Evaluating Full INT8 Quantization and Inference techniques for Causal Language Model

Prawin Kumar Srinivasa Kumar
MSc Embedded Systems
s2731991

***Abstract*—Deploying large language models (LLMs) onto embedded devices presents significant challenges due to their substantial storage and computational requirements. Integer quantization has emerged as a critical technique to address these constraints, with recent research focusing on reducing model complexity while maintaining inference performance. This work investigates a static quantization approach for transformer-based LLMs, targeting full integer (INT8) representation across both linear and non-linear layers. Unlike existing approaches that primarily target linear layer quantization, our methodology develops a holistic quantization strategy that systematically addresses precision limitations across transformer architectures. This involves mapping both the full-precision weights and activations to 8-bit integer representations. Through this, we demonstrate a novel approach to evaluate different transformer-based quantization techniques on causal language models to understand their impact on performance degradation. The research employs the usage of both the quantization techniques used out of the box from model optimization toolkits in addition to the approximations for quantization as explored in literature and evaluates model performance through standard NLP benchmark tasks, with particular emphasis on text generation capabilities supported by the perplexity measurements. The proposed approach provides a detailed comparative analysis of static integer quantization techniques, exploring the trade-offs between model compression and inference accuracy. Through experimental validation, we illustrate the feasibility of deploying sophisticated small language models on resource-constrained platforms, offering insights into advanced quantization methodologies for embedded AI systems. Our experimental results demonstrate that applying INT8 quantization approximations to causal language models significantly degrades their performance. We observed an average relative decrease of approximately 15 percentage points across most NLP benchmark tasks. More critically, the models' autoregressive capabilities were severely compromised, with perplexity scores reaching orders of 8e4. These results indicate that post-training quantization fundamentally disrupts the models' ability to generate coherent token sequences.**

***Index Terms*—INT8 Quantization, Non-Linear Layers Approximations, INT8 Inference, Transformers, Large Language Models**

I. Introduction

Large Language Models (LLMs) have revolutionized natural language processing, achieving unprecedented performance across complex cognitive tasks, from machine translation to sophisticated reasoning. Despite their remarkable capabilities, deploying these models on resource-constrained embedded systems remains a critical challenge due to their substantial computational and memory demands. Modern LLMs, often comprising billions of parameters, present significant barriers to deployment on edge devices with limited processing power, memory, and energy budgets. Static quantization emerges as a pivotal optimization technique to address these deployment constraints, offering a systematic approach to model compression that transforms high-precision floating-point representations into more efficient integer formats. Unlike dynamic quantization methods that introduce runtime computational variability wherein additional computations to determine quantization parameters during runtime are required [33], static quantization provides a deterministic and hardware-friendly mechanism for reducing model complexity. This approach enables precise pre-computation of quantization parameters, facilitating more predictable and energy-efficient inference on embedded platforms. The proposed research develops a static quantization framework for Large Language Models (LLMs), targeting full integer (INT8) inference across transformer architectures. The fundamental objective centres on mapping both model weights and activation functions to 8-bit integer representations while maintaining performance comparable to 32-bit floating-point (FP32) baselines, to reduce computational overhead and latency. When emphasis is placed on edge deployment, the current suite of microcontrollers has additional NPUs for accelerating inference. This particularly focuses on providing speedup where the system can offload 8-bit integer precision matrix multiplications. Therefore, it becomes crucial to quantize the model in order to execute more hardware-friendly operations that can be efficiently offloaded or mapped onto such accelerators. A typical transformer architecture presents intricate computational challenges that significantly complicate quantization efforts. The primary structural elements involve multi-head self-attention mechanisms, characterized by projections of query, key, and value matrices. These mechanisms incorporate linear transformations that map input embeddings, with Softmax normalization introducing non-linear characteristics that fundamentally resist straightforward integer quantization. Feed-forward neural network components further amplify quantization complexity. These layers comprise two sequential linear transformations interspersed with non-linear activation functions such as GELU. The inherent mathematical operations, particularly element-wise transformations and activation mappings, create pronounced precision sensitivity that challenges uniform quantization strategies. Layer normalization represents another critical bottleneck in the quantization process. This technique nor-

malizes layer inputs across feature dimensions through statistical computations involving mean and variance calculations. Such operations demonstrate extreme sensitivity to precision reduction, making them particularly problematic for integer-only implementations. The methodology diverges from existing quantization approaches by implementing a static quantization strategy that addresses both linear and non-linear layer transformations. Drawing inspiration from prior works such as I-BERT [16], I-VIT [23] and FQ-VIT [25], that provide the framework to quantize the non-linear layers present in transformer architecture by making use of hardware friendly approximations. It provides a straightforward approach to mitigating precision limitations inherent in transformer-based model architectures. The details regarding the implementation for reproducing the results from the experiments can be found in section V. The details regarding the approach followed through the course of this work can be followed in section IV. Finally, the results of the experiments can be identified in section VI.

A. Challenges and Contributions

Deploying large language models (LLMs) on embedded devices poses significant challenges, primarily due to their computational complexity and memory requirements. Quantization has emerged as a key strategy to address these issues, enabling efficient utilization of hardware for tasks such as real-time inference. Existing research predominantly focuses on encoder-based transformers like BERT for classification tasks. This leaves autoregressive causal language models which are predominantly used in chatbots like ChatGPT that provide massive generalizability benefits largely unexplored. The unique architecture and inference patterns of causal models, such as their dependence on sequential token generation and the resulting activation memory buildup, present distinct challenges for quantization. Further the non-linear computations present within the transformer in the layers of GELU, LayerNorm and Softmax like exponential, variance and mean are difficult to represent since they exhibit a wider dynamic range and are more susceptible to quantization errors, further complicating efforts to achieve full integer (INT8) inference.

- This work systematically examines the challenges of quantizing non-linear layers within transformer-based causal language models. Using static quantization methods, we evaluate approximation techniques from the works of: I-VIT [23], I-BERT [16]and FQ-VIT [25] implemented in both Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT) settings. These methods are applied to a sub-billion parameter model, OPT-125M [50], to assess their impact on model performance across various natural language processing tasks.
- Our findings highlight the trade-offs between computational efficiency and model performance. While QAT demonstrates the potential to recover some performance losses observed in PTQ, a significant gap persists between fully quantized models and their full-precision counterparts. This performance degradation is particularly pronounced in text generation tasks, where semantic consistency and causal reasoning capabilities are critically affected.
- Further analysis reveals that static quantization, though hardware-friendly, introduces challenges in representing non-linear operations accurately in integer formats. This is evident in the sensitivity of deeper layers to quantization-induced noise, as observed in metrics detailed in section VII. While quantization techniques such as I-VIT and FQ-VIT have demonstrated success in vision tasks, and I-BERT has shown promise in encoder-based language models, their adaptation to causal language models presents unique challenges that require thorough investigation, a gap this work aims to address. Although recent work like I-LLM [13] suggests that dynamic quantization outperforms static quantization approaches, their claims lack sufficient investigation and their closed implementation limits reproducibility.
- By critically evaluating integer inference techniques, this research provides insights into the practical deployment of large language models in constrained computing environments. Our analysis emphasizes the need for layer-wise customization of quantization strategies, mixed-precision approaches to mitigate sensitivity in critical layers, and the development of tailored hardware-specific optimizations.

B. Research Questions

Recent studies reveal a significant gap in applying full integer quantization, specifically static Full-INT8 quantization, for causal language models (CLMs). While existing literature introduces various approximation and quantization techniques for transformer-based models, their effectiveness for causal language models—specifically for sub-billion parameter models, which are easier for edge deployment—remains underexplored. Furthermore, aspects such as the selection of high-quality datasets, calibration methods for accurate quantization parameters, and the downstream impact on model performance are insufficiently addressed. These gaps highlight the need for systematic evaluation and optimization.

To address these challenges, this study investigates the following research questions:

Main Research Question: *What is the feasibility of employing approximation techniques to achieve static Full-INT8 quantization for sub-billion parameter causal language models?*

Objective: Evaluate the effectiveness of Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT) across different approximation methods to determine their suitability for efficient inference without significant performance degradation.

- **Sub-Research Question 1:** *How does the performance of these quantization techniques vary across different datasets, and what are their impacts on model representation, causality, and downstream task performance?*

Objective: Analyze how dataset diversity influences quantization performance, with a focus on maintaining model integrity and effectiveness in tasks such as language generation and comprehension.

- **Sub-Research Question 2:** *How can the impact of approximation techniques on model representation, causality, and downstream tasks be best explained?*
  Objective: Conduct quantization sensitivity analyses and ablation studies to explain how different quantization strategies affect model behaviour and decision-making processes.
- **Sub-Research Question 3:** *What strategies can effectively mitigate the performance loss observed in Post-Training Quantization (PTQ)?*
  Objective: Explore advanced calibration methods, including the use of improved calibration observers and dataset-driven calibration techniques, to recover or minimize approximation loss/error post-quantization.

II. Related Work

A. Post-Training Quantization (PTQ)

This technique has emerged as a critical approach to addressing the high computational and memory demands of large language models (LLMs), particularly given the prohibitive costs associated with fine-tuning-based quantization methods. PTQ is especially attractive for its simplicity and efficiency, particularly for sub-billion parameter models. Current quantization strategies typically target weights, activations, or Key-Value cache (KV-cache) components, each addressing specific computational bottlenecks encountered during LLM inference. Recent research, such as [21], has systematically explored weight-only, weight-activation, and KV-cache quantization techniques. Weight-only quantization accelerates General Matrix Vector Multiply (GEMV) operations during the decoding stage, while weight-activation quantization reduces compute overhead during General Matrix Multiply (GEMM) operations in the prefill stage. KV-cache quantization addresses memory requirements, which become particularly significant when processing large text sequences or substantial batch sizes.

Based on the findings from [21], model size significantly influences quantization tolerance. Smaller models (under 13 billion parameters) demonstrate minimal degradation ($< 2\%$) with W8, W8A8, and KV8 quantization, whereas larger models benefit from W4, W4A8, and KV4 quantization due to inherent parameter redundancies. Further optimizations into weight-only quantization techniques have expanded the quantization landscape. [24] introduced activation-aware quantization, preserving salient weights through per-channel scaling informed by activation distributions. [38] employed Omnidirectionally Calibrated Quantization with techniques like Learnable Weight Clipping and Learnable Equivalent Transformation to modulate weight and activation outliers. [46] developed a training-free INT8 quantization approach that adapts to activation outliers by strategically shifting values. However, it is important to point out out that this type of quantization still involves floating-point computations making it a mixed-precision model.

Advanced quantization strategies have also emerged for managing complex quantization scenarios. [20] implemented sequential column-wise quantization, optimizing configurations through grid search and retaining weak columns at higher precision. [11] utilized Optimal Brain Quantization with Cholesky decomposition for recursive block-wise quantization, addressing memory throughput bottlenecks in large LLMs. While these approaches demonstrate significant progress in weight quantization, persistent challenges remain in quantizing non-linear layers such as layer normalization and the Softmax operation within attention mechanisms. Although PTQ techniques effectively quantize the matrix multiplications in the Query-Key-Value (QKV) projection and the Query-Key (QK) product computations, the Softmax operation used in attention mechanisms is typically preserved at higher precision. This is due to its reliance on exponentiation and normalization, which are inherently resistant to straightforward integer approximations. The high dynamic range of Softmax outputs and its critical role in computing accurate attention weights exacerbate the difficulty of achieving hardware-friendly quantization without substantial performance loss.

The inherent complexity of quantizing these non-linear components represents a critical frontier in LLM inference optimization. Addressing these challenges through unique strategies on non-uniform mixed precision quantization or hardware-aware optimizations presents an opportunity to enhance model efficiency while maintaining competitive accuracy across diverse deployment scenarios.

B. Quantization Aware Training (QAT)

QAT integrates quantization into the training process, allowing models to adjust to quantization-induced errors during optimization. This approach has shown promise for large language models (LLMs) and vision transformers (ViTs), addressing some limitations of Post-Training Quantization (PTQ), particularly in handling non-linear layers and attention mechanisms. In [39], the authors utilized second-order Hessian information to implement mixed-precision quantization for BERT, achieving a five-fold compression with only a 2% accuracy loss. This method demonstrates the potential of Hessian-based optimization for precision allocation, ensuring balanced quantization across layers. However, the computational expense of calculating Hessian eigenvalues may limit scalability to larger models or datasets.

SQuAT [44] introduced a novel training strategy that alternates between sharpness-aware and quantization-aware objectives. This approach effectively addresses issues with adaptive optimizers, achieving strong performance on GLUE tasks and improving robustness to quantization. Despite these advantages, the method requires task-specific tuning and does not extend to generative models like GPT, limiting its applicability to broader use cases. Q-ViT [22] addressed the challenges of low-bit quantization for attention maps

in ViTs, where distortions in attention distances degrade model performance. By introducing entropy maximization and a distillation framework, the method preserves the accuracy of fully quantized ViTs. However, its focus on vision transformers raises questions about its applicability to language models, which have different attention mechanisms and performance metrics.

Edge-QAT [40] proposed an entropy-guided approach to manage outliers in query and key values during attention computations. By using token-aware quantization, the method achieved superior accuracy while preserving critical information in important tokens. However, the added complexity of adaptive precision may pose challenges for deterministic hardware implementation, particularly in real-time inference scenarios. In [7], the authors analyzed latency contributions of different layers in sub-8-bit quantization. They identified non-linear layers like Softmax as primary bottlenecks and proposed Power-of-Two operators and non-uniform dynamic quantization for efficient acceleration. While these strategies reduce computational overhead, they require custom hardware support, limiting generalizability across different platforms. The work in [28] extended QAT to generative models with a data-agnostic distillation framework, enabling the quantization of weights, activations, and Key-Value (KV) caches. This approach ensures compatibility with various models while maintaining performance, but the reliance on distillation adds significant computational overhead.

A key feature of QAT is its ability to adapt models to quantized representations during training, mitigating performance degradation commonly observed in PTQ. However, QAT methods often involve higher computational costs, as they simulate quantized arithmetic during both forward and backward passes, increasing resource requirements. Simulating quantized arithmetic during both forward and backward passes requires substantial resources, making these methods challenging to deploy for larger models or datasets. Further, the issues discussed in the previous section regarding the extent of quantization performed in such models remain the same here. Overall, while QAT methods offer significant advantages in preserving performance for quantized models, they are constrained by high computational demands and the difficulty of quantizing certain non-linear components.

C. Operator Replacement and Approximation

Operator replacement and approximation techniques have emerged as essential strategies for enabling integer-only inference in transformer architectures. By substituting computationally intensive non-linear operations with lightweight approximations, these methods aim to improve hardware compatibility while maintaining acceptable model performance. However, these techniques often come with inherent shortcomings, such as requiring custom hardware support or leaving certain operations in higher precision, which limits their generalizability.

