhwa Kim, and Jae-Joon Kim. Mixture of scales: Memory-efficient token-adaptive binarization for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. URL https://arxiv.org/abs/2406.12311.

Peijie Dong, Lujun Li, Yuedong Zhong, Dayou Du, Ruibo Fan, Yuhan Chen, Zhenheng Tang, Qiang Wang, Wei Xue, Yike Guo, and Xiaowen Chu. Stbllm: Breaking the 1-bit barrier with structured binary llms. In *International Conference on Learning Representations (ICLR)*, 2025. URL https://openreview.net/forum?id=6XUSDvBFkV.

Vladimír Boža and Vladimír Macko. Addition is almost all you need: Compressing neural networks with double binary factorization. *CoRR*, abs/2505.11076, 2025. doi: 10.48550/arXiv.2505.11076. URL https://arxiv.org/abs/2505.11076.

Banseok Lee, Dongkyu Kim, Youngcheon You, and Youngmin Kim. Littlebit: Ultra low-bit quantization via latent factorization. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://openreview.net/forum?id=zJzu9evD5K.

Ziang Long, Penghang Yin, and Jack Xin. Learning quantized neural nets by coarse gradient method for nonlinear classification. *Research in the Mathematical Sciences*, 8(3):48, 2021.

Yuma Ichikawa, Shuhei Kashiwamura, and Ayaka Sakata. High-dimensional learning dynamics of quantized models with straight-through estimator. *arXiv preprint arXiv:2510.10693*, 2025.

Penghang Yin, Jiancheng Lyu, Shuai Zhang, Stanley Osher, Yingyong Qi, and Jack Xin. Understanding straight-through estimator in training activation quantized neural nets. *arXiv preprint arXiv:1903.05662*, 2019.

Yuma Ichikawa and Hiroaki Iwashita. Continuous parallel relaxation for finding diverse solutions in combinatorial optimization problems. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=ix33zd5zCw.

Yuma Ichikawa and Yamato Arai. Optimization by parallel quasi-quantum annealing with gradient-based sampling. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=9EfBeXaXf0.

Haoran Sun, Katayoon Goshvadi, Azade Nova, Dale Schuurmans, and Hanjun Dai. Revisiting sampling for combinatorial optimization. In *International Conference on Machine Learning*, pages 32859–32874. PMLR, 2023.

Yuma Ichikawa. Controlling continuous relaxation for combinatorial optimization. *Advances in Neural Information Processing Systems*, 37:47189–47216, 2024.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes] .

   Justification: The abstract and Introduction explicitly state (i) the limitation of existing layer-wise PTQ, (ii) the proposal of QEP, and (iii) the empirical and theoretical gains; all are substantiated by proofs in Appendix B and experiments in Section 6.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: Section 5.3 and Section 6.1 acknowledge over-fitting risks, calibration-set bias, and the need for layer-wise tuning of α_l; computational trade-offs for large MLP blocks are also discussed.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

Answer: [Yes] .

Justification: All assumptions are stated before Proposition 5.1, Theorem 5.2, etc., and full proofs are given in Appendix B.

Guidelines:

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

4. **Experimental result reproducibility**

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

Answer: [Yes]

Justification: Section 6 and Appendix C detail the datasets, bit-widths, calibration sizes, and default values of α_l; Table 3 reports the runtimes for the quantization process.

Guidelines:

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

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

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

   Answer: [Yes]

   Justification: All datasets are publicly available, and the supplementary material includes an anonymized ZIP file containing the code, execution scripts, and a README file with a list of required packages.

   Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: Datasets in Section 6, calibration procedure, number of tokens, and hyperparameters (group size, α_l defaults) are listed; no additional training optimiser is used because PTQ is post-training.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The impact of random seeds on QuIP and QEP-QuIP is evaluated using error bars and reported in Appendix D.3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.

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

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: Table 3 reports wall-clock time on a single NVIDIA V100 for 7B–70B models.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The work only processes publicly released models and text corpora; no personal, sensitive, or protected data are used.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [No]

Justification: This work focuses on an algorithmic advance in model quantization; given space constraints we prioritized technical details and empirical results, so a full societal-impact discussion was omitted.

Guidelines:

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

11. **Safeguards**

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

Answer: [NA]

Justification: The paper does not release new generative models; it only provides a compression method and therefore poses minimal additional misuse risk.

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: Every external asset we use (Llama-2/3, Mistral-7B, C4, PTB, WikiText-2, ARC-E, PIQA, StoryCloze) is fully cited, publicly available, where the respective download pages state their open-source licenses.

Guidelines:

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

13. **New assets**

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

    Answer: [NA] .

    Justification: The paper introduces no new dataset or model; it only supplies an algorithmic framework.

    Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [NA]

    Justification: No human-subject data or crowdsourcing is involved.

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [NA]

Justification: Not applicable, no human subjects.

Guidelines:

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

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

Answer: [NA]

Justification: An LLM was used only for writing assistance, proofreading, and minor code refactoring; it played no role in the core methodology or experiments.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.

A Additional Related Work

Quantization error mitigation Defensive Quantization (DQ) [Lin et al., 2019] mitigates error accumulation by adding an orthogonality penalty to weights to reduce the correlation-driven amplification of quantization noise and by employing gradient-based quantization to position quantization levels that further suppress propagated error, thereby enhancing robustness. In contrast, the QEP module is a plug-and-play component for general layer-wise quantization that explicitly modulates propagation strength under standard linear, gradient-free quantization methods, such as GPTQ, AWQ, and QuIP.

Relation to GPTAQ GPTAQ [Li et al., 2025] optimizes the same local objective but is closely related to GPTQ and therefore does not easily generalize to other layer-wise PTQ methods, such as AWQ and QuIP. In contrast, QEP adds the correction term in Eq. (4) directly to the pre-trained weights, enabling plug-and-play use with diverse PTQ algorithms and strong performance in low-bit regimes. Moreover, whereas GPTAQ offers no guarantee that its local optimization reduces global quantization error, our analysis provides such a guarantee. QEP also introduces a per-layer propagation-strength parameter, α_l, which mitigates the overfitting often observed with GPTQ and GPTQA.

Extreme low-bit layer-wise PTQ In the ultra-low precision regime, the layer-wise PTQ pipeline described in the main text, such as GPTQ and AWQ with standard linear quantizers, often becomes inadequate. As a result, many methods adopt alternative formalisms that introduce additional degrees of freedom beyond naive rounding to avoid catastrophic quality degradation. Representative examples include SVID-based 1-bit parameterizations [Xu et al., 2024b], token-adaptive mixtures of scaling factors [Jo et al., 2024], and structured sparsity designed for extreme quantization [Dong et al., 2025]. In parallel, *binary-factor* formats decompose each weight matrix into bit-packed sign factors with lightweight diagonal scaling, allowing inference to be largely driven by efficient 1-bit kernels [Boža and Macko, 2025, Lee et al., 2025]. In these formats, optimization is also more challenging: strictly binary variables and highly non-smooth objectives can render straight-through estimators brittle [Long et al., 2021, Ichikawa et al., 2025, Yin et al., 2019], motivating the use of discrete optimization or robust relaxation-based solvers. Recent advances in controlled continuous relaxations and annealing-style objectives have emphasized promising techniques such as QQA and iSCO [Ichikawa and Iwashita, 2025, Ichikawa and Arai, 2025, Sun et al., 2023, Ichikawa, 2024]. A natural direction for future work is to examine how these extreme low-bit approaches interact with our error-propagation perspective. For instance, we should investigate whether combining QEP with binary-factor or sparsity-based methods can further suppress cross-layer error growth while maintaining the strong INT2 performance observed here and yielding larger gains as precision approaches 1 bit.

B Additional Theoretical Results

This section presents proofs of the propositions stated in the main text along with additional theoretical analyzes. To avoid ambiguity, we will fix the following notation throughout this section:

Symbol	**Description**
$\boldsymbol{X} \in \mathbb{R}^{d_1 \times m}$	Calibration dataset (input activations at layer 1)
$\boldsymbol{W}_l \in \mathbb{R}^{n_l \times d_l}$	Full-precision weight matrix at layer l
$\widehat{\boldsymbol{W}}_l \in \mathbb{Q}^{n_l \times d_l}$	Quantized weight matrix at layer l
$\sigma_l(\cdot)$	Activation function at layer l
$\boldsymbol{X}_{l+1} := \sigma_l(\boldsymbol{W}_l \boldsymbol{X}_l)$	Full-precision activations at layer $(l+1)$
$\widehat{\boldsymbol{X}}_{l+1} := \sigma_l(\widehat{\boldsymbol{W}}_l \widehat{\boldsymbol{X}}_l)$	Quantized activations at layer $(l+1)$
$\boldsymbol{\delta}_l := \boldsymbol{X}_l - \widehat{\boldsymbol{X}}_l$	Quantization error matrix at layer l
$\boldsymbol{H}_l := \boldsymbol{X}_l \boldsymbol{X}_l^\top$	Empirical Hessian of a full-precision model at layer l
$\widehat{\boldsymbol{H}}_l := \widehat{\boldsymbol{X}}_l \widehat{\boldsymbol{X}}_l^\top$	Empirical Hessian of a quantized model at layer l

In the following section, we assume that $\boldsymbol{H}_l$ and $\widehat{\boldsymbol{W}_l}$ are invertible. This assumption is standard in existing layer-wise PTQ methods, which also use these inverse matrices. To ensure numerical stability, a diagonal matrix $\rho \boldsymbol{I}$, $\rho > 0$ is commonly added to the Hessian when its inversion becomes numerically unstable. The subsequent analysis remains consistent and valid even when applying this stabilization procedure, which simply involves adding $\rho \boldsymbol{I}$ to the Hessian in the following derivations.

Throughout this section, we examine the first-order linear term in the weight perturbations $\{\boldsymbol{E}_l\}$. Concretely, we define each quantity as $\boldsymbol{A} = \boldsymbol{A}^{(0)} + \boldsymbol{A}^{(1)} + \boldsymbol{R}$, where $\boldsymbol{A}^{(1)}$ collects all terms linear in $\{\boldsymbol{E}_l\}$ and the remainder satisfies $\|\boldsymbol{R}\| = \mathcal{O}(\max_k \|\boldsymbol{E}_k\|_2^2)$ as $\max_k \|\boldsymbol{E}_k\|_2 \to 0$. This matches practice: with INT8 rounding $\|\boldsymbol{E}_l\|_F / \|\boldsymbol{W}_l\|_F = 10^{-2} \sim 10^{-1}$, so quadratic terms are one order of magnitude smaller than any first–order contribution. Furthermore, The baseline PTQ and the QEP pipeline use the same quantiser configuration, hence they induce *errors of the same order*. We therefore write the same symbol $\boldsymbol{E}_l$ for the error matrix in either scheme; any difference is at most a few percent and does not affect first–order bounds.