I-BERT [16] pioneered integer-only inference for the BERT model by introducing lightweight integer approximations for non-linear operations such as GELU activation, Softmax, and LayerNorm. The approach utilized INT8 value storage with INT32 computations. While effective, I-BERT's reliance on INT32 for arithmetic operations and specific approximations highlights the need for custom kernel development to achieve fully hardware-efficient deployment. FQ-ViT [25] addressed the significant accuracy degradation observed when quantizing LayerNorm and Softmax in vision transformers. The study revealed high inter-channel variation in LayerNorm inputs and increased quantization errors from layer-wise approaches. To mitigate these issues, the authors implemented Power-Of-Two (PoT) quantization for LayerNorm inputs and developed logarithmic approximations for Softmax. Despite these advancements, FQ-ViT left GELU layers in FP32 precision, introducing inconsistencies in the quantization flow and requiring additional support for mixed precision inference.

Scale propagation techniques [26] constrained INT8 tensor scales across neural networks, replacing non-linear operations with INT8-compatible alternatives. For instance, polynomial attention utilized RELU functions instead of exponential operations, and approximate square root calculations were implemented via the L1 Batch Norm. While effective for specific layers, these approaches require precise calibration and often struggle with dynamic properties in activations. In [36], researchers expanded on this paradigm by replacing non-linear functions with approximate INT8-compatible operations. A binary search algorithm was employed to ensure quantization constraints, yet this technique often demanded significant computational overhead during calibration. I-ViT [23] introduced shift-based approximations for GELU and Softmax, termed Shift-GELU and Shift-Max, respectively. These methods replaced computationally expensive operations with bit-shifting approximations, improving performance in vision transformers. The Integer-LayerNorm technique used shift-based approximations for square root operations and incorporated quantization-aware fine-tuning to recover accuracy losses. However, the reliance on shift-based techniques and quantization-aware training implies limited applicability without custom hardware optimization.

Matrix multiplication optimizations explored in [17] employed float16/bfloat16 for multiplication while retaining FP32 for addition, reducing computational costs but deviating from pure integer inference. Similarly, [48] converted non-linear computations into hardware-friendly Lookup Table (LUT) operations, simplifying implementation at the cost of generalization across architectures. ShiftAddLLM [47] reparameterized multiplication operations into sequences of shifts and additions. This method quantized weight matrices into binary matrices with group-wise scaling factors, achieving improved accuracy with minimal performance loss. However, these methods often require custom hardware instructions to fully exploit their efficiency. SOLE [43] introduced log-based quantization for exponential functions in Softmax, replacing FP32 division with log-based approximations. This approach optimized LayerNorm computations but

necessitated log2 quantization support, limiting its hardware portability.

The diversity of operator replacement and approximation techniques underscores the complexity of quantizing non-linear layers in transformers. While these methods offer unique insights into balancing computational efficiency and model accuracy, challenges remain. Many techniques, such as log2 quantization, require custom hardware support, and others leave specific layers, like GELU or LayerNorm, in FP32 precision, creating bottlenecks for achieving fully hardware-friendly integer inference. Addressing these limitations through hardware-aware designs and adaptable quantization frameworks remains a critical avenue for future research.

III. Background

Transformers are the foundational architecture behind state-of-the-art Large Language Models (LLMs), revolutionizing natural language processing through their ability to model complex dependencies in sequential data. Introduced in the seminal work *Attention Is All You Need* [41], transformers address the limitations of recurrent and convolutional neural networks by employing a self-attention mechanism that processes input sequences in parallel.

Transformers originally featured an encoder-decoder-based structure. Encoder-only models like BERT excel at understanding tasks which has a bidirectional context when making predictions, while decoder-only architectures such as GPT are optimized for generative tasks which are autoregressive and causal in nature where the predictions only depend on the previous tokens. These specialized variants have been instrumental in advancing natural language understanding and generation across various domains. All the equation and content in this section is derived from the work of [41].
The transformer's key components are summarized below:

a) Tokenizer and Input Representation.: The tokenizer converts raw text into token indices, mapping each word or subword to an integer from a fixed vocabulary. Given a sequence of n tokens $[t_1, t_2, \ldots, t_n]$, these are embedded into dense vectors of dimension d_{model} via an embedding matrix $E \in \mathbb{R}^{V \times d_{\text{model}}}$, where V is the vocabulary size. The output is:

$$x_i = E[t_i], \quad i \in [1, n]. \tag{1}$$

Positional encodings added to embeddings, encode sequential information as:

$$\text{PE}(i,j) = \begin{cases} \sin\left(\frac{i}{10000^{j/d_{\text{model}}}}\right), & \text{if } j \text{ is even}, \\ \cos\left(\frac{i}{10000^{(j-1)/d_{\text{model}}}}\right), & \text{if } j \text{ is odd}. \end{cases} \tag{2}$$

where i is the position and j is the dimension.

b) Encoder Block.: Each encoder block consists of:

- *Multi-Head Self-Attention (MHSA):* Captures relationships between all tokens by computing scaled dot-product attention. For query (Q), key (K), and value (V) matrices:

$$\text{Attention}(Q,K,V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right)V, \tag{3}$$

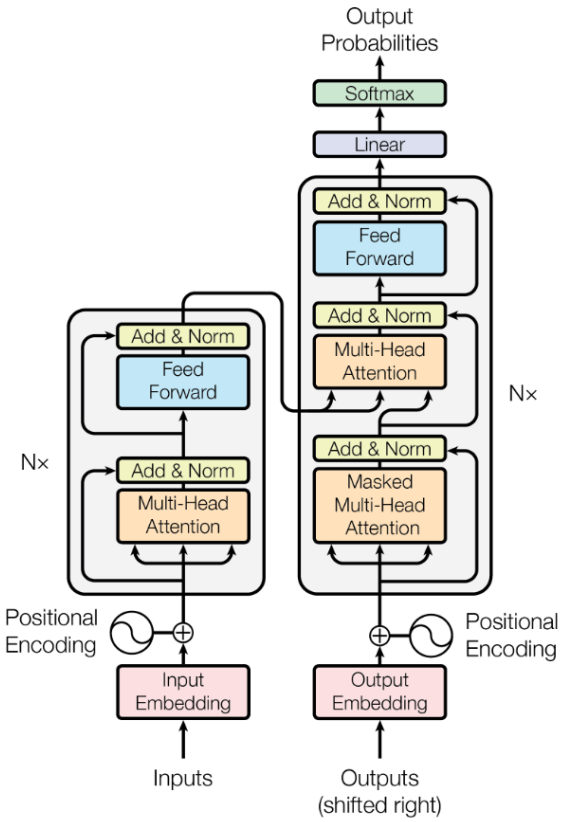

Fig. 1: Transformer Architecture [41]

where d_k is the key dimension. Multi-head attention enables the model to attend to different representation subspaces:

$$\text{MHSA}(X) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h)W^O, \tag{4}$$

with h attention heads, each computed as:

$$\text{head}_i = \text{Attention}(XW_i^Q, XW_i^K, XW_i^V). \tag{5}$$

- *Feed-Forward Neural Network (FFN):* Applies two linear transformations with a non-linear activation, typically GELU:

$$\text{FFN}(x) = \text{GELU}(xW_1 + b_1)W_2 + b_2, \tag{6}$$

where W_1, W_2, b_1, b_2 are learned parameters and GELU is defined as:

$$\text{GELU}(x) = x \cdot \Phi(x) = x \cdot \frac{1}{2}\left[1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right], \tag{7}$$

- *Layer Normalization and Residual Connections:* Stabilize training by normalizing inputs:

$$\text{LayerNorm}(x) = \frac{x - \mu}{\sqrt{\sigma^2 + \epsilon}} \cdot \gamma + \beta, \tag{8}$$

where μ and σ^2 are the mean and variance, and γ, β are learned parameters. Residual connections are added to aid gradient flow:

$$y = x + \text{Sublayer}(x). \tag{9}$$

c) Decoder Block.: The decoder block extends the encoder by incorporating:

- *Masked Multi-Head Self-Attention (M-MHSA):* Similar to MHSA, but masks future tokens to enforce causality during autoregressive generation. This is achieved by modifying the attention matrix with a mask M:

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^\top}{\sqrt{d_k}} + M\right)V. \tag{10}$$

- *Encoder-Decoder Attention:* Attends to encoder outputs, enabling the decoder to focus on relevant input tokens:

$$\text{Attention}(Q, K_{\text{enc}}, V_{\text{enc}}) = \text{softmax}\left(\frac{QK_{\text{enc}}^\top}{\sqrt{d_k}}\right)V_{\text{enc}}. \tag{11}$$

d) Final Linear and Softmax Layer.: The decoder outputs are projected into the vocabulary space using a linear transformation:

$$y = XW + b, \tag{12}$$

where $W \in \mathbb{R}^{d_{\text{model}} \times V}$ and $b \in \mathbb{R}^V$. Softmax converts the logits into probabilities:

$$P(y_i|x_{<i}) = \frac{\exp(y_i)}{\sum_j \exp(y_j)}. \tag{13}$$

Quantization of transformers involves representing the parameters and activations in lower-precision formats, such as INT8. While linear layers (e.g., matrix multiplications) are straightforward to quantize, non-linear operations like Softmax, GELU, and LayerNorm pose significant challenges due to their reliance on floating-point arithmetic. For instance:

- Softmax involves exponentials and division, requiring approximations like:

$$\exp(x) \approx 2^{-z} \cdot L(p), \quad \text{where } L(p) \text{ is a polynomial.} \tag{14}$$

- GELU approximations often require polynomials or piecewise functions to emulate the smooth activation curve.
- LayerNorm depends on mean and variance computations, which involve square roots and division:

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}. \tag{15}$$

These non-linear operations exhibit high dynamic ranges. Further making accurate representations of these operations in integer formats becomes challenging and prone to quantization-induced errors. This foundational understanding of transformer architecture and quantization challenges lays the groundwork for exploring hardware-friendly integer approximation techniques in subsequent sections.

IV. Approach

Neural network quantization addresses the computational challenges of large language models by reducing numerical precision from 32-bit floating-point (FP32) to 8-bit integer (INT8) representations. This technique aims to minimize memory consumption and accelerate inference while maintaining model performance. Our approach focuses on static quantization, which precomputes quantization parameters during a calibration phase, ensuring deterministic behaviour and hardware compatibility. A representative small causal language model is used in the context of this study. The quantization methodology for the model employs symmetric uniform quantization targeting INT8 precision. We implement channel-wise weight quantization and tensor-wise activation quantization, leveraging PyTorch's [8] autograd functionality. The scaling factor (S) and zero point (Z) are computed using min-max calibration:

$$S = \frac{\max - \min}{2^{n_{\text{bits}}}}, \quad Z = \text{round}\left(\frac{-\min}{S}\right), \tag{16}$$

where

$$n_{\text{bits}} = 8. \tag{17}$$

The conversion from a 32-bit floating-point value (x_{fp32}) to an 8-bit integer value (x_{int8}) is given by:

$$x_{\text{int8}} = \text{clip}\left(\text{round}\left(\frac{x_{\text{fp32}}}{S}\right) + Z, -128, 127\right), \tag{18}$$

where $\text{clip}(\cdot, -128, 127)$ ensures the value stays within the valid INT8 range $[-128, 127]$, and $\text{round}(\cdot)$ rounds to the nearest integer.

To enable integer arithmetic while preserving the FP32 model representation, we implemented a Quantization-Dequantization (QDQ) mechanism. This approach involves quantizing inputs to INT8 for computational processing, performing low-precision computations, and subsequently dequantizing to FP32. We strategically inserted QDQ blocks in critical model components, including embedding layers, query-key-value projections, feed-forward networks, and normalization layers. It is important to note that while the model parameters are technically still in FP32, the computations occur in Integer precision these QDQ blocks will later be replaced with INT8 kernels when deploying on the hardware. However, for the context of this work, the evaluation provides insight into the performance of such a model when devised. While plenty of research focuses on the language model quantization only the works of I-BERT, I-VIT and FQ-VIT actually manage to quantize the non-linear layers within the transformer architecture, in essence, able to obtain a full INT inference model. Further, their respective works validate the performance on the required benchmark tasks and verify that the usage of these approximations in the respective task domain is able to maintain the performance (accuracy) with their respective FP32 counterparts. Therefore this research adopts approximation techniques

from I-BERT, FQ-VIT and I-VIT for critical neural network operations. Specifically, we implemented polynomial expansions for Softmax operations and iterative techniques like Newton-Raphson for normalization layer computations which are used in the original works. These approximations were carefully selected to maintain numerical stability while exploring the performance boundaries of quantization exactly as utilized in the original works. Our experimental framework evaluated the quantization approach across diverse NLP benchmarks. We compared Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT) to assess the trade-offs between model accuracy, perplexity, sensitivity metrics and computational efficiency which are discussed in the upcoming sections. The analysis focuses on the performance of the model undergoing this simulated quantization to mimic 8-bit integer representations. The upcoming sections provide detailed reasoning for the parameters selected for performing the experiments/analysis. A detailed list of tasks and sensitivity nomenclature can be found in the appendix.

A. Selected Tasks

The proposed evaluation framework encompasses a comprehensive set of linguistic, reasoning, and domain-specific benchmarks designed to assess the performance capabilities of quantized machine learning models systematically. Leveraging the evaluation harness framework developed by Eleuther AI [12], the benchmarking framework stratifies tasks across three primary analytical domains: question-answering and reasoning, language understanding and generation, and physical and scientific knowledge assessment. The selected tasks evaluate distinct capabilities: Question-answering and reasoning tasks (ARC, BoolQ, Hellaswag) assess multi-step logical inference and commonsense understanding. Language understanding tasks (LAMBADA, Winogrande) evaluate contextual comprehension and predictive modelling. Scientific and physical reasoning tasks (OpenBookQA, SciQ, PIQA) measure domain-specific knowledge and real-world physical understanding.

The primary evaluation metrics across tasks are accuracy [9] and perplexity [10], defined as:

$$\text{Accuracy} = \frac{\text{number of correct predictions}}{\text{total number of samples}} \tag{19}$$

$$\text{Perplexity} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log p(w_i|w_{1:i-1})\right) \tag{20}$$

where N represents the total number of words and $P(w_i)$ the predicted probability of the i-th word.

As shown in Table I, these tasks are categorized by their output format: multiple-choice questions (MCQ), binary-choice questions (BCQ), and cloze-style completion tasks. The output space varies from binary choices to vocabulary-sized spaces for open-ended tasks. Detailed descriptions of individual tasks and their specific evaluation protocols can be found in the Appendix.

Format	Tasks	Output Space
MCQ	ARC, Hellaswag, LogiQA, OpenBookQA, SciQ	$\|\mathcal{O}\| = 4$
BCQ	BoolQ, Winogrande, PIQA	$\|\mathcal{O}\| = 2$
Cloze	LAMBADA, Wikitext	$\|\mathcal{O}\| = \|V\|$

TABLE I: Task Classification by Output Format

B. Approximations for Full-INT8 Quantization

In Table II, we summarize the integer-only approximations for Softmax and LayerNorm across three methods that build upon the standard floating-point equations. I-BERT uses a polynomial approximation to replace floating-point exponentials and an iterative scheme for computing variance in integer form, making exponentiation and division more hardware-friendly without drastically reducing accuracy. I-ViT employs bit-shifting operations, referred to as ShiftMax, to replace exponential functions, and it also computes variance with iterative integer updates; this approach simplifies multiplication and division through shift-based arithmetic. FQ-ViT extends these ideas by introducing power-of-two constraints in the LayerNorm calculation, which allows for more efficient scaling, and by implementing a logarithm-based integer Softmax (Log-Int-Softmax) that handles large exponents through integer log operations. While each approach focuses on fully eliminating floating-point overhead, they balance the trade-off between computational efficiency and accuracy in slightly different ways, often depending on the target hardware platform's capabilities. The complete set of equations for the approximations can be found in the Appendix.