B.1 Derivation of Proposition 5.1

This section presents detailed proofs of Proposition 5.1 stated in the main text.

Proof. First, we rewrite the residual inside the Frobenius norm by using the following relationship: $\boldsymbol{W}_l \boldsymbol{X}_l = \boldsymbol{W}_l \widehat{\boldsymbol{X}_l} + \boldsymbol{W}_l \boldsymbol{\delta}_l$. Thus, the objective can be expressed as follows:

$$\left\| \boldsymbol{W}_l \boldsymbol{X}_l - \widehat{\boldsymbol{W}_l} \widehat{\boldsymbol{X}_l} \right\|_F^2 = \left\| (\boldsymbol{W}_l - \widehat{\boldsymbol{W}_l}) \widehat{\boldsymbol{X}_l} + \boldsymbol{W}_l \boldsymbol{\delta}_l \right\|_F^2 .$$

Since the objective is a strictly convex quadratic function of $\widehat{\boldsymbol{W}_l}$ when $\widehat{\boldsymbol{H}_l}$ is invertible, the stationary point is the unique minimizer. To find the minimizer $\widehat{\boldsymbol{W}_l}$, we set the gradient of the expression with respect to $\widehat{\boldsymbol{W}_l}$, equal to zero. Using standard matrix calculus, we find that the calculus for a stationary point is

$$(\boldsymbol{W}_l - \widehat{\boldsymbol{W}_l}) \widehat{\boldsymbol{X}_l} \widehat{\boldsymbol{X}_l}^\top + \boldsymbol{W}_l \boldsymbol{\delta}_l \widehat{\boldsymbol{X}_l}^\top = \boldsymbol{0}.$$

By defining $\widehat{\boldsymbol{H}_l} := \widehat{\boldsymbol{X}_l} \widehat{\boldsymbol{X}_l}^\top$, the above condition can be rewritten as

$$(\boldsymbol{W}_l - \widehat{\boldsymbol{W}_l}) \widehat{\boldsymbol{H}_l} = -\boldsymbol{W}_l \boldsymbol{\delta}_l \widehat{\boldsymbol{X}_l}^\top .$$

Assuming $\widehat{\boldsymbol{H}_l}$ is invertible, we multiply both sides on the right by $\widehat{\boldsymbol{H}_l}^{-1}$, obtaining

$$\boldsymbol{W}_l - \widehat{\boldsymbol{W}_l} = -\boldsymbol{W}_l \boldsymbol{\delta}_l \widehat{\boldsymbol{X}_l}^\top \widehat{\boldsymbol{H}_l}^{-1},$$

and hence

$$\widehat{\boldsymbol{W}_l} = \boldsymbol{W}_l + \boldsymbol{W}_l \boldsymbol{\delta}_l \widehat{\boldsymbol{X}_l}^\top \widehat{\boldsymbol{H}_l}^{-1}.$$

This closed-form expression is indeed the unique minimizer of the Frobenius norm objective, thus completing the proof. □

B.2 Quantization Error Accumulation

This section demonstrates that, under standard layer-wise PTQ, where each layer is quantized independently without considering downstream effects, the activation difference at the output layer, defined as $\boldsymbol{\delta}_L := \boldsymbol{X}_L - \widehat{\boldsymbol{X}}_L$, grows exponentially with depth, to first order in the quantization noise, under mild conditions.

Proposition B.1. *For each layer $l = 1, \ldots, L+1$, the activation error can be expressed as follows:*

$$\boldsymbol{\delta}_l = -\sum_{k=1}^{l-1} \left(\prod_{s=k+1}^{l-1} \boldsymbol{J}_s \boldsymbol{W}_s \right) \boldsymbol{J}_k \boldsymbol{E}_k \boldsymbol{X}_k + \mathcal{O}\left(\max_{k \le l-1} \|\boldsymbol{E}_k\|_F^2 \right),$$

where the empty product $\prod_{s=l}^{l-1}$ is defined to be the identity matrix, and $\boldsymbol{E}_k := \widehat{\boldsymbol{W}}_k - \boldsymbol{W}_k$ represents the weight quantization error at layer k.

Proof. Consider explicitly the activations at layer l in both full-precision and quantized forms:

$$\boldsymbol{X}_l = \sigma_{l-1}(\boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1}), \ \ \widehat{\boldsymbol{X}_l} = \sigma_{l-1}(\widehat{\boldsymbol{W}}_{l-1}\widehat{\boldsymbol{X}}_{l-1}).$$

By recursively applying this relation back to the first layer, we derive the activation difference $\boldsymbol{\delta}_l$ as

$$\begin{aligned}
\boldsymbol{\delta}_l &= \boldsymbol{X}_l - \widehat{\boldsymbol{X}_l} \\
&= \sigma_{l-1}(\boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1}) - \sigma_{l-1}(\widehat{\boldsymbol{W}}_{l-1}\widehat{\boldsymbol{X}}_{l-1}) \\
&= \boldsymbol{J}_{l-1}(\boldsymbol{W}_{l-1}\boldsymbol{X}_{l-1} - \widehat{\boldsymbol{W}}_{l-1}\widehat{\boldsymbol{X}}_{l-1}) + \mathcal{O}\left(\max\{\boldsymbol{E}_{l-1}^2, \boldsymbol{\delta}_{l-1}^2\}\right) \\
&= \boldsymbol{J}_{l-1}\left[-\boldsymbol{E}_{l-1}\boldsymbol{X}_{l-1} + \boldsymbol{W}_{l-1}\boldsymbol{\delta}_{l-1}\right] + \mathcal{O}\left(\max\{\boldsymbol{E}_{l-1}^2, \boldsymbol{\delta}_{l-1}^2\}\right) \\
&= -\boldsymbol{J}_{l-1}\boldsymbol{E}_{l-1}\boldsymbol{X}_{l-1} + \boldsymbol{J}_{l-1}\boldsymbol{W}_{l-1}\boldsymbol{\delta}_{l-1} + \mathcal{O}\left(\max\{\boldsymbol{E}_{l-1}^2, \boldsymbol{\delta}_{l-1}^2\}\right).
\end{aligned}$$

By explicitly expanding $\boldsymbol{\delta}_{l-1}$, we obtain

$$\boldsymbol{\delta}_{l-1} = -\boldsymbol{J}_{l-2}\boldsymbol{E}_{l-2}\boldsymbol{X}_{l-2} + \boldsymbol{J}_{l-2}\boldsymbol{W}_{l-2}\boldsymbol{\delta}_{l-2} + \mathcal{O}\left(\max\{\boldsymbol{E}_{l-2}^2, \boldsymbol{\delta}_{l-2}^2\}\right).$$

Substituting this expression into the previous equation yields

$$\begin{aligned}
\boldsymbol{\delta}_l &= -\boldsymbol{J}_{l-1}\boldsymbol{E}_{l-1}\boldsymbol{X}_{l-1} \\
&\quad + \boldsymbol{J}_{l-1}\boldsymbol{W}_{l-1}\left[-\boldsymbol{J}_{l-2}\boldsymbol{E}_{l-2}\boldsymbol{X}_{l-2} + \boldsymbol{J}_{l-2}\boldsymbol{W}_{l-2}\boldsymbol{\delta}_{l-2}\right] + \mathcal{O}\left(\max\{\boldsymbol{E}_{l-1}^2, \boldsymbol{E}_{l-2}^2, \boldsymbol{\delta}_{l-2}^2\}\right) \\
&= -\boldsymbol{J}_{l-1}\boldsymbol{E}_{l-1}\boldsymbol{X}_{l-1} - \boldsymbol{J}_{l-1}\boldsymbol{W}_{l-1}\boldsymbol{J}_{l-2}\boldsymbol{E}_{l-2}\boldsymbol{X}_{l-2} \\
&\quad + \boldsymbol{J}_{l-1}\boldsymbol{W}_{l-1}\boldsymbol{J}_{l-2}\boldsymbol{W}_{l-2}\boldsymbol{\delta}_{l-2} + \mathcal{O}\left(\max\{\boldsymbol{E}_{l-1}^2, \boldsymbol{E}_{l-2}^2, \boldsymbol{\delta}_{l-2}^2\}\right).
\end{aligned}$$

By recursively repeating this explicit expansion down to the first layer, we obtain the fully expanded form as follows, noting $\boldsymbol{\delta}_1 = \boldsymbol{0}$:

$$\boldsymbol{\delta}_l = -\sum_{k=1}^{l-1}\left(\prod_{s=k+1}^{l-1}\boldsymbol{J}_s\boldsymbol{W}_s\right)\boldsymbol{J}_k\boldsymbol{E}_k\boldsymbol{X}_k + \mathcal{O}\left(\max_{k\le l-1}\|\boldsymbol{E}_k\|^2\right),$$

where the empty product for $s = l, \dots, l-1$ is defined as the identity matrix. □

Proposition B.2. *Assume each activation $\sigma_l : \mathbb{R}^{n_l\times m} \to \mathbb{R}^{n_l\times m}$ is γ_l-Lipschitz with respect to the Frobenius norm and satisfies $\sigma_l(\boldsymbol{0}) = \boldsymbol{0}$:*

$$\|\sigma_l(\boldsymbol{U}) - \sigma_l(\boldsymbol{V})\|_F \le \gamma_l\|\boldsymbol{U} - \boldsymbol{V}\|_F, \ \ \gamma_l > 0.$$

Let $\boldsymbol{X}_1 = \widehat{\boldsymbol{X}_1} = \boldsymbol{X}$; for $l = 1, \dots, L-1$ define

$$\boldsymbol{X}_{l+1} = \sigma_l(\boldsymbol{W}_l\boldsymbol{X}_l), \ \ \widehat{\boldsymbol{X}}_{l+1} = \sigma_l(\widehat{\boldsymbol{W}}_l\widehat{\boldsymbol{X}}_l), \ \ \widehat{\boldsymbol{W}}_l = \boldsymbol{W}_l + \boldsymbol{E}_l,$$

and $\boldsymbol{\delta}_l \coloneqq \boldsymbol{X}_l - \widehat{\boldsymbol{X}}_l$. Assume $\|\boldsymbol{W}_l\|_2 > 0$ for all $l = 1, \dots, L-1$ and set

$$G_{L-1} \coloneqq \prod_{l=1}^{L-1}\gamma_l\|\boldsymbol{W}_l\|_2, \ \ r \coloneqq \max_{1\le k\le L-1}\frac{\|\boldsymbol{E}_k\|_2}{\|\boldsymbol{W}_k\|_2}.$$