V. Experimental Setup

A. Selected Model

This research investigates quantization techniques for the OPT-125M model from Meta [50]. With 125 million parameters, the model is suitable for deployment on embedded devices with limited computational resources. While extensive research exists on quantizing larger models like GPT-2, fewer studies have explored sub-billion parameter models for edge computing environments. The study addresses this research gap by focusing on models with practical edge deployment potential. The OPT-125M model represents this underexplored category, offering competitive performance on NLP benchmark tasks despite its reduced size. Its recognition on the Hugging Face leaderboard [34] as a previous SOTA. However, now replaced by the SMOL-LM [14] highlights the model's relevance in evaluating the trade-offs between model accuracy and computational efficiency under quantization. This area of research focussing on optimizing smaller models has been the recent trend in the past few months and this work particularly focuses on optimizing the model for space and time complexity by utilizing hardware-friendly approximations without modifying the model architecture. This essentially brings us closer to deploying such language models on microcontrollers which are flexible for AI use cases with slightly relaxed memory constraints. In essence, such a model on FP32 precision would essentially

Method	Softmax Approx.	LayerNorm Approx.
I-BERT	$\exp(x) \approx 2^{-z} L(p)$ $L(p) = 0.3585(p + 1.353)^2 + 0.344$	Iterative variance: $I_{i+1} = \frac{I_i + \left\lfloor \frac{\mathrm{Var}(I_x)}{I_i} \right\rfloor}{2}$
I-ViT	*ShiftMax*: $I_{\text{out}} = \mathrm{IntDiv}\big(I_{\exp}, \sum_j I_{\exp_j}, k_{\text{out}}\big)$	Same iterative LN as I-BERT
FQ-ViT	*Log-Int-Softmax*: $\text{i-exp}(\tilde{X}) = L(p) \gg z$	*Power-of-Two* LN: $\hat{X}_Q = (X_Q - z_p) \ll \alpha$ $Y_Q = \lfloor A\hat{X}_Q + B \rfloor + z_{p_{\text{out}}}$

TABLE II: Integer-Only Approximations for Softmax and LayerNorm (Key Equations)

require 500MB on-device storage. However, with the utilization of such model approximations, we are able to limit it to 125MB which would provide a practical 4x reduction on memory cost and also have all the computations on an 8-bit integer providing us the benefit of accelerations on NPU which could potentially also yield higher speedups minimizing latency costs. Further details about the model architecture can be found in Table V and the original work [50]

B. Experiment 1: PTQ vs QAT on Pile-10K

To address the first research question, we analyzed various approximation methods from the literature, implementing them within their original quantization frameworks under open-source licenses. These approximations were integrated with our target model to evaluate their effectiveness in both post-training quantization (PTQ) and quantization-aware training (QAT) scenarios. For I-BERT and I-VIT specifically, we conducted QAT evaluations to align with their original implementation approaches, ensuring a fair comparison. Here, it is important to note that FQ-VIT is originally a PTQ work and therefore QAT experiments on thisweres not performed as a part of this work. Our experimental methodology consisted of several phases. First, we performed a thorough comparison of all techniques using PTQ, followed by a focused comparison of I-BERT and I-VIT approximations in QAT which are in line with their original designs. We used the Pile-10K [32] as our initial calibration dataset, which was consistent with common practice in model optimization frameworks, and standardized the calibration step size to 512 across all experiments.

C. Experiment 2: Dataset Impact on QAT Performance

To address the second research question regarding performance recovery, we conducted QAT experiments using three distinct datasets: a downsampled version of C4[1], Pile-Val [19], and Pile-10K [32]. More details about the dataset and the samples used for training and validation can be found in the Appendix. Training continued until optimal convergence, with careful consideration given to overfitting on smaller datasets. We implemented early stopping with a three-epoch threshold, used a learning rate of 1e-6, and employed the AdamW optimizer with a linear scheduler.

D. Experiment 3: Impact of different Observers on PTQ in FQ-VIT

Building upon FQ-VIT's methodology, we investigated four distinct observers for Post-Training Quantization (PTQ) calibration: MinMax, which determines quantization scale by capturing the absolute minimum and maximum values in the tensor distribution; OMSE (Optimized Mean Squared Error), which iteratively adjusts quantization parameters by performing a binary search over possible clipping values to find the configuration that yields the smallest L2 distance between original and quantized tensors; EMA (Exponential Moving Average), which updates quantization parameters using a weighted average of historical and current statistics ($Q\_params\_t = \alpha * Q\_params\_(t-1) + (1-\alpha) * current\_stats$, where α is typically set between 0.9 and 0.99) to reduce the impact of temporary fluctuations; and Percentile-based, which sets quantization bounds by excluding values outside specified percentiles (typically 0.1% and 99.9%) to prevent extreme outliers from expanding the quantization range unnecessarily. These observers were implemented and evaluated specifically for quantizing weights and activations in causal autoregressive language models during the calibration phase.

E. Experiment 4: Quantization Sensitivity Analysis

This work was further substantiated by conducting ablation studies through selectively disabling quantization for specific modules and analyzing their individual contributions to overall task performance. The ablation study examined five distinct model configurations: the baseline FP32 model and four quantized variants. The first quantized variant is one without the approximations for the non-linear layers while maintaining the remaining compute flow in Integer precision. The second variant is one with all the approximations for the non-linear layers introduced. The third one makes use of the first quantized variant as backbone with the inclusion of the SoftMax approximation and the final variant like the previous also uses the first variant as backbone with layernorm approximation included. This is exactly aligned

with the original approximation configuration utilized in table V where the previous five configurations (or non-linear modules) are selectively disabled for obtaining results for the impact of approximation on downstream tasks. This analysis provided valuable insights into our research questions specifically the main RQ and sub RQ2 while maintaining methodological consistency with previous work and providing solid reasoning to understand the impact of static quantization on the causal model. Additionally, we developed a quantization sensitivity framework to identify potential bottlenecks and gain deeper insights into model behaviour under quantization. This framework involved hooking activation tensors at various model layers and comparing them with their quantized counterparts. We employed multiple metrics from the literature to assess the noise introduction patterns across different model components during the quantization process, and the approximations were incorporated to answer our primary research question.

VI. Results

The detailed architecture overview of the model architecture incorporating the techniques from the approximations and overall obtaining a statically quantized 8-bit integer* representation for the OPT-125M is presented in V. For better comparison an equivalent statically quantized 8-bit model whose quantization configuration as performed in the SOTA model optimization toolkit [15] is listed in the table. This section presents the analysis of task performance for quantized models from the previously discussed configuration, focusing on accuracy and perplexity metrics across selected NLP tasks. The evaluation is conducted in two stages. The first analyzes the performance of Post-Training Quantization (PTQ) versus Quantization-Aware Training (QAT) using I-BERT and I-VIT approximations on the Pile-10K dataset. The second examines how dataset selection impacts task performance, comparing I-VIT QAT results across Pile-10K [32], C4 [1], and Pile-Val [19] datasets. Please note that the results in this section have their relative drop/increase in comparison with the FP32 baselines indicated with $\uparrow$ / $\downarrow$ in $\%$ format.

A. *PTQ vs QAT on Pile-10K (I-BERT and I-VIT)*

The first set of experiments evaluates the impact of PTQ and QAT techniques for the I-BERT and I-VIT approximations on the Pile-10K dataset. Accuracy and perplexity metrics were measured across eight benchmark NLP tasks and two variants of the LAMBADA dataset. From the accuracy results in table III, QAT consistently improves performance compared to PTQ across most tasks. This improvement is particularly evident for I-VIT, where QAT recovers performance degradation observed in PTQ by fine-tuning the model to account for quantized representations. Tasks such as **BoolQ** and **HellaSwag** exhibit significant gains under QAT, indicating the ability of the fine-tuned model to handle task-specific distributions. For I-BERT, the performance improvement with QAT is smaller but steady, reflecting its inability to maintain task performance in PTQ configurations due to its effective approximations.

In terms of perplexity from the table IV, QAT demonstrates clear advantages over PTQ. I-VIT QAT achieves the lowest perplexity across all text generative tasks evaluated as a part of this study overperforming the other approximations significantly and also yielding results better than its PTQ counterparts indicating its effectiveness in generative tasks. PTQ, while competitive for simpler tasks like **ARC-Easy** and **Winogrande**, shows clear limitations in more complex tasks that involve long-range dependencies and reasoning.

These results highlight the **limitations of PTQ for complex tasks and the effectiveness of QAT** in addressing these challenges by adapting the model to quantized representations during training.

B. *Dataset Impact on QAT Performance*

The second set of experiments examines how dataset selection affects the QAT performance of both I-VIT and I-BERT approximations across Pile-10K, C4, and Pile-Val datasets. The study evaluated accuracy and perplexity metrics for benchmark NLP tasks, revealing distinct patterns between the approximation methods. As shown in the accuracy comparison table VI, dataset selection proves to be a crucial factor for both approximation methods, though their responses differ. I-VIT demonstrates superior performance with the C4 dataset, particularly excelling in complex reasoning tasks like HellaSwag and SciQ. While I-BERT follows a similar trend, it consistently achieves lower scores on these same tasks when trained on C4. Interestingly, the Pile-Val dataset emerges as the stronger performer for both methods on tasks requiring precise linguistic understanding, such as BoolQ and Winogrande. This suggests that the better prepared and concise version of the original Pile dataset i.e. Pile Val provides better representations for the model to adapt towards yielding significantly better results. The Pile-10K dataset shows the weakest performance across both approximation methods, with I-BERT exhibiting greater sensitivity to the dataset's limitations. This performance gap is most evident in tasks like LogiQA, where I-BERT shows notably lower accuracy compared to I-VIT, highlighting I-BERT's greater vulnerability to constraints in dataset size and diversity.

For perplexity from table VII, the results reveal interesting distinctions between the approximation methods. In I-VIT, C4 achieves the lowest perplexity for LAMBADA-Standard task in comparison with I-BERT on the same task on the c4 dataset. This demonstrates poor generalizability constraint from I-BERT in comparison to the I-VIT approximation. However, both methods show that Pile-Val achieves better perplexity on WikiText compared to Pile-10K, with I-VIT reaching 204.75 and I-BERT achieving 545.70, suggesting that this processed dataset offers benefits to both approximation methods across traditional language modelling tasks.

The comparative analysis reveals several key insights:

Task	Original	Static Quant (MP)	I-VIT (PTQ)	I-VIT (QAT)	I-BERT (PTQ)	I-BERT (QAT)	FQ-VIT (PTQ)
ARC-Challenge	22.78	23.29 (↑ +2.25%)	24.40 (↑ +7.11%)	23.21 (↑ +1.87%)	23.89 (↑ +4.87%)	21.59 (↓ -5.23%)	22.46 (↓ -1.40%)
ARC-Easy	39.98	39.52 (↓ -1.15%)	26.56 (↓ -33.56%)	31.52 (↓ -21.16%)	29.42 (↓ -26.41%)	28.70 (↓ -28.22%)	26.12 (↓ -34.67%)
BoolQ	55.44	54.34 (↓ -1.98%)	41.04 (↓ -25.96%)	55.20 (↓ -0.43%)	41.71 (↓ -24.77%)	37.83 (↓ -31.76%)	41.89 (↓ -24.44%)
HellaSwag	31.35	31.28 (↓ -0.22%)	26.23 (↓ -16.35%)	27.46 (↓ -12.39%)	27.50 (↓ -12.28%)	25.23 (↓ -19.52%)	27.98 (↓ -10.75%)
LogiQA	27.65	27.80 (↑ +0.54%)	24.58 (↓ -11.11%)	20.43 (↓ -26.13%)	24.42 (↓ -11.68%)	21.80 (↓ -21.16%)	20.12 (↓ -27.23%)
OpenBookQA	28.00	27.60 (↓ -1.43%)	26.20 (↓ -6.43%)	24.00 (↓ -14.29%)	28.80 (↑ +2.86%)	26.60 (↓ -5.00%)	29.45 (↑ +5.18%)
PIQA	61.97	62.13 (↑ +0.26%)	48.86 (↓ -21.15%)	54.41 (↓ -12.20%)	52.72 (↓ -14.92%)	51.90 (↓ -16.25%)	52.89 (↓ -14.65%)
SciQ	67.00	66.70 (↓ -0.45%)	27.90 (↓ -58.36%)	46.70 (↓ -30.30%)	28.40 (↓ -57.61%)	24.90 (↓ -62.84%)	27.89 (↓ -58.37%)
Winogrande	50.36	50.59 (↑ +0.46%)	51.70 (↑ +2.66%)	49.41 (↓ -1.89%)	49.64 (↓ -1.43%)	50.43 (↑ +0.14%)	49.45 (↓ -1.81%)
Lambada (OpenAI)	37.84	37.12 (↓ -1.91%)	0.60 (↓ -98.41%)	10.21 (↓ -73.02%)	0.12 (↓ -99.68%)	0.00 (↓ -100.00%)	0.10 (↓ -99.74%)

TABLE III: Accuracy Comparison of the OPT-125M variants across NLP benchmark tasks

Task	Original	Static Quant (MP)	I-VIT (PTQ)	I-VIT (QAT)	I-BERT (PTQ)	I-BERT (QAT)	FQ-VIT (PTQ)
WikiText	31.90	32.82	6,599.97	569.65	80,955.83	1,338.35	7,856.00
Lambada-Standard	73.09	81.26	6,235,784.24	3,350.63	8,236,179.00	898,466.83	124,089.00
Lambada-OpenAI	26.02	27.12	1,036,283.65	1,075.38	629,777.24	92,591.04	4,636,890.00

TABLE IV: Perplexity Comparison of the OPT-125M variants across NLP benchmark tasks

Module	Layer	PyTorch Representation	Dimensions	Parameters
Embedding	Input Tokens	`torch.Tensor`	2048×768	-
	Embedding Table	`nn.Embedding`	50272×768	38,567,936
Self-Attention (12 Blocks)	Query, Key, Value Projections	`nn.Linear`	$768 \rightarrow 768$	21,261,312
	Query-Key Matrix Multiply	`torch.bmm`	$(2048, 768) \cdot (768, 2048)$	-
	Softmax	`nn.Softmax`	$(2048, 2048)$	-
	Value Weighted Multiply	`torch.bmm`	$(2048, 2048) \cdot (2048, 768)$	-
	Output Projection	`nn.Linear`	$768 \rightarrow 768$	7,087,104
Attention Norm	LayerNorm (Attention)	`LayerNorm`	768	18,432
Feed-Forward Network (FFN, 12 Blocks)	FC1	`nn.Linear`	$768 \rightarrow 3072$	28,311,552
	ReLU	`nn.ReLU`	-	-
	FC2	`nn.Linear`	$3072 \rightarrow 768$	28,311,552
FFN Norm	LayerNorm (FFN)	`LayerNorm`	768	18,432
Output Layers	Final LayerNorm	`LayerNorm`	768	1,536
	LM Head	`nn.Linear`	$768 \rightarrow 50272$	38,567,936

Module	Layer	Static (MP) Conf	Approx Conf
Embedding	Input Tokens	**FP32**	FP32
	Embedding Table	**FP32**	INT8*
Self-Attention	Query, Key, Value Projections	INT8	INT8*
	Query-Key Matrix Multiply	**FP32**	INT8*
	Softmax	**FP32**	INT8*
	Value Weighted Multiply	**FP32**	INT8*
	Output Projection	INT8	INT8*
	Layer Norm	**FP32**	INT8*
Feed-Forward Network (FFN)	FC1	INT8	INT8*
	ReLU	INT8	INT8*
	FC2	INT8	INT8*
Output Layers	Final LayerNorm	**FP32**	INT8*
	LM Head	INT8	FP32

TABLE V: Detailed Architecture of the OPT-125M Model with Static (MP) and Approximation Configurations (I-VIT, I-BERT, FQ-VIT)

1) Both approximation methods are highly sensitive to dataset selection, but I-BERT shows greater performance degradation with suboptimal datasets.
2) I-VIT demonstrates more stable performance across different datasets, maintaining higher accuracy and lower perplexity consistently. While I-BERT shows stronger dependency on dataset quality, with performance gaps widening on smaller or less diverse datasets
3) Complex reasoning tasks benefit more from C4's diversity in both methods. Linguistic understanding tasks show better results with cleaner Pile-Val's representations. Further, both methods struggle with limited data in Pile-10K, but I-BERT's performance drops more significantly

These results demonstrate that **dataset characteristics play a critical role in determining the effectiveness of quantized models, with the impact being more pronounced for I-BERT approximation.** While C4 provides a balanced foundation for both methods, Pile-Val offers advantages since it is a much more processed/cleaned and compact version of the original Pile dataset. The comparison reveals that I-VIT generally maintains better performance across datasets, while I-BERT requires more careful dataset selection to achieve optimal results.