Then the final activation mismatch satisfies the explicit bound

$$\|\boldsymbol{\delta}_L\|_F \le \left((1+r)^{L-1} - 1\right)G_{L-1}\|\boldsymbol{X}\|_F. \tag{7}$$

Proof. Since $\sigma_l(\boldsymbol{0}) = \boldsymbol{0}$ and σ_l are γ_l-Lipschitz,

$$\|\boldsymbol{X}_{l+1}\|_F = \|\sigma_l(\boldsymbol{W}_l\boldsymbol{X}_l) - \sigma_l(\boldsymbol{0})\|_F \le \gamma_l\|\boldsymbol{W}_l\boldsymbol{X}_l\|_F \le \gamma_l\|\boldsymbol{W}_l\|_2\|\boldsymbol{X}_l\|_F.$$

By induction,

$$\|\boldsymbol{X}_l\|_F \le \left(\prod_{t=1}^{l-1}\gamma_t\|\boldsymbol{W}_t\|_2\right)\|\boldsymbol{X}\|_F = G_{l-1}\|\boldsymbol{X}\|_F, \ \ l \ge 1. \tag{8}$$

Using Lipschitz continuity again,

$$\begin{aligned}
\|\boldsymbol{\delta}_{l+1}\|_F &= \|\sigma_l(\boldsymbol{W}_l\boldsymbol{X}_l) - \sigma_l(\widehat{\boldsymbol{W}}_l\widehat{\boldsymbol{X}}_l)\|_F \\
&\le \gamma_l\|\boldsymbol{W}_l\boldsymbol{X}_l - \widehat{\boldsymbol{W}}_l\widehat{\boldsymbol{X}}_l\|_F.
\end{aligned}$$

Since $\widehat{\boldsymbol{W}}_l = \boldsymbol{W}_l + \boldsymbol{E}_l$ and $\boldsymbol{\delta}_l = \boldsymbol{X}_l - \widehat{\boldsymbol{X}}_l$,

$$\boldsymbol{W}_l\boldsymbol{X}_l - \widehat{\boldsymbol{W}}_l\widehat{\boldsymbol{X}}_l = \boldsymbol{W}_l\boldsymbol{X}_l - (\boldsymbol{W}_l + \boldsymbol{E}_l)\widehat{\boldsymbol{X}}_l = -\boldsymbol{E}_l\boldsymbol{X}_l + \widehat{\boldsymbol{W}}_l\boldsymbol{\delta}_l,$$

hence, by the triangle inequality and $\|AB\|_F \le \|A\|_2\|B\|_F$,

$$\|\boldsymbol{\delta}_{l+1}\|_F \le \gamma_l\Big(\|\boldsymbol{E}_l\|_2\|\boldsymbol{X}_l\|_F + \|\widehat{\boldsymbol{W}}_l\|_2\|\boldsymbol{\delta}_l\|_F\Big). \tag{9}$$

By definition of r, $\|\boldsymbol{E}_l\|_2 \le r\|\boldsymbol{W}_l\|_2$. Additionally, $\|\widehat{\boldsymbol{W}}_l\|_2 \le \|\boldsymbol{W}_l\|_2 + \|\boldsymbol{E}_l\|_2 \le (1+r)\|\boldsymbol{W}_l\|_2$. Combining these with (8) in (9) yields

$$\|\boldsymbol{\delta}_{l+1}\|_F \le \gamma_l\|\boldsymbol{W}_l\|_2\Big(rG_{l-1}\|\boldsymbol{X}\|_F + (1+r)\|\boldsymbol{\delta}_l\|_F\Big).$$

Define the normalized quantity

$$a_l \coloneqq \frac{\|\boldsymbol{\delta}_l\|_F}{G_{l-1}\|\boldsymbol{X}\|_F}, \ \ l \ge 1.$$

Note $a_1 = 0$ because $\boldsymbol{\delta}_1 = \boldsymbol{0}$. Dividing the previous inequality by $G_l\|\boldsymbol{X}\|_F$ (where $G_l = G_{l-1}\gamma_l\|\boldsymbol{W}_l\|_2$) yields

$$a_{l+1} \le r + (1+r)a_l.$$

Let $b_l \coloneqq a_l + 1$. Then $b_{l+1} \le (1+r)b_l$ and $b_1 = 1$; hence $b_l \le (1+r)^{l-1}$; therefore

$$a_l \le (1+r)^{l-1} - 1.$$

Taking $l = L$ gives Eq. (7). □

Proposition B.3. *Assume each activation $\sigma_l : \mathbb{R}^{n_l\times m} \to \mathbb{R}^{n_l\times m}$ is γ_l-Lipschitz with respect to the Frobenius norm and satisfies $\sigma_l(\boldsymbol{0}) = \boldsymbol{0}$:*

$$\|\sigma_l(\boldsymbol{U}) - \sigma_l(\boldsymbol{V})\|_F \le \gamma_l\|\boldsymbol{U} - \boldsymbol{V}\|_F.$$

Assume moreover that σ_l is Fréchet differentiable at the full-precision pre-activation $\boldsymbol{Y}_l \coloneqq \boldsymbol{W}_l\boldsymbol{X}_l$ for each $l = 1, \dots, L-1$. Let $\boldsymbol{X}_1 = \widehat{\boldsymbol{X}}_1 = \boldsymbol{X}$ and for $l = 1, \dots, L-1$ define

$$\boldsymbol{X}_{l+1} = \sigma_l(\boldsymbol{W}_l\boldsymbol{X}_l), \ \ \widehat{\boldsymbol{X}}_{l+1} = \sigma_l(\widehat{\boldsymbol{W}}_l\widehat{\boldsymbol{X}}_l), \ \ \widehat{\boldsymbol{W}}_l = \boldsymbol{W}_l + \boldsymbol{E}_l.$$

Assume $\|\boldsymbol{W}_k\|_2 > 0$ for $k = 1, \dots, L-1$ and define

$$\boldsymbol{G}_{L-1} \coloneqq \prod_{l=1}^{L-1}\gamma_l\|\boldsymbol{W}_l\|_2, \ \ r \coloneqq \max_{1\le k\le L-1}\frac{\|\boldsymbol{E}_k\|_2}{\|\boldsymbol{W}_k\|_2}.$$

Define the first-order component *$\boldsymbol{\delta}_L^{(1)}$ as follows. For $t \in \mathbb{R}$ let $\widehat{\boldsymbol{W}}_l(t) \coloneqq \boldsymbol{W}_l + t\boldsymbol{E}_l$ and define $\widehat{\boldsymbol{X}}_1(t) \coloneqq \boldsymbol{X}$, $\widehat{\boldsymbol{X}}_{l+1}(t) \coloneqq \sigma_l(\widehat{\boldsymbol{W}}_l(t)\widehat{\boldsymbol{X}}_l(t))$. Let $\boldsymbol{\delta}_l(t) \coloneqq \boldsymbol{X}_l - \widehat{\boldsymbol{X}}_l(t)$. Then $\boldsymbol{\delta}_L^{(1)}$ is defined by the derivative*

$$\boldsymbol{\delta}_L^{(1)} \coloneqq \frac{d}{dt}\bigg|_{t=0}\boldsymbol{\delta}_L(t).$$

Under these assumptions,

$$\|\boldsymbol{\delta}_L^{(1)}\|_F \le (L-1)r\boldsymbol{G}_{L-1}\|\boldsymbol{X}\|_F. \tag{10}$$

In particular, if $\gamma_l\|\boldsymbol{W}_l\|_2 \le 1 + \varepsilon$ for all l, then

$$\|\boldsymbol{\delta}_L^{(1)}\|_F \le (L-1)r(1+\varepsilon)^{L-1}\|\boldsymbol{X}\|_F. \tag{11}$$

Proof. Since $\sigma_l(\boldsymbol{0}) = \boldsymbol{0}$ and σ_l is γ_l-Lipschitz,

$$\|\boldsymbol{X}_{l+1}\|_F = \|\sigma_l(\boldsymbol{W}_l\boldsymbol{X}_l) - \sigma_l(\boldsymbol{0})\|_F \le \gamma_l\|\boldsymbol{W}_l\boldsymbol{X}_l\|_F \le \gamma_l\|\boldsymbol{W}_l\|_2\|\boldsymbol{X}_l\|_F.$$

By induction,

$$\|\boldsymbol{X}_l\|_F \le \left(\prod_{t=1}^{l-1}\gamma_t\|\boldsymbol{W}_t\|_2\right)\|\boldsymbol{X}\|_F, \ \ l \ge 1. \tag{12}$$

Fix $l \in \{1, \dots, L-1\}$. By assumption, σ_l is Fréchet differentiable at $\boldsymbol{Y}_l := \boldsymbol{W}_l \boldsymbol{X}_l$. Let $\boldsymbol{J}_l := D\sigma_l(\boldsymbol{Y}_l)$ denote its Fréchet derivative. Because σ_l is γ_l-Lipschitz, the operator norm of $\boldsymbol{J}_l$, induced by the Frobenius norm, satisfies

$$\|\boldsymbol{J}_l\|_{\mathrm{op}} \le \gamma_l. \tag{13}$$

Indeed, for any $\boldsymbol{H}$,

$$\|\boldsymbol{J}_l[\boldsymbol{H}]\|_F = \lim_{t \to 0} \frac{\|\sigma_l(\boldsymbol{Y}_l + t\boldsymbol{H}) - \sigma_l(\boldsymbol{Y}_l)\|_F}{|t|} \le \lim_{t \to 0} \frac{\gamma_l \|t\boldsymbol{H}\|_F}{|t|} = \gamma_l \|\boldsymbol{H}\|_F.$$

Now consider $\widehat{\boldsymbol{X}}_{l+1}(t) = \sigma_l(\widehat{\boldsymbol{W}}_l(t)\widehat{\boldsymbol{X}}_l(t))$. Since matrix multiplication is smooth and σ_l is differentiable at $\boldsymbol{Y}_l$, the chain rule gives

$$\left.\frac{d}{dt}\right|_{t=0} \widehat{\boldsymbol{X}}_{l+1}(t) = \boldsymbol{J}_l \left[\boldsymbol{E}_l \boldsymbol{X}_l + \boldsymbol{W}_l \left.\frac{d}{dt}\right|_{t=0} \widehat{\boldsymbol{X}}_l(t) \right].$$

Because $\boldsymbol{\delta}_l(t) = \boldsymbol{X}_l - \widehat{\boldsymbol{X}}_l(t)$, we have $\frac{d}{dt}\big|_{t=0} \widehat{\boldsymbol{X}}_l(t) = -\boldsymbol{\delta}_l^{(1)}$. Therefore,

$$\boldsymbol{\delta}_{l+1}^{(1)} = -\boldsymbol{J}_l[\boldsymbol{E}_l\boldsymbol{X}_l - \boldsymbol{W}_l\boldsymbol{\delta}_l^{(1)}] = -\boldsymbol{J}_l(\boldsymbol{E}_l\boldsymbol{X}_l) + \boldsymbol{J}_l(\boldsymbol{W}_l\boldsymbol{\delta}_l^{(1)}), \;\; \boldsymbol{\delta}_1^{(1)} = \boldsymbol{0}.$$

Taking Frobenius norms, using (13) and $\|\boldsymbol{AB}\|_F \le \|\boldsymbol{A}\|_2 \|\boldsymbol{B}\|_F$,

$$\|\boldsymbol{\delta}_{l+1}^{(1)}\|_F \le \gamma_l \|\boldsymbol{E}_l\|_2 \|\boldsymbol{X}_l\|_F + \gamma_l \|\boldsymbol{W}_l\|_2 \|\boldsymbol{\delta}_l^{(1)}\|_F. \tag{14}$$

Define $a_l := \|\boldsymbol{\delta}_l^{(1)}\|_F$. From (14) and $a_1 = 0$, straightforward induction yields

$$a_L \le \sum_{k=1}^{L-1} \left(\prod_{s=k+1}^{L-1} \gamma_s \|\boldsymbol{W}_s\|_2 \right) \gamma_k \|\boldsymbol{E}_k\|_2 \|\boldsymbol{X}_k\|_F.$$

Apply Eq. (12) to $\|\boldsymbol{X}_k\|_F$ and factor out $\boldsymbol{G}_{L-1}$:

$$\|\boldsymbol{\delta}_L^{(1)}\|_F \le \boldsymbol{G}_{L-1} \left(\sum_{k=1}^{L-1} \frac{\|\boldsymbol{E}_k\|_2}{\|\boldsymbol{W}_k\|_2} \right) \|\boldsymbol{X}\|_F \le (L-1) r \boldsymbol{G}_{L-1} \|\boldsymbol{X}\|_F,$$

which is Eq. (10). If additionally $\gamma_l \|\boldsymbol{W}_l\|_2 \le 1 + \varepsilon$ for all l, then $\boldsymbol{G}_{L-1} \le (1+\varepsilon)^{L-1}$, proving Eq. (11). □

Proposition B.4. *Consider the 1-dimensional network ($d_l = n_l = 1$) with $\sigma_l(z) = z$. Let $W_l = 1 + \varepsilon$ for all $l = 1, \dots, L-1$ with $\varepsilon > 0$, and let the input be $X = C > 0$. Choose quantized weights*

$$\widehat{W}_l = W_l + E_l, \;\; E_l \equiv c_E > 0, \;\; l = 1, \dots, L-1.$$

Then, for all $L \ge 2$, the exact activation mismatch at layer L satisfies

$$|\delta_L| = |X_L - \widehat{X}_L| \ge (L-1) c_E C (1+\varepsilon)^{L-2}. \tag{15}$$

In particular,

$$|\delta_L| \ge \frac{c_E C}{1+\varepsilon} (1+\varepsilon)^{L-1}. \tag{16}$$

Proof. Since σ_l is the identity map, we have

$$X_L = (1+\varepsilon)^{L-1} C, \;\; \widehat{X}_L = (1+\varepsilon+c_E)^{L-1} C.$$

Hence

$$|\delta_L| = C \left| (1+\varepsilon+c_E)^{L-1} - (1+\varepsilon)^{L-1} \right|.$$

Apply the mean value theorem to $f(t) = t^{L-1}$ on the interval $[1+\varepsilon, \; 1+\varepsilon+c_E]$: there exists $\xi \in (1+\varepsilon, 1+\varepsilon+c_E)$ such that

$$(1+\varepsilon+c_E)^{L-1} - (1+\varepsilon)^{L-1} = f'(\xi) c_E = (L-1)\xi^{L-2} c_E.$$

Since $\xi \ge 1+\varepsilon$, we obtain

$$|\delta_L| \ge (L-1) c_E (1+\varepsilon)^{L-2} C,$$

which proves Eq. (15). Finally, Eq. (16) follows from $(1+\varepsilon)^{L-2} = {}^{1}/_{1+\varepsilon}(1+\varepsilon)^{L-1}$ and $L-1 \ge 1$ for $L \ge 2$. □

B.3 Derivation of Theorem 5.2 and Corollary 5.4

This section presents a rigorous statement and proof of Theorem 5.2 and Corollary 5.4. We first formally restate Theorem 5.2 below.

Theorem B.5. *Consider an L-layer network*

$$\boldsymbol{X}_1 = \boldsymbol{X}, \;\; \boldsymbol{X}_{l+1} = \sigma_l(\boldsymbol{W}_l \boldsymbol{X}_l), \;\; l = 1, \dots, L.$$

Assume each σ_l is γ_l-Lipschitz with respect to $\|\cdot\|_F$ and satisfies $\sigma_l(\mathbf{0}) = \mathbf{0}$. Let the quantized forward pass be

$$\widehat{\boldsymbol{X}}_1 = \boldsymbol{X}, \;\; \widehat{\boldsymbol{X}}_{l+1} = \sigma_l(\widehat{\boldsymbol{W}}_l \widehat{\boldsymbol{X}}_l).$$

Define the activation mismatch as $\boldsymbol{\delta}_l \coloneqq \boldsymbol{X}_l - \widehat{\boldsymbol{X}}_l$.

Fix any matrices $\boldsymbol{E}_l$ and set

$$\widehat{\boldsymbol{W}}_l^{\mathrm{BASE}} \coloneqq \boldsymbol{W}_l + \boldsymbol{E}_l.$$

For QEP, define each l

$$\boldsymbol{W}_l^*(\alpha_l) \coloneqq \boldsymbol{W}_l + \alpha_l \boldsymbol{W}_l \boldsymbol{\delta}_l \widehat{\boldsymbol{X}}_l^\top \widehat{\boldsymbol{H}}_l^{-1}, \;\; \alpha_l \in [0, 1],$$

and set

$$\widehat{\boldsymbol{W}}_l^{\mathrm{QEP}} \coloneqq \boldsymbol{W}_l^*(\alpha_l) + \boldsymbol{E}_l.$$

Define the per-layer pre-activation residuals

$$\boldsymbol{R}_l^M \coloneqq \boldsymbol{W}_l \boldsymbol{X}_l - \widehat{\boldsymbol{W}}_l^M \widehat{\boldsymbol{X}}_l^M, \;\; M \in \{\mathrm{BASE}, \mathrm{QEP}\},$$

the global Lipschitz upper bound on the output mismatch

$$\mathcal{U}^M \coloneqq \sum_{k=1}^{L} \left(\prod_{s=k+1}^{L} \gamma_s \|\boldsymbol{W}_s\|_2 \right) \gamma_k \|\boldsymbol{R}_k^M\|_F.$$

Then for every choice of $\{\alpha_l\}_{l=1}^L \subset [0, 1]$,

$$\mathcal{U}^{\mathrm{QEP}} \leq \mathcal{U}^{\mathrm{BASE}}.$$

Consequently,

$$\|\boldsymbol{\delta}_{L+1}^{\mathrm{QEP}}\|_F \leq \mathcal{U}^{\mathrm{QEP}} \leq \mathcal{U}^{\mathrm{BASE}}, \;\; \|\boldsymbol{\delta}_{L+1}^{\mathrm{BASE}}\|_F \leq \mathcal{U}^{\mathrm{BASE}}.$$

Proof. Fix a method $M \in \{\mathrm{BASE}, \mathrm{QEP}\}$. By Lipschitz continuity,

$$\|\boldsymbol{\delta}_{l+1}^M\|_F = \|\sigma_l(\boldsymbol{W}_l \boldsymbol{X}_l) - \sigma_l(\widehat{\boldsymbol{W}}_l^M \widehat{\boldsymbol{X}}_l^M)\|_F \leq \gamma_l \|\boldsymbol{W}_l \boldsymbol{X}_l - \widehat{\boldsymbol{W}}_l^M \widehat{\boldsymbol{X}}_l^M\|_F = \gamma_l \|\boldsymbol{R}_l^M\|_F.$$

Iterating this inequality through the remaining layers yields

$$\|\boldsymbol{\delta}_{L+1}^M\|_F \leq \sum_{k=1}^{L} \left(\prod_{s=k+1}^{L} \gamma_s \|\boldsymbol{W}_s\|_2 \right) \gamma_k \|\boldsymbol{R}_k^M\|_F = \mathcal{U}^M.$$

It remains to show $\mathcal{U}^{\mathrm{QEP}} \leq \mathcal{U}^{\mathrm{BASE}}$. We prove a stronger per-layer inequality:

$$\|\boldsymbol{R}_l^{\mathrm{QEP}}\|_F \leq \|\boldsymbol{R}_l^{\mathrm{BASE}}\|_F, \;\; \forall l. \tag{17}$$

Fix l and write $\widehat{\boldsymbol{X}}_l \coloneqq \widehat{\boldsymbol{X}}_l^{\mathrm{QEP}}$ and $\boldsymbol{\delta}_l \coloneqq \boldsymbol{X}_l - \widehat{\boldsymbol{X}}_l$. Define the orthogonal projection

$$\boldsymbol{P}_l \coloneqq \widehat{\boldsymbol{X}}_l^\top (\widehat{\boldsymbol{X}}_l \widehat{\boldsymbol{X}}_l^\top)^{-1} \widehat{\boldsymbol{X}}_l,$$

which satisfies $\boldsymbol{P}_l^2 = \boldsymbol{P}_l$ and $\boldsymbol{P}_l^\top = \boldsymbol{P}_l$.