Overall, these findings suggest that while both approximation methods can achieve reasonable performance, I-VIT shows more resilience to dataset variations. The choice between methods should consider both the target task requirements and available training data, with I-VIT being more suitable for scenarios with dataset constraints and I-BERT potentially offering competitive performance when

Task	Original (FP32)	I-VIT Approximation			I-BERT Approximation		
		Pile-10K	Pile-Val	C4	Pile-10K	Pile-Val	C4
ARC-Challenge	22.78	23.21 ↑ (+1.87%)	21.76 ↓ (-4.48%)	22.10 ↓ (-3.00%)	21.59 ↓ (-2.45%)	21.12 ↓ (-7.29%)	21.54 ↓ (-5.44%)
ARC-Easy	39.98	31.52 ↓ (-21.16%)	33.08 ↓ (-17.26%)	34.43 ↓ (-13.88%)	28.70 ↓ (-28.34%)	31.89 ↓ (-20.24%)	33.12 ↓ (-17.16%)
BoolQ	55.44	55.20 ↓ (-0.43%)	62.20 ↑ (+12.19%)	59.36 ↑ (+7.07%)	41.71 ↓ (-23.59%)	49.15 ↑ (-9.50%)	52.23 ↑ (-3.23%)
HellaSwag	31.35	27.46 ↓ (-12.39%)	27.46 ↓ (-12.39%)	27.52 ↓ (-12.20%)	25.23 ↓ (-16.68%)	26.34 ↓ (-16.00%)	26.45 ↓ (-15.63%)
LogiQA	27.65	20.43 ↓ (-26.13%)	27.65 ≈ (0.00%)	24.73 ↓ (-10.57%)	21.89 ↓ (-22.38%)	26.12 ↓ (-5.53%)	23.45 ↓ (-5.19%)
OpenBookQA	28.00	24.00 ↓ (-14.29%)	24.80 ↓ (-11.43%)	26.60 ↓ (-5.00%)	26.60 ↓ (-5.00%)	25.65 ↓ (-5.54%)	25.72 ↓ (-6.29%)
PIQA	61.97	54.41 ↓ (-12.20%)	56.58 ↓ (-8.70%)	58.32 ↓ (-5.89%)	51.90 ↓ (-16.25%)	54.23 ↓ (-12.49%)	56.18 ↓ (-9.34%)
SciQ	67.00	46.70 ↓ (-30.30%)	48.90 ↓ (-27.01%)	53.50 ↓ (-20.15%)	24.90 ↓ (-62.84%)	36.32 ↓ (-40.87%)	38.85 ↓ (-44.10%)
Winogrande	50.36	49.41 ↓ (-1.89%)	51.30 ↑ (+1.87%)	48.54 ↓ (-3.62%)	50.43 ↓ (+0.14%)	49.65 ↓ (-1.41%)	47.12 ↓ (-6.43%)
Lambada (OpenAI)	37.84	10.21 ↓ (-73.01%)	18.44 ↓ (-51.28%)	18.40 ↓ (-51.39%)	0.10 ↓ (-98.82%)	15.85 ↓ (-64.47%)	16.92 ↓ (-60.28%)

TABLE VI: Accuracy Comparison of Different Quantization Methods Across Datasets

Task	Original (FP32)	I-VIT Approximation			I-BERT Approximation		
		Pile-10K	Pile-Val	C4	Pile-10K	Pile-Val	C4
WikiText	32	569 ↑ (+1686%)	204 ↑ (+541%)	214 ↑ (+572%)	1338↑ (+4097%)	545 ↑ (+1610%)	657 ↑ (+1961%)
Lambada-Standard	73	3350 ↑ (+4485%)	1384 ↑ (+1794%)	408 ↑ (+458%)	898466 ↑ (+1229047%)	45540 ↑ (+62208%)	48767 ↑ (+66622%)
Lambada-OpenAI	26	1075 ↑ (+4032%)	315 ↑ (+1112%)	1291 ↑ (+4863%)	92591 ↑ (+355846%)	48571 ↑ (+186569%)	50267 ↑ (+193089%)

TABLE VII: Perplexity Comparison of Different Quantization Methods Across Datasets

Task	MinMax	OMSE	Percentile	EMA
Accuracy (%)				
ARC-Challenge	22.46	22.89	23.01	22.78
ARC-Easy	26.12	26.45	26.89	26.34
BoolQ	41.89	42.05	42.33	41.96
HellaSwag	27.98	28.12	28.45	28.21
LogiQA	20.12	20.34	20.68	20.45
OpenBookQA	29.45	29.67	30.01	29.78
PIQA	52.89	53.01	53.45	53.12
SciQ	27.89	28.05	28.34	28.12
Winogrande	49.45	49.67	50.12	49.78
Lambada (OpenAI)	0.10	0.12	0.11	0.11
Perplexity				
Wikitext	7856	8923	7890	9878
Lambada Standard	124 089	134 378	144 198	154 167
Lambada OpenAI	4 636 890	4 737 589	4 837 125	5 037 089

TABLE VIII: Quantization Performance Across Different Observer Types

paired with high-quality, task-aligned datasets.

C. Impact of Different Observers on PTQ in FQ-VIT

The choice of observers in post-training quantization significantly influences model performance across various tasks. MinMax observers, while providing stable quantization ranges, demonstrate conservative performance that leads to suboptimal accuracy across most tasks as evident in VIII. The OMSE observer offers a more balanced approach, showing modest improvements over MinMax by optimizing the quantization range through error minimization. The percentile-based observer emerges as the most effective, particularly in tasks requiring nuanced understanding such as BoolQ and Winogrande, attributed to its ability to exclude outliers while maintaining resolution for relevant activation ranges. The EMA observer, with its adaptive nature, provides a middle-ground approach between MinMax and OMSE, though it shows limitations in capturing the full complexity of language modelling tasks as evidenced by higher perplexity metrics. **Notably, all observers exhibit more significant degradation in language modelling capabilities compared to reasoning and comprehension tasks, suggesting that current PTQ approaches may require task-specific observer selection or additional optimization techniques for language modelling scenarios**.

D. Quantization Sensitivity Analysis

In this section, the activation tensors of each layer within the model architecture are used for comparison. The quantized tensor in the model with approximations is evaluated against its FP32 baselines from the original model to determine quantization sensitivity, The assessment of quantization effects on transformers requires a set of metrics to evaluate the quality and fidelity of the quantized representations compared to their full-precision counterparts. This section presents an in-depth analysis of key metrics used to measure quantization sensitivity the set of which are detailed in the Appendix. Each metric provides unique insights into quantization effects. For instance, SQNR offers a comprehensive view of signal preservation, while HAWQ

provides a theoretically grounded sensitivity assessment. MSE provides direct error quantification, and kurtosis helps identify potentially problematic distributions. KL divergence and cosine similarity provide complementary perspectives in quantization analysis: KL divergence measures how well the overall statistical distribution of values is preserved between the original and quantized tensors, while cosine similarity measures the degree of linear correlation between the tensors. Therefore, combining these metrics provides a suitable framework for analyzing quantization sensitivity, enabling informed decisions about precision requirements across different layers and operations in large language models.

1) Analysis of Sensitivity Metrics Across Depth: While it is possible to inspect different layers across the language model architecture. One of the approximated layers in the model within the attention blocks specifically the Attention Layernorm is inspected with the sensitivity tool where the metrics are evaluated against the counterpart tensor from the original model in FP32 precision. The visualization can be found in Figure 2 Further, layers are inspected whose visualizations can be found in the Appendix.

The sensitivity analysis of the self-attention layer normalization reveals complex interactions between different metrics across network depth, providing insights into quantization effects and architectural implications. The analysis encompasses multiple complementary metrics: Signal-to-Quantization-Noise Ratio (SQNR), Mean Squared Error (MSE), Kullback-Leibler (KL) divergence, cosine distance, HAWQ sensitivity score, and kurtosis. The SQNR exhibits a distinctive pattern, peaking at approximately 27 dB in layer 0 before experiencing a sharp decline. This behaviour suggests optimal signal fidelity in the initial layer, followed by increased quantization noise in subsequent layers. This finding is particularly significant when considered alongside the MSE, which shows a general upward trend, especially pronounced from layer 4 onwards, culminating at layer 11. The inverse relationship between SQNR and MSE in layers 6-11 provides strong evidence of quantization noise directly impacting representation accuracy. The layer-wise analysis reveals several critical regions. Layer 1 demonstrates a notable spike in HAWQ sensitivity while maintaining relatively stable performance in other metrics, indicating a unique sensitivity to weight perturbations without compromising overall quantization properties. The middle layers (4-8) exhibit increased volatility across multiple metrics, suggesting this region plays a crucial role in information transformation within the network. Distributional characteristics, as measured by KL divergence, which is technically a limitation of the plot as the range of the values is around e-2 whose original values are relatively aligned with the steadily increasing cosine distance (reaching approximately 0.6 by layer 11). This divergence between metrics indicates that while the quantization process preserves overall statistical distributions, it progressively affects the geometric relationships between quantized and non-quantized representations. The kurtosis metric's decreasing trend further supports this observation, suggesting a gradual dispersion of activation distributions as network depth increases. These findings have several important implications for quantization strategy design:

1) The high initial SQNR followed by a sharp drop suggests the potential benefit of higher precision quantization in the first layer to preserve input feature fidelity.
2) The steady increase in both MSE and cosine distance indicates cumulative quantization effects, which might necessitate adaptive precision requirements for deeper layers.
3) The periodic spikes in HAWQ sensitivity, particularly notable at layers 1 and 4, could align with architectural boundaries, suggesting potential locations for optimization focus.

While the focus of this work is to achieve full integer quantization and inference through this experiment we see patterns that demonstrate that the use of the approximations contributes significantly to the error propagation. These insights suggest a fine-grained approach to quantization design, where layer-specific characteristics inform precision requirements and optimization strategies. The analysis particularly highlights the importance of carefully considering the initial and deep layers, where quantization effects appear most pronounced. Future work might explore adaptive quantization schemes that account for these layer-wise sensitivities while maintaining model efficiency in order to maintain the performance of the model with the approximations, however that would contradict the benefits obtained with full integer quantization and inference making it unfriendly for edge deployment.

From the figures 3 and 4 it becomes more evident that the MSE across approximations clearly seem to increase across depth for the attention layer norm while the SQNR seems to decrease which points out to a common conclusion which is that as depth with the layer increases the errors start compounding which increases the noise introduced by the approximations during quantization. Further, the layers in the initial stages of the model seem the least error-prone compared to those found in the later stages/higher depths. Moreover from both the figures the **I-VIT approximation seems to be the most effective overall having a higher SQNR value and lower MSE in comparison with the I-BERT and FQ-VIT approximations**. The results found in tables III clearly point to the same conclusion. With the findings from this experiment, the results are supplemented with further evidence that the approximations indeed yield/contribute to better results in comparison with the other approximations evaluated as a part of this study.

E. Ablation Study

The ablation study examined five distinct model configurations: the baseline FP32 model and four quantized variants. The results for these can be identified in Figure 5. It is important to note that for easier visualization the tasks have been grouped into four different categories namely Language