By the construction of $\boldsymbol{W}_l^*(\alpha_l)$, we have the exact identity

$$\boldsymbol{W}_l^*(\alpha_l) \widehat{\boldsymbol{X}}_l = \boldsymbol{W}_l \widehat{\boldsymbol{X}}_l + \alpha_l \boldsymbol{W}_l \boldsymbol{\delta}_l \boldsymbol{P}_l.$$

Therefore, using $\widehat{\boldsymbol{W}}_l^{\mathrm{QEP}} = \boldsymbol{W}_l^*(\alpha_l) + \boldsymbol{E}_l$,

$$\begin{aligned}
\boldsymbol{R}_l^{\mathrm{QEP}} &= \boldsymbol{W}_l\boldsymbol{X}_l - (\boldsymbol{W}_l^*(\alpha_l) + \boldsymbol{E}_l)\widehat{\boldsymbol{X}}_l \\
&= \boldsymbol{W}_l(\widehat{\boldsymbol{X}}_l + \boldsymbol{\delta}_l) - \boldsymbol{W}_l^*(\alpha_l)\widehat{\boldsymbol{X}}_l - \boldsymbol{E}_l\widehat{\boldsymbol{X}}_l \\
&= \boldsymbol{W}_l\boldsymbol{\delta}_l - \alpha_l\boldsymbol{W}_l\boldsymbol{\delta}_l\boldsymbol{P}_l - \boldsymbol{E}_l\widehat{\boldsymbol{X}}_l \\
&= \boldsymbol{W}_l\boldsymbol{\delta}_l(\boldsymbol{I} - \alpha_l\boldsymbol{P}_l) - \boldsymbol{E}_l\widehat{\boldsymbol{X}}_l.
\end{aligned}$$

Consider the BASE residual at the same layer evaluated on the same $\widehat{\boldsymbol{X}}_l$:

$$\widetilde{\boldsymbol{R}}_l^{\mathrm{BASE}} \coloneqq \boldsymbol{W}_l\boldsymbol{X}_l - (\boldsymbol{W}_l + \boldsymbol{E}_l)\widehat{\boldsymbol{X}}_l = \boldsymbol{W}_l\boldsymbol{\delta}_l - \boldsymbol{E}_l\widehat{\boldsymbol{X}}_l.$$

Hence

$$\boldsymbol{R}_l^{\mathrm{QEP}} = \widetilde{\boldsymbol{R}}_l^{\mathrm{BASE}} - \alpha_l\boldsymbol{W}_l\boldsymbol{\delta}_l\boldsymbol{P}_l.$$

Since $\boldsymbol{P}_l$ is an orthogonal projection and $0 \le \alpha_l \le 1$, Lemma B.8 implies

$$\|\boldsymbol{W}_l\boldsymbol{\delta}_l(\boldsymbol{I} - \alpha_l\boldsymbol{P}_l)\|_F \le \|\boldsymbol{W}_l\boldsymbol{\delta}_l\|_F.$$

Therefore, by the triangle inequality,

$$\|\boldsymbol{R}_l^{\mathrm{QEP}}\|_F = \|\boldsymbol{W}_l\boldsymbol{\delta}_l(\boldsymbol{I} - \alpha_l\boldsymbol{P}_l) - \boldsymbol{E}_l\widehat{\boldsymbol{X}}_l\|_F \le \|\widetilde{\boldsymbol{R}}_l^{\mathrm{BASE}}\|_F.$$

Finally, since $\widehat{\boldsymbol{X}}_l^{\mathrm{BASE}}$ is the BASE activation produced by the BASE recursion, $\boldsymbol{R}_l^{\mathrm{BASE}}$ is exactly $\widetilde{\boldsymbol{R}}_l^{\mathrm{BASE}}$ evaluated at $\widehat{\boldsymbol{X}}_l^{\mathrm{BASE}}$. Thus, taking $\widehat{\boldsymbol{X}}_l = \widehat{\boldsymbol{X}}_l^{\mathrm{BASE}}$ in the above inequality yields Eq. (17). Summing with nonnegative weights $(\prod_{s=k+1}^{L}\gamma_s\|\boldsymbol{W}_s\|_2)\gamma_k$ yields $\mathcal{U}^{\mathrm{QEP}} \le \mathcal{U}^{\mathrm{BASE}}$, thus completing the proof. □

We further demonstrate that the final quantization error decreases monotonically as each propagation strength parameter α_l approaches 1.

Corollary B.6. *Fix the layer index $l \in \{1, \dots, L\}$ and assume that $\widehat{\boldsymbol{H}}_l \coloneqq \widehat{\boldsymbol{X}}_l\widehat{\boldsymbol{X}}_l^\top$ is invertible. Let the activation mismatch be $\boldsymbol{\delta}_l \coloneqq \boldsymbol{X}_l - \widehat{\boldsymbol{X}}_l$. Define the orthogonal projection*

$$\boldsymbol{P}_l \coloneqq \widehat{\boldsymbol{X}}_l^\top(\widehat{\boldsymbol{X}}_l\widehat{\boldsymbol{X}}_l^\top)^{-1}\widehat{\boldsymbol{X}}_l \in \mathbb{R}^{m\times m}.$$

For $\alpha \in [0, 1]$, define the QEP corrected weight in the continuous domain as

$$\boldsymbol{W}_l^*(\alpha) \coloneqq \boldsymbol{W}_l + \alpha\boldsymbol{W}_l\boldsymbol{\delta}_l\widehat{\boldsymbol{X}}_l^\top\widehat{\boldsymbol{H}}_l^{-1},$$

and let $\widehat{\boldsymbol{W}}_l(\alpha) \coloneqq \boldsymbol{W}_l^*(\alpha) + \boldsymbol{E}_l$ be for some fixed matrix $\boldsymbol{E}_l$. Then, for any $0 \le \alpha' \le \alpha \le 1$,

$$\|\boldsymbol{W}_l\boldsymbol{\delta}_l(\boldsymbol{I} - \alpha\boldsymbol{P}_l)\|_F \le \|\boldsymbol{W}_l\boldsymbol{\delta}_l(\boldsymbol{I} - \alpha'\boldsymbol{P}_l)\|_F. \tag{18}$$

Moreover, the pre-activation residual satisfies the exact identity

$$\boldsymbol{W}_l\boldsymbol{X}_l - \widehat{\boldsymbol{W}}_l(\alpha)\widehat{\boldsymbol{X}}_l = \boldsymbol{W}_l\boldsymbol{\delta}_l(\boldsymbol{I} - \alpha\boldsymbol{P}_l) - \boldsymbol{E}_l\widehat{\boldsymbol{X}}_l, \tag{19}$$

and hence the following upper bound is also monotone in α:

$$\|\boldsymbol{W}_l\boldsymbol{X}_l - \widehat{\boldsymbol{W}}_l(\alpha)\widehat{\boldsymbol{X}}_l\|_F \le \|\boldsymbol{W}_l\boldsymbol{\delta}_l(\boldsymbol{I} - \alpha\boldsymbol{P}_l)\|_F + \|\boldsymbol{E}_l\|_2\|\widehat{\boldsymbol{X}}_l\|_F. \tag{20}$$

Proof. First, $\boldsymbol{P}_l$ is an orthogonal projection. Indeed,

$$\boldsymbol{P}_l^\top = \widehat{\boldsymbol{X}}_l^\top(\widehat{\boldsymbol{X}}_l\widehat{\boldsymbol{X}}_l^\top)^{-1}\widehat{\boldsymbol{X}}_l = \boldsymbol{P}_l,$$

and

$$\boldsymbol{P}_l^2 = \widehat{\boldsymbol{X}}_l^\top(\widehat{\boldsymbol{X}}_l\widehat{\boldsymbol{X}}_l^\top)^{-1}\underbrace{\widehat{\boldsymbol{X}}_l\widehat{\boldsymbol{X}}_l^\top}_{=\widehat{\boldsymbol{H}}_l}(\widehat{\boldsymbol{X}}_l\widehat{\boldsymbol{X}}_l^\top)^{-1}\widehat{\boldsymbol{X}}_l = \boldsymbol{P}_l.$$

Hence, Lemma B.8 applies with $\boldsymbol{Z} \coloneqq \boldsymbol{W}_l\boldsymbol{\delta}_l$ and $\boldsymbol{P} \coloneqq \boldsymbol{P}_l$, yielding

$$\|\boldsymbol{W}_l\boldsymbol{\delta}_l(\boldsymbol{I} - \alpha\boldsymbol{P}_l)\|_F \le \|\boldsymbol{W}_l\boldsymbol{\delta}_l(\boldsymbol{I} - \alpha'\boldsymbol{P}_l)\|_F,$$

which proves Eq. (18).

Next, using the definitions of $\boldsymbol{W}_l^*(\alpha)$ and $\widehat{\boldsymbol{H}}_l^{-1} = (\widehat{\boldsymbol{X}}_l\widehat{\boldsymbol{X}}_l^\top)^{-1}$,

$$\boldsymbol{W}_l^*(\alpha)\widehat{\boldsymbol{X}}_l = \boldsymbol{W}_l\widehat{\boldsymbol{X}}_l + \alpha\boldsymbol{W}_l\boldsymbol{\delta}_l\widehat{\boldsymbol{X}}_l^\top(\widehat{\boldsymbol{X}}_l\widehat{\boldsymbol{X}}_l^\top)^{-1}\widehat{\boldsymbol{X}}_l = \boldsymbol{W}_l\widehat{\boldsymbol{X}}_l + \alpha\boldsymbol{W}_l\boldsymbol{\delta}_l\boldsymbol{P}_l.$$

Therefore, since $\widehat{\boldsymbol{W}}_l(\alpha) = \boldsymbol{W}_l^*(\alpha) + \boldsymbol{E}_l$ and $\boldsymbol{X}_l = \widehat{\boldsymbol{X}}_l + \boldsymbol{\delta}_l$,

$$\begin{aligned}\boldsymbol{W}_l\boldsymbol{X}_l - \widehat{\boldsymbol{W}}_l(\alpha)\widehat{\boldsymbol{X}}_l &= \boldsymbol{W}_l(\widehat{\boldsymbol{X}}_l + \boldsymbol{\delta}_l) - (\boldsymbol{W}_l^*(\alpha) + \boldsymbol{E}_l)\widehat{\boldsymbol{X}}_l \\ &= \boldsymbol{W}_l\boldsymbol{\delta}_l - \alpha\boldsymbol{W}_l\boldsymbol{\delta}_l\boldsymbol{P}_l - \boldsymbol{E}_l\widehat{\boldsymbol{X}}_l \\ &= \boldsymbol{W}_l\boldsymbol{\delta}_l(\boldsymbol{I} - \alpha\boldsymbol{P}_l) - \boldsymbol{E}_l\widehat{\boldsymbol{X}}_l,\end{aligned}$$

which is (19). Finally, (20) follows from the triangle inequality and $\|\boldsymbol{E}_l\widehat{\boldsymbol{X}}_l\|_F \le \|\boldsymbol{E}_l\|_2\|\widehat{\boldsymbol{X}}_l\|_F$. □

B.4 Relationship of QEP Correction and Ridge Regularization

We formally establish a rigorous mathematical connection between the Quantization Error Propagation (QEP) correction parameter α_l and the ridge regularization parameter λ. We show that tuning the QEP parameter α_l is equivalent to adjusting the strength of ridge regularization with parameter λ. We prove the monotone inverse relationship between these two parameters.

Proposition B.7. *The QEP update with mixing factor $\alpha_l \in [0, 1]$ is*

$$\widehat{\boldsymbol{W}}_l^*(\alpha_l) = \boldsymbol{W}_l(\boldsymbol{I} + \alpha\boldsymbol{\delta}_l\widehat{\boldsymbol{X}}_l^\top\widehat{\boldsymbol{H}}_l^{-1}).$$

The unique minimizer of the ridge objective

$$\min_{\widehat{\boldsymbol{W}}_l \in \mathbb{R}^{n_l \times d_l}} f(\widehat{\boldsymbol{W}}_l), \;\; f(\widehat{\boldsymbol{W}}_l) = \|\boldsymbol{W}_l\boldsymbol{X}_l - \widehat{\boldsymbol{W}}_l\widehat{\boldsymbol{X}}_l\|_F^2 + \lambda_l\|\boldsymbol{W}_l - \widehat{\boldsymbol{W}}_l\|_F^2, \;\; \lambda_l \ge 0,$$

equals

$$\widehat{\boldsymbol{W}}_l^*(\lambda_l) = \boldsymbol{W}_l\left(\boldsymbol{I} + \boldsymbol{\delta}_l\widehat{\boldsymbol{X}}_l^\top(\widehat{\boldsymbol{H}}_l + \lambda\boldsymbol{I})^{-1}\right). \tag{21}$$

Let the positive definite matrices be

$$\boldsymbol{G}(\alpha_l) := \alpha\hat{\boldsymbol{H}}_l^{-1}, \quad \boldsymbol{R}(\lambda_l) := (\widehat{\boldsymbol{H}}_l + \lambda\boldsymbol{I})^{-1}.$$

Then

$$\alpha_1 \le \alpha_2 \Rightarrow \boldsymbol{G}(\alpha_1) \preceq \boldsymbol{G}(\alpha_2), \;\; \lambda_1 \le \lambda_2 \Rightarrow \boldsymbol{R}(\lambda_1) \succeq \boldsymbol{R}(\lambda_2),$$

and the scalar mapping as follows:

$$\alpha(\lambda) := \frac{1}{d}\mathrm{Tr}\widehat{\boldsymbol{H}}_l\boldsymbol{R}(\lambda) = \frac{1}{d_l}\sum_{i=1}^{d_l}\frac{\gamma_i}{\gamma_i + \lambda_l}$$

with $\gamma_1 \ge \cdots \ge \gamma_{d_1} > 0$ the eigenvalues of $\widehat{\boldsymbol{H}}_l$, is strictly decreasing, satisfies $\alpha(0) = 1$ and $\lim_{\lambda\to\infty}\alpha(\lambda) = 0$, and obeys

$$\mathrm{Tr}\widehat{\boldsymbol{H}}_l\boldsymbol{G}(\alpha(\lambda)) = \mathrm{Tr}\widehat{\boldsymbol{H}}_l\boldsymbol{R}(\lambda).$$

Thus, decreasing λ from $+\infty$ to 0 corresponds to increasing α_l from 0 to 1.

Proof. A standard differential identity $\partial\|\boldsymbol{A}\|_F^2 = 2\boldsymbol{A}$ gives

$$\nabla_{\widehat{\boldsymbol{W}}_l} f(\widehat{\boldsymbol{W}}_l) = 2\left(\widehat{\boldsymbol{W}}_l\widehat{\boldsymbol{H}}_l - \boldsymbol{W}_l\boldsymbol{X}_l\widehat{\boldsymbol{X}}_l^\top\right) + 2\lambda(\widehat{\boldsymbol{W}}_l - \boldsymbol{W}_l).$$

Setting this gradient **0** yields

$$\widehat{\boldsymbol{W}}_l(\widehat{\boldsymbol{H}}_l + \lambda\boldsymbol{I}) = \boldsymbol{W}_l(\boldsymbol{X}_l\widehat{\boldsymbol{X}}_l^\top + \lambda\boldsymbol{I}),$$

and right multiplication by inverse of $\widehat{\boldsymbol{H}}_l + \lambda\boldsymbol{I}$ produces Eq. (21). Convexity of f ensures uniqueness.

Diagonalise $\widehat{\boldsymbol{H}}_l = \boldsymbol{U}\boldsymbol{\Gamma}\boldsymbol{U}^\top$ with $\Gamma = \mathrm{diag}(\gamma_1, \ldots, \gamma_{d_l})$. Then $\boldsymbol{G}(\alpha_l) = \boldsymbol{U}\alpha_l\boldsymbol{\Gamma}^{-1}\boldsymbol{U}^\top$ has eigenvalues $^{\alpha_l}/_{\gamma_l}$, which increase strictly with α_l, while $\boldsymbol{R}(\lambda_l) = \boldsymbol{U}(\boldsymbol{\Gamma} + \lambda\boldsymbol{I})^{-1}\boldsymbol{U}^\top$ has eigenvalues $^1/_{\gamma_i+\lambda}$, which decrease with λ_l, which means Loewner relations follow directly.

Furthermore, the following equation holds:

$$\alpha(\lambda) = \frac{1}{d_1}\mathrm{Tr}(\widehat{\boldsymbol{H}}_l \boldsymbol{R}(\lambda)) = \frac{1}{d_l}\sum_{i=1}^{d_1} \frac{\gamma_i}{\gamma_i + \lambda}$$

Each summand has derivative

$$\frac{\partial}{\partial \lambda}\frac{\gamma_i}{\gamma_i + \lambda} = -\frac{\gamma_i}{(\gamma_i + \lambda)^2} < 0,$$

which means $\alpha'(\lambda) < 0, \forall \lambda \geq 0$. Thus, $\alpha(\cdot)$ is strictly decreasing on $[0, \infty)$. One has

$$\lim_{\lambda \to 0} \frac{\gamma_i}{\gamma_i + \lambda} = 1, \quad \lim_{\lambda \to \infty} \frac{\gamma_i}{\gamma_i + \lambda} = 0.$$

$\alpha(\cdot)$ is strictly decreasing from 1 to 0 and smooth on $[0, \infty)$. Because α is continous, strictly decreasing, it is a bijection from $[0, +\infty)$ onto $(0, 1]$. By construction

$$\mathrm{Tr}\widehat{\boldsymbol{H}}_l \boldsymbol{G}(\alpha(\lambda)) = \mathrm{Tr}\widehat{\boldsymbol{H}}_l \boldsymbol{R}(\lambda).$$

Thus, decreasing λ from $+\infty$ to 0 corresponds to increasing α_l from 0 to 1.

□

B.5 Technical Lemma

Lemma B.8. *Let $\boldsymbol{Z} \in \mathbb{R}^{m \times n}$ be arbitrary, and let $P \in \mathbb{R}^{n \times n}$ be an orthogonal projection, i.e., $\boldsymbol{P}^2 = \boldsymbol{P}$ and $\boldsymbol{P}^\top = \boldsymbol{P}$. For every pair $0 \leq \alpha' \leq \alpha \leq 1$,*

$$\|\boldsymbol{Z}(\boldsymbol{I} - \alpha \boldsymbol{P})\|_F \leq \|\boldsymbol{Z}(I - \alpha' \boldsymbol{P})\|_F \leq \|\boldsymbol{Z}\|_F. \tag{22}$$

Proof. Write $f(\alpha) := \|\boldsymbol{Z}(\boldsymbol{I} - \alpha \boldsymbol{P})\|_F^2$. Because $\boldsymbol{P}^\top = \boldsymbol{P}$ and $\boldsymbol{P}^2 = \boldsymbol{P}$,

$$f(\alpha) = \mathrm{Tr}\left[(\boldsymbol{I} - \alpha \boldsymbol{P})\boldsymbol{Z}^\top \boldsymbol{Z}(\boldsymbol{I} - \alpha \boldsymbol{P})\right] = \|\boldsymbol{Z}\|_F^2 - 2\alpha(1-\alpha)\underbrace{\mathrm{Tr}(\boldsymbol{Z}^\top \boldsymbol{Z}\boldsymbol{P})}_{t \geq 0}.$$

Thus, $f'(\alpha) = -(2-\alpha)t \leq 0$ on $[0, 1]$ indicates that $f(\alpha)$ is non-increasing. Taking square roots yields the first inequality in Eq. (22). Setting $\alpha' = 0$ yields the second inequality: $\|\boldsymbol{Z}(\boldsymbol{I} - \alpha \boldsymbol{P})\|_F \leq \|\boldsymbol{Z}\|_F$. □

C Additional Implementation Details

C.1 Damping for Hessian

A standard numerical issue in PTQ arises when the Hessian matrix $\widehat{\boldsymbol{H}}_l$ is ill-conditioned or singular, rendering its inversion unstable or undefined. Following GPTQ [Frantar et al., 2022], we resolve this issue by employing a damping strategy that adds a small scalar value λ to the diagonal elements of $\widehat{\boldsymbol{H}}_l$ to ensure positive definiteness. In our implementation, we set λ to the mean of the diagonal elements of $\widehat{\boldsymbol{H}}_l$, providing a straightforward yet effective method to stabilize the inversion process.

D Additional Experiments

D.1 Additional Perplexity Results

Due to space constraints, the main text reports perplexity results solely for the WikiText-2 dataset. Here, we provide additional results for PTB (Table 6) and C4 (Table 7), along with extended results for WikiText-2 (Table 5). These supplementary results further validate that QEP consistently enhances PTQ performance, particularly in low-bit quantization scenarios.