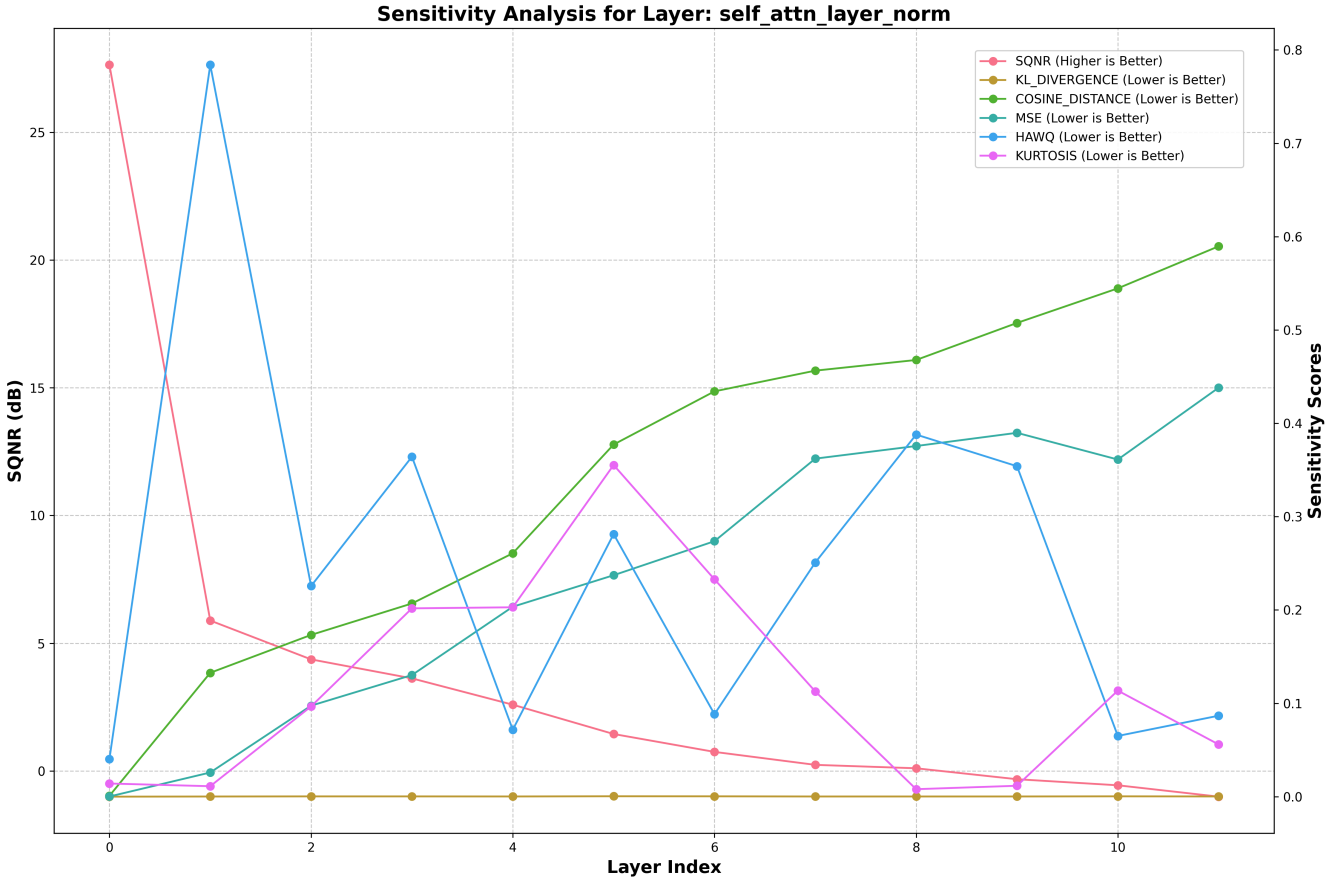

Fig. 2: Quantization sensitivity on the Attention LayerNorm in OPT-125M

Modelling (LAMBADA), Commonsense Reasoning (BoolQ, Winogrande), Logical/Practical (LogiQA, PiQA, Hellaswag) and Science/Knowledge (ARC, OpenbookQA, SciQ). Here the average accuracy across the grouped categories is used for visualization for easier understanding from Figure 5. The performance patterns across different cognitive tasks reveal several notable insights. The CommonSense Reasoning task demonstrates the highest baseline performance at 53% for FP32, with all quantized variants maintaining relatively consistent performance around 44%, suggesting that common sense reasoning capabilities remain stable under various quantization schemes. A particularly interesting observation emerges in the Science/Knowledge domain, where the quantized version without approximation (29%) outperforms the variant with approximation (25%), while both SoftMax and layernorm configurations maintain comparable performance (approximately 29%). This unexpected behaviour suggests that simpler quantization approaches might be sufficient for knowledge-based tasks. The Logical/Practical reasoning category shows a more uniform degradation pattern, with the FP32 baseline at 40% and quantized variants clustering around 35%, indicating that logical reasoning capabilities degrade gracefully under quantization. The most striking contrast appears in Language Modeling, where performance drops dramatically from 38% in FP32 to approximately 10% across quantized variants, with minimal variation among different quantization configurations. This suggests that language modelling tasks are particularly sensitive to precision reduction, regardless of the specific quantization approach employed. **These findings indicate that while quantization inevitably impacts model performance, the extent of degradation is highly task-dependent, with CommonSense Reasoning showing the most resilience and Language Modeling requiring careful consideration before applying any form of quantization.**

VII. Discussion

Our investigation into quantization effects on non-linear layers in causal language models reveals significant methodological and implementation challenges. The primary constraints emerge from the limitations inherent in model representation, with our current study's reliance on a single model architecture, OPT-125M. While this provides conclusive insights into a specific model architecture there are different models in existence in different parameter counts. While post-processing quantization to 8-bit integer precision is straightforward, maintaining this precision throughout the inference pipeline presents significant challenges in language models. This creates an opportunity to develop novel approximation techniques specifically optimized for

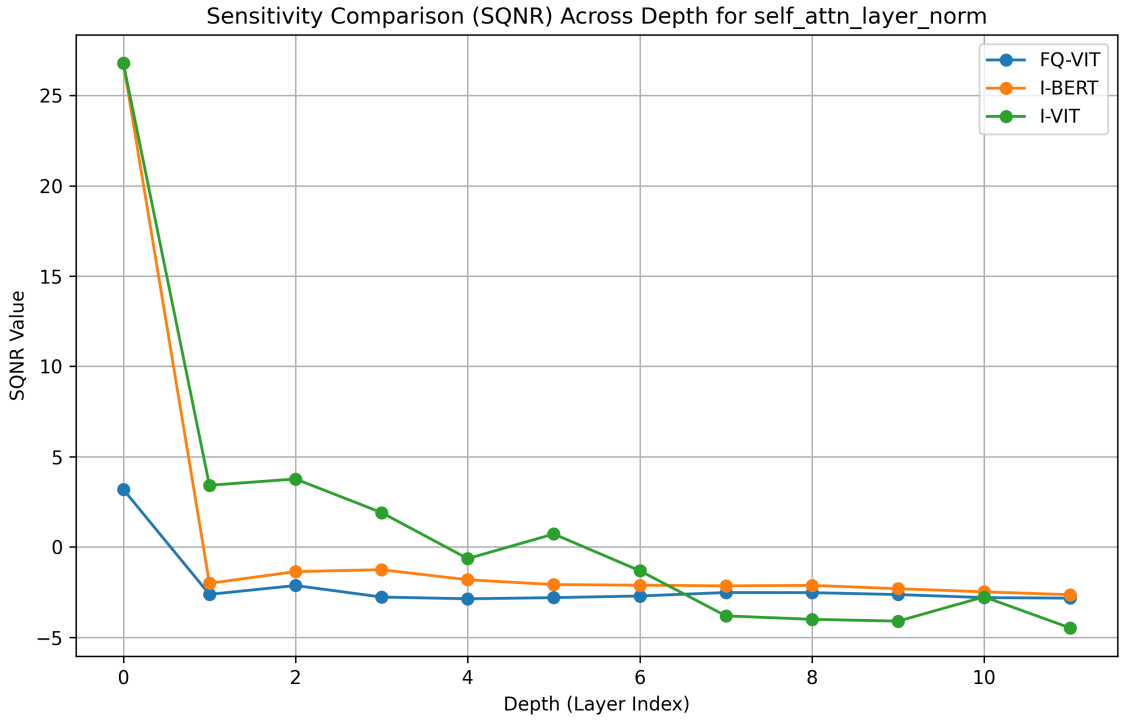

Fig. 3: Attention Layer SQNR scores across depth with different approximations

autoregressive causal language models. Successfully maintaining full integer precision during inference would enable efficient deployment on microcontroller-based Neural Processing Units (NPUs), thereby leveraging the versatility of language models for various edge AI applications. This research identified limitations in existing static quantization frameworks, particularly in their kernel designs. These frameworks in the current stage require intermediate dequantization to higher precision, preventing true 8-bit inference. In order to truly realize full integer inference for such language models specific kernels need to be designed that would essentially remove the QDQ bottleneck within the current framework. However, due to the poor inference results on downstream tasks and significant developmental overhead, such kernels were not pursued in this work as they would essentially only provide speedups that would have no impact on overall model accuracy. Recent studies exploring dynamic quantization [13] have highlighted similar limitations in static quantization techniques when applied to causal models, notably demonstrating comparable performance degradation with the LLaMA 7B model. However, due to the lack of reproducible implementations, we could not include these dynamic quantization approaches in our comparative analysis.

From the experiments, we were able to conclusively answer the research questions devised by having a proof of concept evaluation done on a sub-billion parameter causal language model further, while there might be some further optimizations that can be performed in the context of the QAT experiments where the hyperparameters for the training setup can be further enhanced to potentially yield better results. Due to the training cost associated with the experiments they remain unexplored in the context of this study. Advancing quantization techniques requires a multi-faceted approach addressing architectural diversity, hardware heterogeneity, and preservation of model accuracy. Future research must develop end-to-end quantization solutions that maintain performance across diverse computing platforms while enabling efficient deployment in resource-constrained environments. The most promising avenue for future research based upon this work lies in developing hybrid mixed-precision models where fine-grained layer-wise non-uniform quantization can be performed by mixing different approximation techniques for transformer-based models into a single model in order to best recover model performance. Further, by selectively maintaining higher precision for sensitive modules while quantizing less critical components, significant performance improvements can be achieved at the cost of losing on the benefits yielded by the original full integer quantization. For instance, maintaining critical layers either in bfloat16 or int16 should still yield memory and compute optimizations and further tradeoffs could be explored thereon. This necessitates a detailed investigation into adaptive quantization strategies, where there is adaptive precision handling for quantization based on sensitivity, and sophisticated loss functions that minimize sensitivity in non-

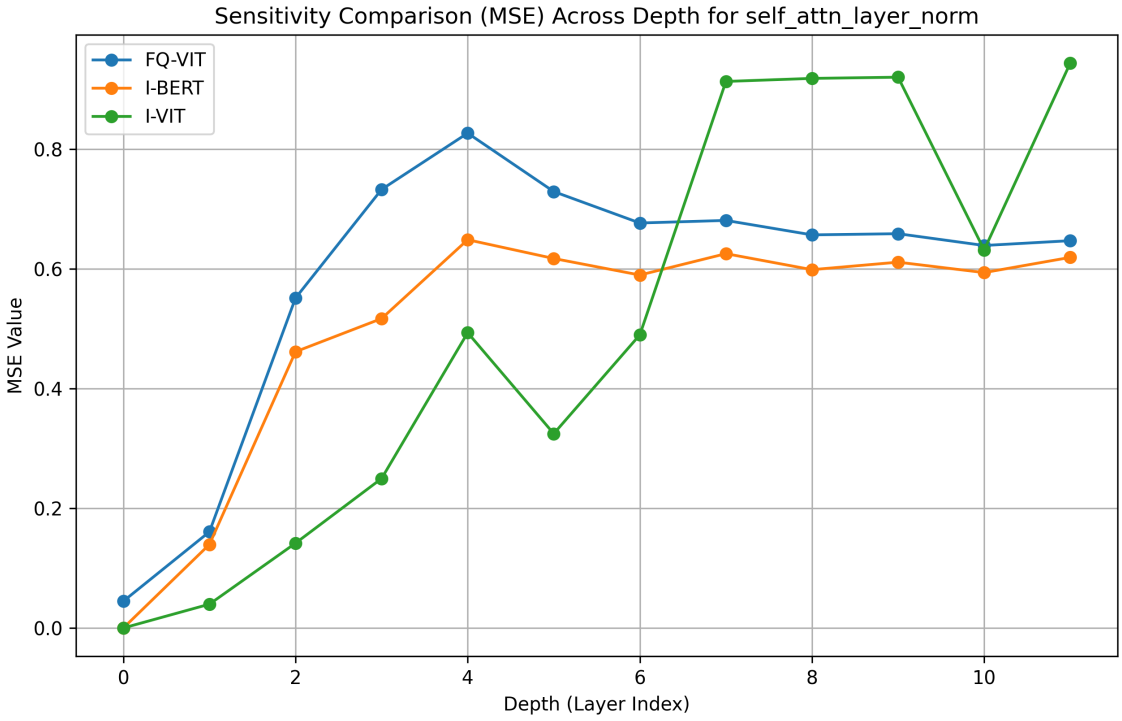

Fig. 4: Attention Layer MSE scores across depth with different approximations

linear layers. In this work, we systematically evaluate the integration of approximation techniques designed in research in the context of autoregressive causal language models. While this work makes use of the OPT-125M model that uses the ReLU activation unlike other language models in existence with the GELU approximation, therefore this work did not make use of the approximation for this layer from the original works that had an equivalent approximation for the GELU activation function.

VIII. Conclusion

This study presents the first comprehensive analysis of static quantization techniques for sub-billion parameter causal language models, emphasizing the feasibility of edge deployment through INT8 inference for text generation tasks. By systematically evaluating quantization approximation methods across model depths and layers, critical insights were uncovered regarding the performance limitations of existing static quantization approaches. Although the literature survey identified promising methods that could facilitate static quantization while maintaining integer precision inference, this study highlights that none of the three techniques evaluated successfully match the floating-point precision baselines on downstream tasks. Among these, I-VIT emerges as the most effective technique under the given model and task constraints.

Experimental results reveal performance variations across downstream tasks, with Quantization-Aware Training (QAT) offering only marginal improvements. However, findings indicate that current static quantization methods remain suboptimal for the selected small causal language model. This aligns with prior observations in the I-LLM study on the LLaMA 7B model, where integrating static quantization approximations from I-BERT led to perplexities around 8e4 [13]. Similarly, in the original works of I-VIT and FQ-VIT, applying approximations to vision-based transformers like DeiT-T (5.7M parameters) yielded accuracies of 72.24% and 71.61%, respectively, compared to the original FP32 accuracy of 72.21%. These results pertain to simpler vision-based tasks, where I-VIT's quantization-aware training and FQ-VIT's calibration were conducted on the extensive 14 million-record ImageNet dataset [6]. In contrast, this study employed smaller, more representative datasets due to the high training costs, resulting in longer lead times.

Moreover, classification tasks in vision models are inherently simpler than those in language models. While overall performance declined post-approximation across most tasks, slight improvements (1%) were observed in specific tasks like Arc Challenge (24.40 vs. 22.78), BoolQ (55.20 vs. 55.44), and Winogrande (50 vs. 50.36), suggesting task-specific dependencies that warrant further investigation. This points to potential task-specific applications of these quantization techniques in language models to preserve performance.

Additionally, in I-BERT, the RoBERTa-base model (125M parameters) underwent training with their proposed approxi-

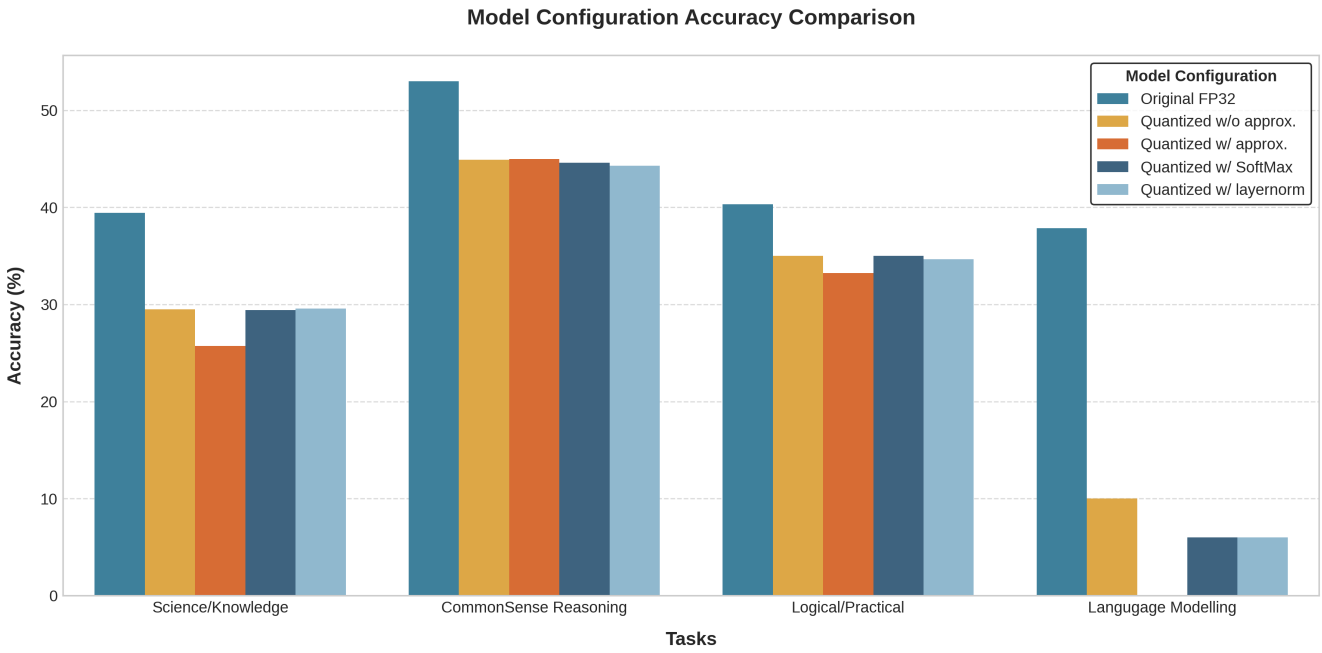

Fig. 5: Ablation Study on the OPT-125M model

mations on the SuperGLUE dataset [42], achieving a modest 0.3% improvement on the GLUE benchmark. However, it's important to note that these encoder-based models are bidirectional and primarily designed for language and semantic understanding, aligning more with classification tasks in vision transformers. Conversely, the OPT-125M model used in this study is causal and autoregressive, differing significantly in design and objectives, which may explain the performance disparities.

The sensitivity analyses conducted provide a vital framework for understanding the impact of quantization on model performance. This analysis enabled the identification of problematic non-linear layers within the attention block and highlighted how error propagation intensifies with network depth due to quantization approximations. The framework, inspired by multiple studies, offers a clear methodology for pinpointing critical modules, facilitating future research and reproducibility in optimizing such approximations for causal language models.

Experimental findings reveal that all literature-sourced approximations evaluated here yield suboptimal performance, with an average relative accuracy drop of 17% across most NLP benchmark tasks. Furthermore, the model's autoregressive and causal nature deteriorates, as evidenced by perplexity results on cloze-style tasks. This compromises the model's ability to generate meaningful predictions, undermining its generalizability and contextual understanding—key attributes that make these models versatile.

Datasets play a crucial role in mitigating performance degradation post-quantization. Higher-quality datasets like Pile-Val and c4 lead to better model representations and more coherent sentence generation in comparison with the original Pile-10K, reducing the relative performance drop to within 7% on most NLP benchmark tasks. This enhancement improved post-training quantization results by 8% in accuracy and delivered a 12× improvement in perplexity on Wikitext, a standard benchmark for this metric.

Investigating observer effects, inspired by FQ-VIT, showed no significant improvements across benchmark tasks. However, percentile-based and min-max observers provided slight (1%) relative performance gains. Further analysis of non-linear approximations revealed that combining approximations yielded the worst performance, while selectively disabling quantization for layers like softmax and layer norm resulted in intermediate performance compared to fully linear modules and fully quantized models as found in figure 5 and V.

Similar to other quantization studies, this work does not address inherent model biases arising from training data, leaving this aspect outside the study's scope. As model quantization rapidly advances, this research serves as a foundational reference, offering a structured approach to understanding the challenges of applying static quantization approximations in causal language models to achieve full-integer inference within the framework explored in this study.

REFERENCES

[1] Allen Institute for AI. *C4: Colossal Clean Crawled Corpus*. https://huggingface.co/datasets/allenai/c4. Accessed: 2024-11-20. 2020.

[2] Ron Banner, Yury Nahshan, and Daniel Soudry. "Post Training 4-bit Quantization of Convolutional Networks for Rapid-Deployment". In: *Advances in Neural Information Processing Systems*. 2019.

[3] Yonatan Bisk et al. "PIQA: Reasoning about Physical Commonsense in Natural Language". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 34.05 (2020), pp. 7432–7439.

[4] Christopher Clark et al. "BoolQ: Exploring the Surprising Difficulty of Natural Yes/No Questions". In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 2019, pp. 2924–2936.

[5] Peter Clark et al. "Think you have Solved Question Answering? Try ARC, the AI2 Reasoning Challenge". In: *Thirty-Second AAAI Conference on Artificial Intelligence*. 2018.

[6] Jia Deng et al. "ImageNet: A large-scale hierarchical image database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE. 2009, pp. 248–255.

[7] Peiyan Dong et al. "Packqvit: Faster sub-8-bit vision transformers via full and packed quantization on the mobile". In: *Advances in Neural Information Processing Systems* 36 (2024).

[8] Ansel et.al. "PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation". In: *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2 (ASPLOS '24)*. ACM, Apr. 2024. DOI: 10.1145/3620665.3640366. URL: https://pytorch.org/assets/pytorch2-2.pdf.

[9] Hugging Face. *Accuracy*. Hugging Face. 2024. URL: https://huggingface.co/docs/transformers/en/accuracy (visited on 02/10/2024).

[10] Hugging Face. *Perplexity*. Hugging Face. 2024. URL: https://huggingface.co/docs/transformers/en/perplexity (visited on 02/10/2024).

[11] Elias Frantar et al. "OPTQ: Accurate quantization for generative pre-trained transformers". In: *The Eleventh International Conference on Learning Representations*. 2022.

[12] Leo Gao et al. *A framework for few-shot language model evaluation*. Version v0.4.3. July 2024. DOI: 10.5281/zenodo.12608602. URL: https://zenodo.org/records/12608602.

[13] Xing Hu et al. "I-LLM: Efficient Integer-Only Inference for Fully-Quantized Low-Bit Large Language Models". In: *arXiv preprint arXiv:2405.17849* (2024).

[14] Hugging Face. *SmolLM: Efficient Small Language Models*. 2024. URL: https://github.com/huggingface/smollm (visited on 02/10/2024).

[15] Intel. *Intel Neural Compressor*. 2020. URL: https://intel.github.io/neural-compressor/ (visited on 08/10/2024).

[16] Sehoon Kim et al. "I-bert: Integer-only bert quantization". In: *International conference on machine learning*. PMLR. 2021, pp. 5506–5518.

[17] Atli Kosson and Martin Jaggi. "Multiplication-Free Transformer Training via Piecewise Affine Operations". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh et al. Vol. 36. Curran Associates, Inc., pp. 8208–8223. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/19df21cd4931bd0caaa4d8480e9a59cd-Paper-Conference.pdf.

[18] Raghuraman Krishnamoorthi. "Quantizing Deep Convolutional Networks for Efficient Inference: A Whitepaper". In: *arXiv preprint arXiv:1806.08342* (2018).

[19] MIT HAN Lab. *Pile Validation Backup Dataset*. https://huggingface.co/datasets/mit-han-lab/pile-val-backup. Accessed: 2024-11-20. 2021.

[20] Changhun Lee et al. "Owq: Outlier-aware weight quantization for efficient fine-tuning and inference of large language models". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 38. 12. 2024, pp. 13355–13364.

[21] Shiyao Li et al. "Evaluating quantized large language models". In: *arXiv preprint arXiv:2402.18158* (2024).

[22] Yanjing Li et al. "Q-vit: Accurate and fully quantized low-bit vision transformer". In: *Advances in neural information processing systems* 35 (2022), pp. 34451–34463.

[23] Zhikai Li and Qingyi Gu. "I-vit: Integer-only quantization for efficient vision transformer inference". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2023, pp. 17065–17075.

[24] Ji Lin et al. "AWQ: Activation-aware Weight Quantization for On-Device LLM Compression and Acceleration". In: *Proceedings of Machine Learning and Systems* 6 (2024), pp. 87–100.

[25] Yang Lin et al. "Fq-vit: Post-training quantization for fully quantized vision transformer". In: *arXiv preprint arXiv:2111.13824* (2021).

[26] Ye Lin et al. "Towards fully 8-bit integer inference for the transformer model". In: *arXiv preprint arXiv:2009.08034* (2020).

[27] Jian Liu et al. "LogiQA: A Challenge Dataset for Machine Reading Comprehension with Logical Reasoning". In: *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence* (2022), pp. 4239–4245.

[28] Zechun Liu et al. "Llm-qat: Data-free quantization aware training for large language models". In: *arXiv preprint arXiv:2305.17888* (2023).

[29] Eldad Meller et al. “Same, Same but Different: Recovering Neural Network Quantization Error Through Weight Factorization”. In: *International Conference on Machine Learning*. 2019.

[30] Szymon Migacz. “8-bit Inference with TensorRT”. In: *GPU Technology Conference*. NVIDIA. 2017.

[31] Todor Mihaylov et al. “Can a Suit of Armor Conduct Electricity? A New Dataset for Open Book Question Answering”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. 2018, pp. 2381–2391.

[32] Neel Nanda. *The Pile-10k Dataset*. https://huggingface.co/datasets/NeelNanda/pile-10k. Accessed: 2024-11-20. 2021.

[33] ONNX Runtime. *ONNX Model Optimization: Quantization*. 2024. URL: https://onnxruntime.ai/docs/performance/model-optimizations/quantization.html (visited on 07/10/2024).

[34] Open LLM Leaderboard. *Open LLM Leaderboard 2*. Hugging Face. 2024. URL: https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard (visited on 02/10/2024).

[35] Denis Paperno et al. “The LAMBADA dataset: Word prediction requiring a broad discourse context”. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. 2016, pp. 1525–1534.

[36] Andy Rock et al. “INT8 Transformers for Inference Acceleration”. In: *36th Conference on Neural Information Processing Systems (NeurIPS)*. 2022.

[37] Keisuke Sakaguchi et al. “WinoGrande: An Adversarial Winograd Schema Challenge at Scale”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05. 2020, pp. 8732–8740.

[38] Wenqi Shao et al. “Omniquant: Omnidirectionally calibrated quantization for large language models”. In: *arXiv preprint arXiv:2308.13137* (2023).

[39] Sheng Shen et al. “Q-bert: Hessian based ultra low precision quantization of bert”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 05. 2020, pp. 8815–8821.

[40] Xuan Shen et al. “EdgeQAT: Entropy and Distribution Guided Quantization-Aware Training for the Acceleration of Lightweight LLMs on the Edge”. In: *arXiv preprint arXiv:2402.10787* (2024).

[41] Ashish Vaswani et al. “Attention is all you need”. In: *Advances in neural information processing systems* 30 (2017).

[42] Alex Wang et al. “GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding”. In: *International Conference on Learning Representations (ICLR)*. 2019.

[43] Wenxun Wang et al. “SOLE: Hardware-Software Co-design of Softmax and LayerNorm for Efficient Transformer Inference”. In: *2023 IEEE/ACM International Conference on Computer Aided Design (ICCAD)*. IEEE. 2023, pp. 1–9.

[44] Zheng Wang et al. “Squat: Sharpness-and quantization-aware training for bert”. In: *arXiv preprint arXiv:2210.07171* (2022).

[45] Johannes Welbl, Nelson F. Liu, and Matt Gardner. “Crowdsourcing Multiple Choice Science Questions”. In: *Proceedings of the 3rd Workshop on Noisy User-generated Text*. 2017, pp. 94–106.

[46] Guangxuan Xiao et al. “Smoothquant: Accurate and efficient post-training quantization for large language models”. In: *International Conference on Machine Learning*. PMLR. 2023, pp. 38087–38099.

[47] Haoran You et al. “ShiftAddLLM: Accelerating Pretrained LLMs via Post-Training Multiplication-Less Reparameterization”. In: *arXiv preprint arXiv:2406.05981* (2024).

[48] Joonsang Yu et al. “Nn-lut: neural approximation of non-linear operations for efficient transformer inference”. In: *Proceedings of the 59th ACM/IEEE Design Automation Conference*. 2022, pp. 577–582.

[49] Rowan Zellers et al. “HellaSwag: Can a Machine Really Finish Your Sentence?” In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019, pp. 4791–4800.

[50] Susan Zhang et al. “Opt: Open pre-trained transformer language models”. In: *arXiv preprint arXiv:2205.01068* (2022).

[51] Shuchang Zhou et al. “DoReFa-Net: Training Low Bitwidth Convolutional Neural Networks with Low Bitwidth Gradients”. In: *arXiv preprint arXiv:1606.06160* (2016).

APPENDIX

A. Training and Dataset Details

- **C4 (en) Approximation:** The original dataset contains 364,868,892 samples for the ‘en’ configuration. Due to limited computational resources, a downsampled subset of 1,000,000 examples was selected. These examples were tokenized and grouped into blocks of size 512, resulting in 843,844 training samples and 95,204 validation samples.
- **Pile-10k Approximation:** The original Pile-10k dataset contains 10,000 samples, which were tokenized and grouped into blocks of size 128 for training purposes, resulting in approximately 9346 training samples.
- **Pile-val Approximation:** The Pile-val dataset contains a total of 214,670 samples. After tokenization and grouping into blocks of size 512, 673,968 training samples were obtained, with 0.1% of these used for validation.

The training and evaluation performance are analyzed in terms of Training Loss, Validation Loss, and Perplexity. The models on the Pile-10k were trained for **30 epochs**. However, due to larger training costs and to allow for a comprehensive comparison across datasets, the others were trained until the optimal convergence scenario with early stopping.

B. *Question-Answering and Reasoning Tasks*

This domain rigorously assesses the model's capacity for sophisticated question comprehension and multi-step logical inference.

a) AI2 Reasoning Challenge (ARC): [5]:

The ARC dataset comprises two distinct subsets: *ARC Easy*, containing 2,376 grade-school science questions with established human performance exceeding 90%, and *ARC Challenge*, comprising 1,172 more complex queries requiring sophisticated multi-hop reasoning and causal understanding. Accuracy serves as the primary evaluation metric.

b) BoolQ: [4]:

BoolQ represents a boolean question-answering corpus of 15,942 binary-choice questions paired with corresponding Wikipedia passages. Performance evaluation incorporates both accuracy and F1 score metrics, with the F1 score formulated as:

$$\text{F1} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

However, the evaluation framework utilized in this work only reports accuracy for this task.

c) Hellaswag: [49]:

Hellaswag evaluates commonsense reasoning through 70,000 multiple-choice scenarios, each presenting a contextual paragraph and four potential completions. The benchmark achieves human performance at 95.6%, with accuracy as the definitive evaluation metric.

C. *Language Understanding and Generation Tasks*

This category critically examines the model's linguistic comprehension and generative capabilities across complex contextual representations.

a) LAMBADA: [35]:

The LAMBADA dataset evaluates predictive language modelling by challenging models to predict the terminal word in a passage based on extensive contextual information. Comprising 10,022 passages, the evaluation metrics include accuracy and perplexity. Accuracy is defined as:

$$\text{Accuracy}_{\text{LAMBADA}} = \frac{|\{x : \text{pred}(x) = \text{target}(x)\}|}{|D_{\text{test}}|}$$

Perplexity, a standard metric for language models, quantifies how well a model predicts a sequence of words. It is computed as:

$$\text{Perplexity} = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log P(w_i)\right)$$

where N is the total number of words in the dataset, and $P(w_i)$ represents the predicted probability of the i-th word. Lower perplexity values indicate better predictive performance.

b) LogiQA: [27]:

LogiQA focuses on logical reasoning through 8,678 multiple-choice questions spanning deductive reasoning, conditional logic, quantitative analysis, and categorical syllogisms. Evaluation metrics encompasses accuracy.

c) Winogrande: [37]: Winogrande assesses commonsense reasoning via sentence completion tasks with binary discriminative choices. The dataset includes 44,000 instances, achieving human-level performance at 94%. Accuracy serves as the evaluation metric.