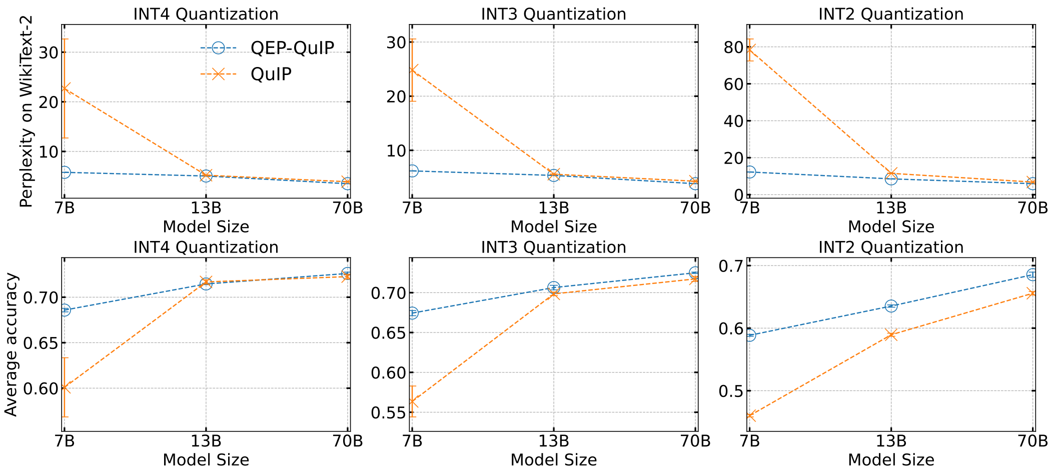

Figure 3: Results averaged over 5 random seeds comparing QuIP with and without QEP across different quantization levels. Each subplot shows results for INT4, INT3, and INT2 quantization, respectively, with the horizontal axis indicating model size (7B, 13B, 70B). The top row reports perplexity on WikiText-2 (lower is better), while the bottom row shows the average of normalized accuracy scores on ARC (easy), PIQA, and StoryCloze benchmarks (higher is better), representing generalization capability. Error bars represent the standard error of the mean (SEM). Models using QEP-QuIP consistently outperform or match the performance of baseline QuIP, especially under more aggressive quantization (INT3 and INT2).

D.2 Detailed Accuracy Results for Individual Tasks

Due to space limitations, the main text reports only the average accuracy across three tasks. Here, we provide task-specific accuracies for PIQA (Table 8), StoryCloze (Table 9), and ARC-Easy (Table 10), further confirming that QEP consistently improves layer-wise PTQ.

D.3 Stability of QuIP Results Across Random Seeds

We assess the stability of QuIP-only experiments by averaging five independent runs per configuration. Model sizes, quantization levels, and benchmarks align with the main Experiments section. Figure 3 plots QuIP with or without QEP at three quantization levels. Each marker is the mean of five seeds, and the error bars show the standard error of the mean. The top row gives perplexity on WikiText 2; the bottom row reports mean normalized accuracy on ARC easy, PIQA, and StoryCloze. Seed-to-seed variation is small and does not change the main conclusions. QEP-QuIP keeps its advantage, especially at INT3 and INT2. The main text lists the best seed per configuration for consistency with past work. This appendix confirms that the gains are not seed-specific but robust and reproducible, supporting using QEP.

D.4 Comparison with OmniQuant Baseline

For completeness, we compare QEP-enhanced *layer-wise* PTQ with *block-wise* OmniQuant [Shao et al., 2023] on LLaMA-2-7B using WikiText-2 perplexity; lower values indicate better performance. As shown in Table 11, QuIP+QEP achieves the lowest perplexity at INT4/INT3 and remains stable at INT2, whereas OmniQuant diverges. These findings align with recent PTQ benchmarks that indicate OmniQuant's underperformance relative to layer-wise PTQ [Zhao et al., 2025].

Table 11: WikiText-2 perplexity for LLaMA-2-7B at different bit-widths. NaN denotes divergence.

Method	INT4	INT3	INT2
RTN+QEP	6.017	17.309	97153.266
GPTQ+QEP	5.933	7.898	7214.328
AWQ+QEP	5.756	11.131	229888.406
QuIP+QEP	**5.753**	**6.154**	**11.972**
OmniQuant	5.880	7.065	NaN

Table 5: Perplexities (↓) on WikiText-2 for Llama-2 (7B, 13B, 70B) under eight quantization settings.

Bits	Method	QEP	Llama-2-7B	Llama-2-13B	Llama-2-70B
INT4g128	RTN	✗	5.726	4.984	3.463
		✓	**5.687**	**4.966**	**3.431**
	GPTQ	✗	5.698	4.987	3.419
		✓	**5.609**	**4.969**	**3.416**
	AWQ	✗	5.599	4.987	3.408
		✓	**5.580**	**4.969**	**3.404**
INT4	RTN	✗	6.116	5.206	3.672
		✓	**6.017**	**5.165**	**3.621**
	GPTQ	✗	6.083	5.167	3.594
		✓	**5.933**	**5.127**	**3.576**
	AWQ	✗	5.831	5.064	3.484
		✓	**5.756**	**5.041**	**3.479**
INT3g128	RTN	✗	6.662	5.518	3.978
		✓	**6.330**	**5.412**	**3.882**
	GPTQ	✗	6.411	5.459	3.880
		✓	**6.160**	**5.358**	**3.838**
	AWQ	✗	6.247	5.315	3.740
		✓	**6.108**	**5.295**	**3.724**
INT3	RTN	✗	539.866	10.688	7.530
		✓	**17.309**	**7.458**	**5.648**
	GPTQ	✗	10.881	6.632	4.860
		✓	**7.898**	**6.245**	**4.102**
	AWQ	✗	15.299	6.448	4.362
		✓	**11.131**	**6.092**	**4.103**
INT2g32	RTN	✗	90.692	10.563	6.802
		✓	**12.249**	**7.920**	**5.869**
	GPTQ	✗	12.023	8.394	5.621
		✓	**9.245**	**7.362**	**5.445**
	AWQ	✗	15887.204	106933.227	63663.707
		✓	**51.874**	**80654.797**	**37096.516**
INT2g64	RTN	✗	431.595	26.220	10.312
		✓	**19.371**	**9.917**	**6.992**
	GPTQ	✗	278.302	11.584	6.546
		✓	**14.737**	**8.685**	**6.030**
	AWQ	✗	**217111.860**	**121737.148**	**71703.781**
		✓	241136.594	126944.578	74227.539
INT2g128	RTN	✗	4270.828	122.063	27.268
		✓	**35.291**	**12.779**	**8.799**
	GPTQ	✗	43.915	**16.653**	8.123
		✓	**17.886**	19.952	**6.825**
	AWQ	✗	**222344.250**	**122795.898**	**72446.680**
		✓	247751.203	126813.172	74192.570
INT2	RTN	✗	**17783.918**	**51152.832**	26077.172
		✓	97153.266	61158.555	**26063.672**
	GPTQ	✗	13051.469	**1301.395**	107.458
		✓	**7214.328**	2782.353	**52.472**
	AWQ	✗	**199448.797**	93036.517	**81834.344**
		✓	229888.406	**74735.836**	88684.156

Table 6: Perplexities (↓) on PTB for Llama-2 (7B, 13B, 70B) under eight quantization settings. "N/A" denotes numerical overflow (NaN).

Bits	Method	QEP	Llama-2-7B	Llama-2-13B	Llama-2-70B
INT4g128	RTN	✗	61.750	53.835	**24.146**
		✓	**47.798**	**49.503**	24.604
	GPTQ	✗	N/A	51.133	**24.101**
		✓	N/A	**50.072**	24.243
	AWQ	✗	43.894	**53.863**	**24.525**
		✓	**40.445**	55.345	24.554
INT4	RTN	✗	82.641	60.749	23.545
		✓	**50.168**	**53.117**	**23.346**
	GPTQ	✗	N/A	53.561	24.720
		✓	**124291.961**	**53.537**	**24.149**
	AWQ	✗	60.261	**56.152**	25.542
		✓	**46.937**	57.445	**24.411**
INT3g128	RTN	✗	55.467	64.638	**23.586**
		✓	**48.576**	**54.866**	24.776
	GPTQ	✗	N/A	**57.079**	**24.091**
		✓	N/A	62.083	24.092
	AWQ	✗	64.932	**57.273**	**24.668**
		✓	**52.356**	61.479	26.309
INT3	RTN	✗	**37167.801**	294.802	64.002
		✓	5514.820	**113.856**	**34.212**
	GPTQ	✗	44807.926	106.715	27.839
		✓	N/A	**81.117**	**27.469**
	AWQ	✗	130.308	121.698	26.887
		✓	**81.606**	**93.260**	**25.592**
INT2g32	RTN	✗	20280.412	262.244	63.428
		✓	**1685.683**	**96.913**	**36.677**
	GPTQ	✗	18292.635	152.169	**29.163**
		✓	N/A	**110.507**	30.465
	AWQ	✗	47850.137	60977.195	48520.398
		✓	**3741.642**	**47591.414**	**20185.246**
INT2g64	RTN	✗	9252.538	551.510	153.528
		✓	**1096.720**	**158.306**	**42.991**
	GPTQ	✗	N/A	275.949	**37.024**
		✓	N/A	**187.477**	37.384
	AWQ	✗	**202939.484**	**113584.867**	**79866.031**
		✓	220728.234	117658.867	82598.511
INT2g128	RTN	✗	9685.755	1213.282	767.896
		✓	**4462.478**	**207.651**	**63.806**
	GPTQ	✗	10694.694	395.689	56.685
		✓	N/A	**325.407**	**45.569**
	AWQ	✗	**202164.484**	**113784.242**	**80543.727**
		✓	222388.375	117059.742	82493.251
INT2	RTN	✗	31824.279	**42619.883**	**26063.672**
		✓	**10824.680**	55286.305	26077.172
	GPTQ	✗	N/A	3868.426	2438.034
		✓	N/A	**3850.578**	**4050.844**
	AWQ	✗	**183984.766**	87673.695	**90442.352**
		✓	198744.750	**62160.063**	91939.883

Table 7: Perplexities (↓) on C4 for Llama-2 (7B, 13B, 70B) under eight quantization settings.