D. *Scientific Knowledge Assessment*

This domain critically assesses the model's comprehensive understanding of scientific principles across diverse disciplinary contexts.

a) OpenBookQA: [31]:

OpenBookQA contains 5,957 multiple-choice science questions requiring integrated factual knowledge and commonsense reasoning. Each query is contextually supported by supplementary "open book" reference information. Accuracy constitutes the primary evaluation metric.

b) SciQ: [45]:

SciQ presents 13,679 multiple-choice questions spanning physics, chemistry, and biology, with strategically generated distractors derived through crowd-sourcing methodologies. Accuracy remains the definitive evaluation metric.

*1) **Physical Reasoning Assessment**:*

a) PIQA: [3]:

PIQA evaluates physical commonsense reasoning through binary-choice scenarios, focusing on a computational understanding of physical principles and their pragmatic real-world applications. Accuracy provides the fundamental performance metric. The evaluated tasks demonstrate diverse output configurations, characterized by varying output space cardinalities $|\mathcal{O}|$, ranging from binary discriminative choices to expansive vocabulary-based representations for open-ended computational tasks.

In transformer-based models, non-linear operations such as LayerNorm and Softmax are crucial for stable training and inference. However, these operations often pose challenges for integer-only (INT8) quantization. In this section, we describe the baseline equations for LayerNorm and Softmax, followed by the integer-only approximations used in three different quantized model configurations obtained from their original works: I-BERT, I-ViT, and FQ-ViT.

E. Baseline Equations

a) Softmax.: Softmax is commonly defined as:

$$\text{Softmax}(x)_i = \frac{\exp(x_i)}{\sum_{j=1}^{k}\exp(x_j)}. \tag{21}$$

To avoid numerical overflow, a shifted version is often used:

$$\tilde{x}_i = x_i - \max_j(x_j), \quad \text{Softmax}(x)_i = \frac{\exp(\tilde{x}_i)}{\sum_{j=1}^{k}\exp(\tilde{x}_j)}. \tag{22}$$

b) LayerNorm.: Layer normalization (LayerNorm) normalizes input features by their mean and variance:

$$\text{LayerNorm}(X) = \frac{X - \mu_X}{\sqrt{\sigma_X^2 + \epsilon}} \cdot \gamma + \beta, \tag{23}$$

where μ_X and σ_X are the mean and standard deviation of the elements in X, and γ, β are learnable parameters. Equivalently, one can rewrite:

$$\text{LayerNorm}(X) = \frac{\gamma}{\sqrt{\sigma_X^2 + \epsilon}} X + \frac{\beta}{\sqrt{\sigma_X^2 + \epsilon}} - \frac{\gamma \mu_X}{\sqrt{\sigma_X^2 + \epsilon}}. \tag{24}$$

F. *I-BERT Approximations*

a) Integer-only Softmax (I-BERT).: I-BERT employs a polynomial approximation of the exponential function in an integer-only domain. Starting from the shifted Softmax formulation (Eqs. (21)–(22)), the exponential function $\exp(x)$ is approximated by:

$$\exp(x) \approx 2^{-z} \cdot L(p), \tag{25}$$

where

$$x = (-\ln 2)\, z + p, \quad L(p) = 0.3585\,(p + 1.353)^2 + 0.344. \tag{26}$$

Here, z is the integer quotient that controls the shift (bit-shift in integer domain), and $L(p)$ is a second-order polynomial approximation to $\exp(p)$. This polynomial approximation avoids floating-point exponentiation while retaining sufficient accuracy for Softmax.

b) Integer-only LayerNorm (I-BERT).: The LayerNorm operation is defined as in Eq. (23). In I-BERT, the standard deviation (or variance) computation is approximated through an iterative integer-based approach:

$$\sigma \approx \sqrt{\text{Var}(x)} \quad \longrightarrow \quad I_{i+1} = \frac{I_i + \left\lfloor \frac{\text{Var}(I_x)}{I_i} \right\rfloor}{2}. \tag{27}$$

This iterative method refines the estimate of σ solely using integer arithmetic and requires a small number of iterations to converge to an acceptably accurate value. Once σ is computed, the normalization follows via integer arithmetic with scaling and shifting factors.

G. *I-ViT Approximations*

a) ShiftMax: Integer-only Softmax (I-ViT).: I-ViT proposes a shift-based Softmax (ShiftMax) that relies on integer bit-shifts to approximate exponentials. The main idea is to decompose the exponent into integer parts and use shift operations instead of floating-point exponentiation. For an input integer I_{in}, the scaled exponent is computed in an integer manner, and the final Softmax output I_{out} is obtained by:

$$I_{\text{out}} = \text{IntDiv}\Big(I_{\text{exp}}, \sum_j^d I_{\text{exp}_j},\, k_{\text{out}}\Big), \tag{28}$$

where $\text{IntDiv}(\cdot)$ denotes an integer division approximation, and k_{out} controls the output scaling. Thus,

$$I_{\text{out}} \cdot S_{\text{out}} \;\approx\; \text{Softmax}(I_{\text{in}} \cdot S_{\text{in}}). \tag{29}$$

b) Integer-only LayerNorm (I-ViT).: Similar to I-BERT, the LayerNorm in I-ViT also relies on iterative integer-based computations for mean and standard deviation:

$$\text{LayerNorm}(x) = \frac{x - \text{Mean}(x)}{\sqrt{\text{Var}(x)}} \cdot \gamma + \beta, \tag{30}$$

with

$$I_{i+1} = \frac{I_i + \left\lfloor \frac{\text{Var}(x)}{I_i} \right\rfloor}{2}. \tag{31}$$

The difference is that I-ViT heavily uses shift-based arithmetic (bit shifting) to simplify multiplications and divisions, thus reducing hardware complexity.

H. *FQ-ViT Approximations*

FQ-ViT extends integer-only quantization to full transformer-based architectures, focusing on practical approximations for both LayerNorm and Softmax. Below, we use the specific formulations from *Quantized Inference for LayerNorm* and *Quantized Inference for Softmax*, often referred to as *Power-of-Two LayerNorm* and *Log-Int-Softmax*, respectively.

a) Power-of-Two LayerNorm (FQ-ViT).: Recall the LayerNorm definition:

$$\text{LayerNorm}(X) = \frac{X - \mu_X}{\sqrt{\sigma_X^2 + \epsilon}} \cdot \gamma + \beta. \tag{32}$$

FQ-ViT enforces Power-of-Two Factors (PTF) for the quantized activation:

$$X_Q = Q(X|b) = \text{clip}\Big(b \cdot \frac{X}{2^\alpha s} + z_p,\, 0,\, 2^b - 1\Big), \tag{33}$$

where s, z_p are scalar quantization parameters, and α is the power-of-two factor. Shifting the quantized activation by α yields

$$\hat{X}_Q = (X_Q - z_p) \ll \alpha. \tag{34}$$

The mean and variance are then computed approximately as:

$$\mu_X \approx \frac{s}{C} \sum_{i=1}^{C} \hat{X}_{Q_i} \;\to\; \frac{s}{C} M_1, \quad \mu_{X^2} \approx \frac{s^2}{C} \sum_{i=1}^{C} \hat{X}_{Q_i}^2 \;\to\; \frac{s^2}{C} M_2, \tag{35}$$

$$\sigma_X^2 = \mu_{X^2} - \mu_X^2 \;\approx\; \frac{s^2}{C^2}\big(C\, M_2 - M_1^2\big), \tag{36}$$

$$\sqrt{\sigma_X^2 + \epsilon} \;\approx\; \frac{s}{C} \sqrt{C\, M_2 - M_1^2}. \tag{37}$$

Thus, the integrated LayerNorm inference is approximated as:

$$Y_Q = \left\lfloor \frac{s_{\text{in}}\, \gamma}{s_{\text{out}} \sqrt{\sigma_X^2 + \epsilon}}\, \hat{X}_Q \;+\; \frac{\beta}{s_{\text{out}} \sqrt{\sigma_X^2 + \epsilon}} \;-\; \frac{\gamma\, \mu_X}{s_{\text{out}} \sqrt{\sigma_X^2 + \epsilon}} \right\rfloor + z_{p_{\text{out}}}. \tag{38}$$

By fusing constants,

$$A = \frac{s_{\text{in}}\, \gamma}{s_{\text{out}} \sqrt{\sigma_X^2 + \epsilon}}, \quad B = \frac{\beta - \gamma\, \mu_X}{s_{\text{out}} \sqrt{\sigma_X^2 + \epsilon}}, \tag{39}$$

the final quantized inference becomes:

$$Y_Q = \Big\lfloor A\, \hat{X}_Q + B \Big\rfloor + z_{p_{\text{out}}}. \tag{40}$$

b) Log-Int-Softmax (FQ-ViT).: Softmax in FQ-ViT is similarly handled by an integer-friendly approach. Recall:

$$\text{Softmax}(X)_i = \frac{\exp(X_i)}{\sum_{j=1}^{J} \exp(X_j)}. \tag{41}$$

For numerical stability:

$$\tilde{X}_i = X_i - \max(X), \quad \text{Softmax}(X)_i = \frac{\exp(\tilde{X}_i)}{\sum_{j=1}^{J} \exp(\tilde{X}_j)}. \tag{42}$$

The exponential is decomposed as:

$$\exp(\tilde{X}) = 2^{-z} \exp(p) = \exp(p) \gg z, \tag{43}$$

where $z = \left\lfloor -\frac{\tilde{X}}{\ln 2} \right\rfloor$, $p = \tilde{X} + z \ln 2$. A polynomial approximation is used for $\exp(p)$:

$$L(p) = 0.3585\,(p + 1.353)^2 + 0.344 \;\approx\; \exp(p), \tag{44}$$

giving an integer exponent approximation

$$\text{i-exp}(\tilde{X}) := L(p) \gg z. \tag{45}$$

FQ-ViT then introduces a *log-integer-Softmax* (LIS):

$$\text{LIS}(s{\cdot}X_Q) = N{-}\text{clip}\Big(\log_2\lfloor \tfrac{\sum \text{i-exp}(X_Q)}{\text{i-exp}(X_Q)} \rfloor, 0,\, 2^b{-}1\Big) = N{-}\text{Attn}_Q, \tag{46}$$

where $N = 2^b - 1$. The final attention (or Softmax) result can be multiplied by the Values V in the attention block using shifted integer arithmetic which yields an integer-friendly formulation of Softmax suitable for hardware-efficient inference.

The above approximations (I-BERT, I-ViT, and FQ-ViT) demonstrate how critical operations like Softmax and LayerNorm can be implemented using only integer arithmetic. These methods replace floating-point exponentiation, division, and square roots with polynomial approximations, bit-shifting, and iterative refinements, making it feasible to execute full-INT8 inference while maintaining adequate model accuracy. As hardware accelerators evolve, such integer-only methods will further reduce computation and memory overhead during deployment.

I. Signal-to-Quantization-Noise Ratio (SQNR)

ignal-to-Quantization-Noise Ratio (SQNR) is a fundamental metric that quantifies the relationship between the original signal power and the noise introduced by quantization. For a tensor X and its quantized version X_q, SQNR is defined as:

$$SQNR = 10 \log_{10} \left(\frac{\sum_i X_i^2}{\sum_i (X_i - X_{q_i})^2} \right) \tag{47}$$

Additional clarification:

The numerator $\sum_i X_i^2$ represents the total power of the original signal The denominator $\sum_i (X_i - X_{q_i})^2$ represents the total quantization noise power Higher SQNR values (measured in dB) indicate better quantization quality Typically, each additional bit in quantization adds approximately 6.02 dB to the SQNR

SQNR provides valuable insights into the overall quality of quantization, with higher values indicating better preservation of the original signal. This metric has been extensively used in works like [51], where the authors demonstrated its effectiveness in evaluating different bit-width configurations.

J. Hessian Aware Quantization (HAWQ) Metric

HAWQ represents an advanced approach to quantization where the Hessian trace is computed to determine the sensitivity of a particular block in the network towards quantization. The sensitivity metric is computed as:

$$S_{HAWQ} = \lambda_{max}(H) \tag{48}$$

where:

H is the Hessian matrix of the loss function with respect to the layer parameters $\lambda_{max}(H)$ is the maximum eigenvalue of the Hessian matrix H

The Hessian matrix H is defined as:

$$H = \begin{bmatrix} \frac{\partial^2 L}{\partial w_1^2} & \frac{\partial^2 L}{\partial w_1 \partial w_2} & \cdots & \frac{\partial^2 L}{\partial w_2 \partial w_1} & \frac{\partial^2 L}{\partial w_2^2} & \cdots \\ \vdots & \vdots & \ddots & & & \end{bmatrix} \tag{49}$$

where L is the loss function and w_i are the layer parameters. Here, the eigenvalues indicate the curvature of the loss surface. Further, larger eigenvalues suggest higher sensitivity to parameter changes. Moreover, this technique is computationally expensive therefore the Hutchinson method for trace estimation is utilized here.

K. Mean Square Error (MSE)

MSE provides a direct measure of the quantization error magnitude:

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (X_i - X_{q_i})^2 \tag{50}$$

While simpler than other metrics, MSE remains valuable for its interpretability and direct relationship to quantization noise. It has been effectively used in works like [18] to guide bit-width selection.

L. Kurtosis

Kurtosis measures the "tailedness" of the activation distribution, providing insights into the presence of outliers that might be particularly sensitive to quantization:

$$Kurt(X) = \frac{\mathbb{E}[(X - \mu)^4]}{(\mathbb{E}[(X - \mu)^2])^2} - 3 \tag{51}$$

This metric is particularly useful for identifying layers where distributions deviate significantly from normality, as demonstrated in [2]. Higher kurtosis values often indicate greater sensitivity to quantization.

M. Kullback-Leibler (KL) Divergence

KL divergence measures the difference between the original and quantized activation distributions:

$$D_{KL}(P||Q) = \sum_i P(x_i) \log \left(\frac{P(x_i)}{Q(x_i)} \right) \tag{52}$$

where P and Q represent the distributions of the original and quantized tensors respectively. This metric is particularly valuable for assessing how well the quantization preserves the statistical properties of the original tensors, as shown in [30].

N. Cosine Distance

Cosine distance measures the angular deviation between the original and quantized representations:

$$d_{cos}(X, X_q) = 1 - \frac{X \cdot X_q}{\|X\| \|X_q\|} \tag{53}$$

This metric is particularly useful for assessing how well the quantization preserves the directional information of activation vectors, which is crucial for maintaining model accuracy. Research by [29] has demonstrated its effectiveness in quantization analysis.