Bits	Method	QEP	Llama-2-7B	Llama-2-13B	Llama-2-70B
INT4g128	RTN	✗	7.584	6.869	5.826
		✓	**7.513**	**6.839**	**5.786**
	GPTQ	✗	7.522	6.860	5.778
		✓	**7.421**	**6.828**	**5.770**
	AWQ	✗	7.443	6.840	5.772
		✓	**7.416**	**6.829**	**5.767**
INT4	RTN	✗	8.165	7.146	6.012
		✓	**7.945**	**7.067**	**5.947**
	GPTQ	✗	7.866	7.069	5.905
		✓	**7.719**	**6.998**	**5.880**
	AWQ	✗	7.721	6.962	5.842
		✓	**7.634**	**6.932**	**5.828**
INT3g128	RTN	✗	8.977	7.582	6.266
		✓	**8.510**	**7.402**	**6.150**
	GPTQ	✗	8.502	7.463	6.105
		✓	**8.185**	**7.316**	**6.072**
	AWQ	✗	8.300	7.310	6.036
		✓	**8.105**	**7.264**	**6.019**
INT3	RTN	✗	524.279	13.883	10.886
		✓	**21.436**	**10.284**	**8.202**
	GPTQ	✗	11.780	8.826	7.067
		✓	**9.950**	**8.429**	**6.869**
	AWQ	✗	17.418	9.049	6.631
		✓	**13.934**	**8.257**	**6.353**
INT2g32	RTN	✗	225.440	13.879	9.720
		✓	**16.148**	**10.561**	**8.459**
	GPTQ	✗	14.365	10.719	7.932
		✓	**11.839**	**9.685**	**7.717**
	AWQ	✗	9028.133	76591.883	57596.215
		✓	**51.811**	**49645.738**	**33026.816**
INT2g64	RTN	✗	553.766	30.445	15.155
		✓	**22.089**	**12.762**	**9.850**
	GPTQ	✗	20.860	13.394	8.981
		✓	**14.084**	**11.039**	**8.508**
	AWQ	✗	**164477.422**	**95241.625**	**64913.477**
		✓	181582.719	98917.820	67203.359
INT2g128	RTN	✗	4811.772	131.665	47.878
		✓	**34.022**	**15.398**	**12.081**
	GPTQ	✗	33.370	18.008	10.535
		✓	**18.184**	**12.704**	**9.433**
	AWQ	✗	**168465.266**	**95617.305**	**65646.594**
		✓	187329.625	98457.031	67248.492
INT2	RTN	✗	**28258.385**	**52642.387**	**24912.074**
		✓	108424.680	71050.250	29042.623
	GPTQ	✗	3048.671	**299.684**	56.719
		✓	**276.638**	629.527	**30.874**
	AWQ	✗	**156266.797**	81233.602	**73251.945**
		✓	177576.750	**64098.504**	75607.211

Table 8: Accuracy (↑) on PIQA for Llama-2 (7B, 13B, 70B) under eight quantization settings.

Bits	Method	QEP	Llama-2-7B	Llama-2-13B	Llama-2-70B
INT4g128	RTN	✗	**0.773**	**0.792**	0.804
		✓	**0.773**	0.790	**0.806**
	GPTQ	✗	0.770	0.789	**0.807**
		✓	**0.771**	**0.792**	0.806
	AWQ	✗	**0.768**	0.790	0.807
		✓	0.764	**0.791**	**0.810**
INT4	RTN	✗	0.763	**0.789**	0.811
		✓	**0.767**	0.788	**0.812**
	GPTQ	✗	0.755	**0.789**	0.804
		✓	**0.761**	0.787	**0.811**
	AWQ	✗	0.760	**0.789**	0.807
		✓	**0.763**	0.784	**0.814**
INT3g128	RTN	✗	0.757	0.770	0.793
		✓	**0.761**	**0.779**	**0.806**
	GPTQ	✗	0.758	0.778	0.806
		✓	**0.764**	**0.782**	**0.807**
	AWQ	✗	0.760	**0.780**	**0.805**
		✓	**0.765**	**0.780**	**0.805**
INT3	RTN	✗	0.563	0.705	0.724
		✓	**0.677**	**0.752**	**0.764**
	GPTQ	✗	0.720	0.757	0.783
		✓	**0.745**	**0.770**	**0.791**
	AWQ	✗	0.647	0.760	0.787
		✓	**0.725**	**0.770**	**0.801**
INT2g32	RTN	✗	0.588	0.696	0.760
		✓	**0.693**	**0.735**	**0.771**
	GPTQ	✗	0.690	0.732	0.772
		✓	**0.714**	**0.748**	**0.776**
	AWQ	✗	0.568	0.505	**0.503**
		✓	**0.702**	**0.514**	0.501
INT2g64	RTN	✗	0.597	0.614	0.714
		✓	**0.676**	**0.710**	**0.748**
	GPTQ	✗	0.647	0.705	0.745
		✓	**0.677**	**0.713**	**0.765**
	AWQ	✗	0.502	**0.506**	0.502
		✓	**0.702**	**0.506**	**0.504**
INT2g128	RTN	✗	0.511	0.566	0.635
		✓	**0.652**	**0.678**	**0.721**
	GPTQ	✗	0.581	0.639	0.715
		✓	**0.659**	**0.683**	**0.747**
	AWQ	✗	**0.501**	0.505	**0.503**
		✓	**0.501**	**0.507**	**0.503**
INT2	RTN	✗	0.509	0.493	0.499
		✓	**0.510**	**0.506**	0.510
	GPTQ	✗	**0.500**	**0.509**	0.511
		✓	0.493	**0.507**	**0.544**
	AWQ	✗	**0.507**	0.504	0.502
		✓	0.505	0.504	0.504

Table 9: Accuracy (↑) on StoryCloze for Llama-2 (7B, 13B, 70B) under eight quantization settings.

Bits	Method	QEP	Llama-2-7B	Llama-2-13B	Llama-2-70B
INT4g128	RTN	✗	0.765	0.785	0.791
		✓	**0.770**	**0.788**	**0.794**
	GPTQ	✗	0.768	0.784	0.793
		✓	**0.771**	**0.789**	**0.798**
	AWQ	✗	**0.777**	0.782	0.792
		✓	**0.777**	**0.785**	**0.798**
INT4	RTN	✗	0.756	**0.777**	0.796
		✓	**0.763**	**0.777**	**0.798**
	GPTQ	✗	0.765	**0.776**	**0.794**
		✓	**0.766**	0.775	0.792
	AWQ	✗	0.760	0.774	0.789
		✓	**0.766**	**0.777**	**0.794**
INT3g128	RTN	✗	0.749	0.766	**0.790**
		✓	**0.756**	**0.773**	0.789
	GPTQ	✗	**0.763**	**0.776**	0.793
		✓	0.759	0.770	**0.796**
	AWQ	✗	**0.761**	0.767	**0.795**
		✓	**0.761**	**0.782**	**0.795**
INT3	RTN	✗	0.546	0.669	0.738
		✓	**0.672**	**0.728**	**0.776**
	GPTQ	✗	0.722	0.752	0.780
		✓	**0.745**	**0.766**	**0.782**
	AWQ	✗	0.689	**0.767**	**0.787**
		✓	**0.702**	0.764	0.782
INT2g32	RTN	✗	0.645	0.668	0.745
		✓	**0.704**	**0.721**	**0.776**
	GPTQ	✗	0.758	0.715	0.724
		✓	**0.763**	**0.748**	**0.766**
	AWQ	✗	0.660	0.511	0.516
		✓	**0.703**	**0.570**	**0.569**
INT2g64	RTN	✗	0.607	0.617	0.718
		✓	**0.670**	**0.696**	**0.766**
	GPTQ	✗	0.654	0.686	0.756
		✓	**0.712**	**0.720**	**0.758**
	AWQ	✗	**0.476**	**0.479**	**0.476**
		✓	0.474	**0.479**	0.475
INT2g128	RTN	✗	0.509	0.577	0.647
		✓	**0.651**	**0.677**	**0.741**
	GPTQ	✗	0.588	0.634	0.724
		✓	**0.649**	**0.690**	**0.753**
	AWQ	✗	**0.475**	**0.478**	**0.476**
		✓	**0.475**	**0.478**	**0.476**
INT2	RTN	✗	0.468	**0.491**	**0.482**
		✓	**0.488**	0.487	**0.482**
	GPTQ	✗	0.485	0.501	0.539
		✓	**0.514**	**0.513**	**0.589**
	AWQ	✗	**0.489**	**0.478**	0.475
		✓	0.482	0.476	**0.477**

Table 10: Accuracy (↑) on ARC-Easy for Llama-2 (7B, 13B, 70B) under eight quantization settings.

Bits	Method	QEP	Llama-2-7B	Llama-2-13B	Llama-2-70B
INT4g128	RTN	✗	**0.554**	0.567	**0.596**
		✓	0.540	**0.572**	**0.596**
	GPTQ	✗	**0.531**	0.573	0.586
		✓	0.521	**0.579**	**0.592**
	AWQ	✗	**0.537**	0.577	0.585
		✓	0.526	**0.580**	**0.592**
INT4	RTN	✗	0.521	**0.582**	0.590
		✓	**0.524**	0.574	**0.593**
	GPTQ	✗	**0.525**	**0.575**	**0.594**
		✓	0.512	0.570	0.589
	AWQ	✗	0.529	0.572	0.580
		✓	**0.532**	**0.577**	**0.591**
INT3g128	RTN	✗	**0.528**	**0.569**	**0.575**
		✓	0.517	0.556	0.572
	GPTQ	✗	0.521	**0.568**	**0.580**
		✓	**0.515**	**0.568**	0.569
	AWQ	✗	**0.534**	**0.561**	**0.597**
		✓	0.527	**0.561**	0.592
INT3	RTN	✗	0.322	0.450	**0.459**
		✓	**0.391**	**0.485**	0.541
	GPTQ	✗	0.468	0.514	0.550
		✓	**0.474**	**0.520**	**0.551**
	AWQ	✗	0.416	0.539	0.588
		✓	**0.452**	**0.540**	**0.602**
INT2g32	RTN	✗	0.339	0.445	0.533
		✓	**0.426**	**0.474**	**0.557**
	GPTQ	✗	0.421	0.481	0.506
		✓	**0.441**	**0.486**	**0.547**
	AWQ	✗	0.352	0.272	**0.263**
		✓	**0.449**	**0.280**	**0.263**
INT2g64	RTN	✗	0.332	0.371	0.467
		✓	**0.390**	**0.430**	**0.557**
	GPTQ	✗	0.377	0.455	0.485
		✓	**0.404**	**0.458**	**0.548**
	AWQ	✗	**0.266**	**0.270**	0.262
		✓	0.265	**0.270**	**0.263**
INT2g128	RTN	✗	0.269	0.253	0.395
		✓	**0.376**	**0.407**	**0.479**
	GPTQ	✗	0.338	0.383	0.443
		✓	**0.367**	**0.418**	**0.508**
	AWQ	✗	**0.266**	**0.269**	0.260
		✓	0.265	**0.269**	**0.261**
INT2	RTN	✗	**0.265**	0.253	**0.263**
		✓	0.262	**0.264**	0.261
	GPTQ	✗	0.263	**0.256**	0.257
		✓	**0.272**	0.265	**0.281**
	AWQ	✗	**0.267**	**0.270**	**0.262**
		✓	0.262	**0.270**	0.261