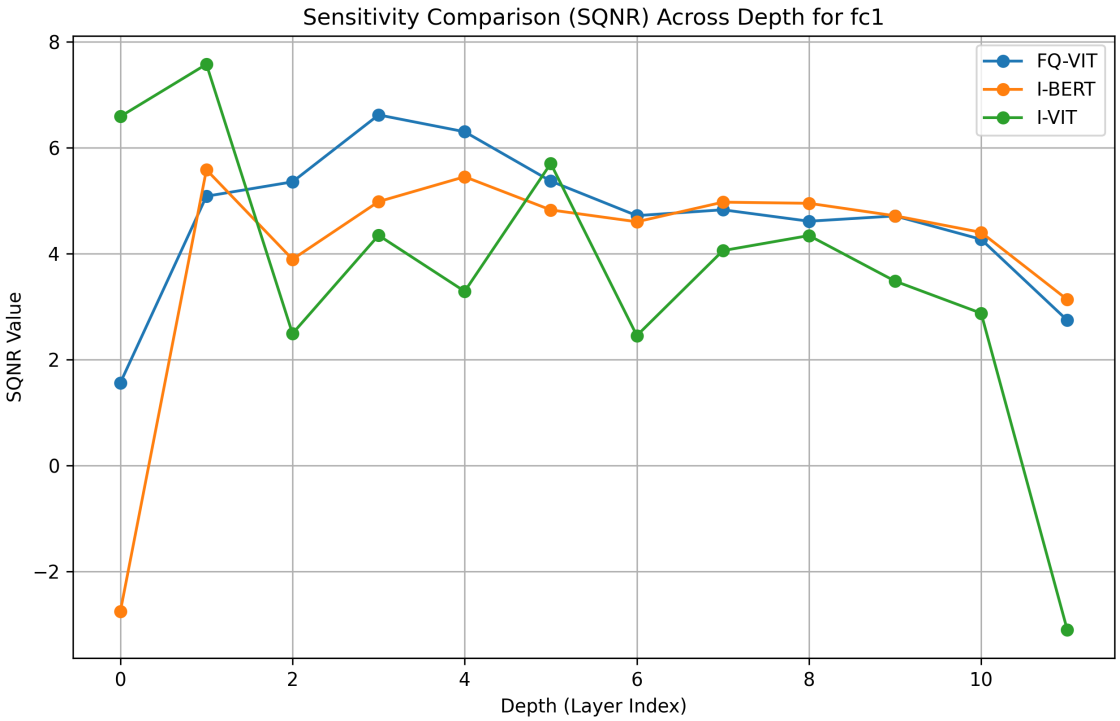

Fig. 6: Quantization sensitivity on the Fully Connected Layer in OPT-125M

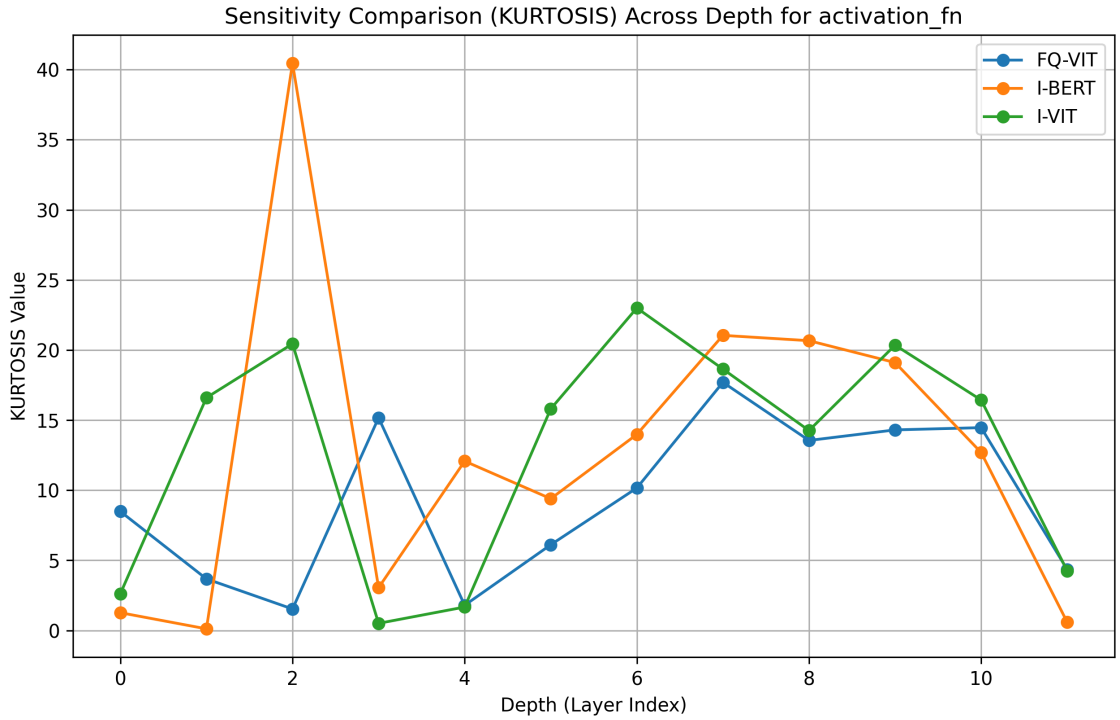

Fig. 7: Quantization sensitivity on the Activation layer in OPT-125M

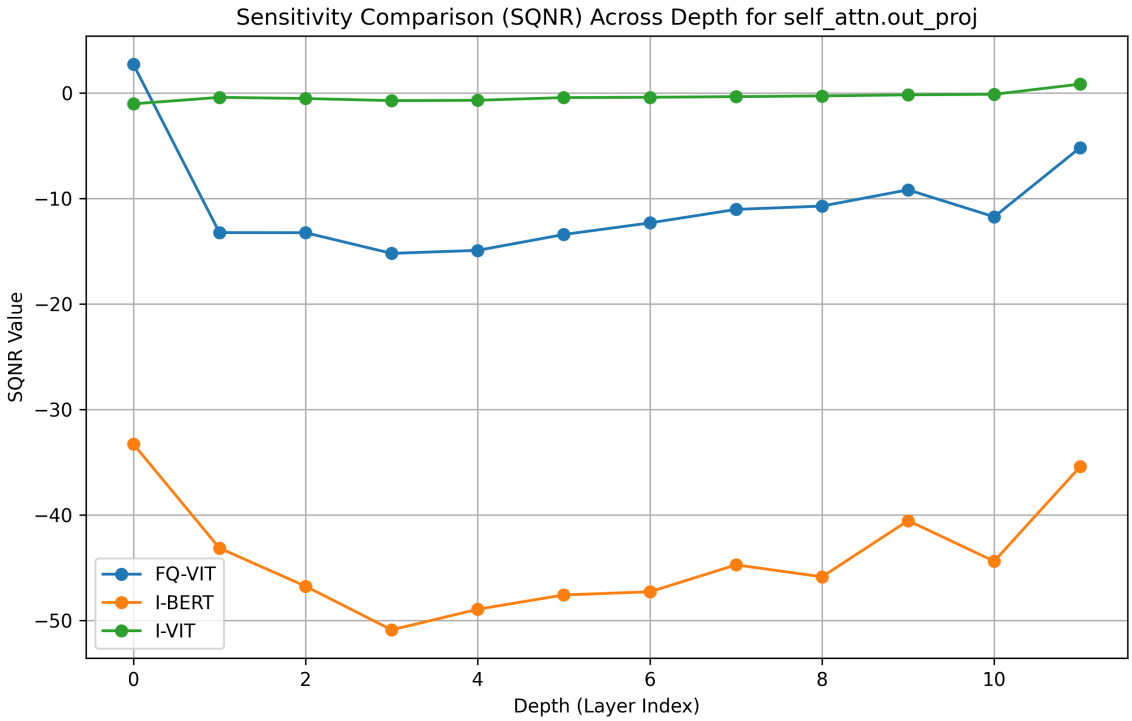

Fig. 8: Quantization sensitivity on the Attention Output layer in OPT-125M

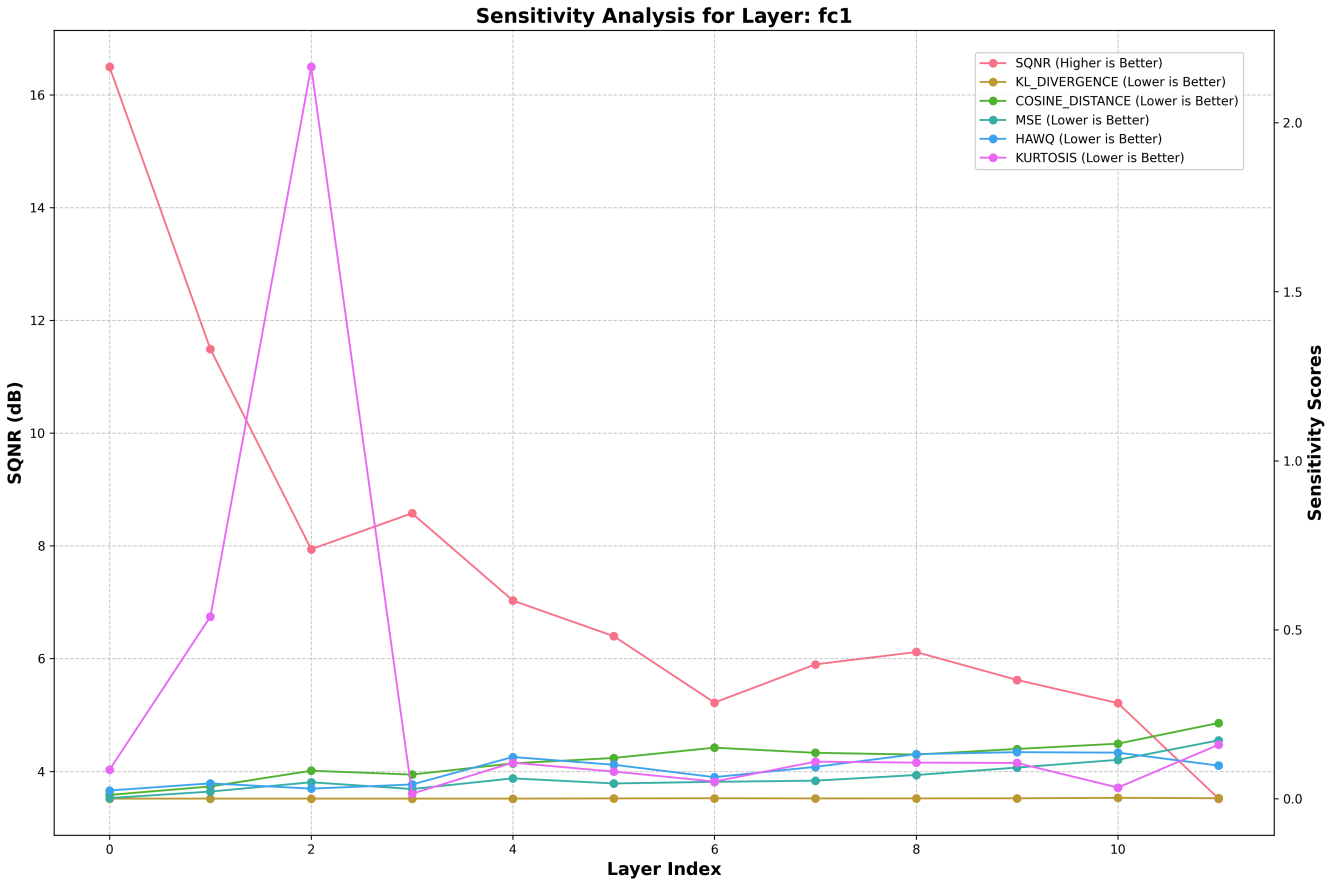

Fig. 9: Quantization sensitivity on the Fully Connected Layer in OPT-125M

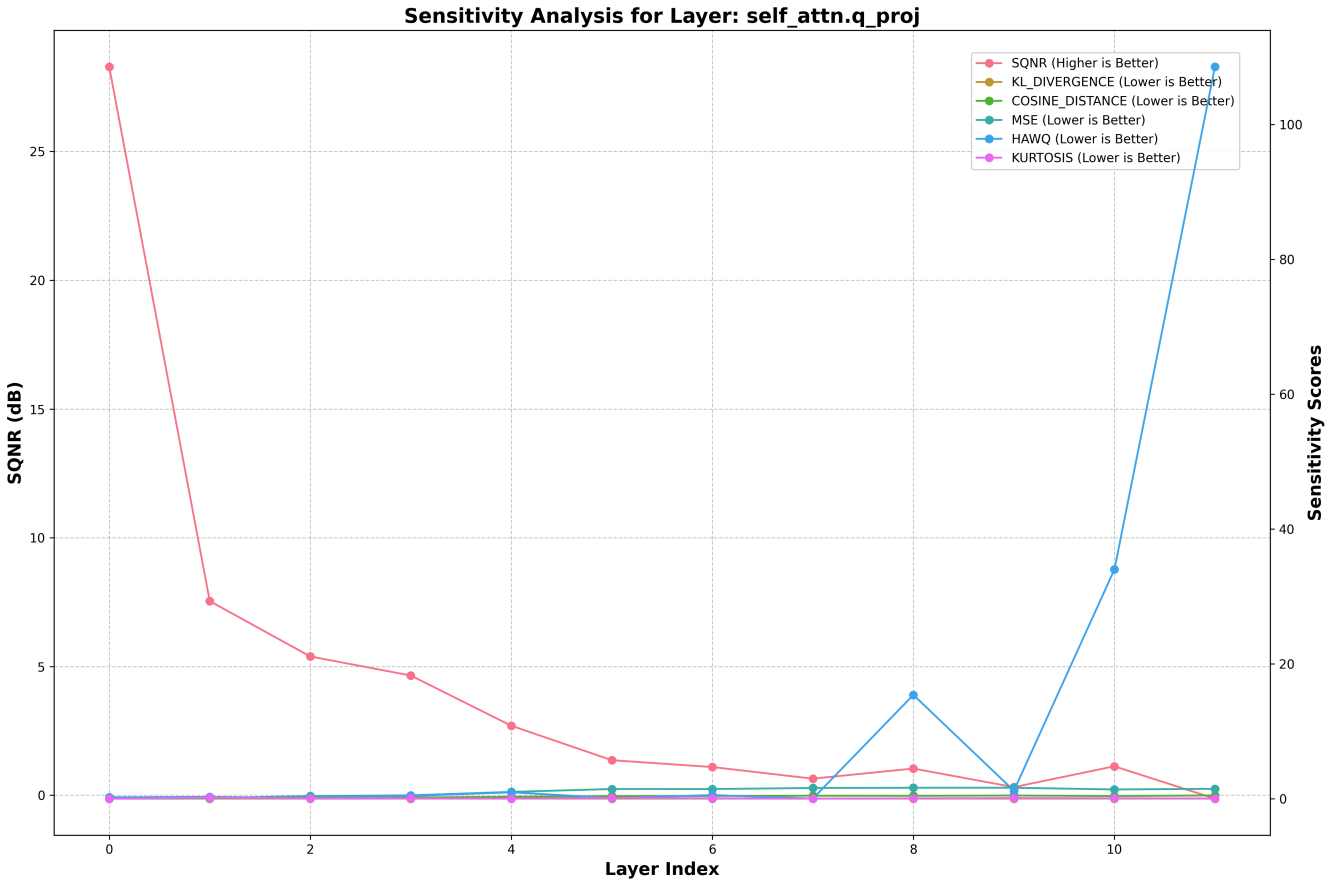

Fig. 10: Quantization sensitivity on the Query Layer projection in OPT-125M

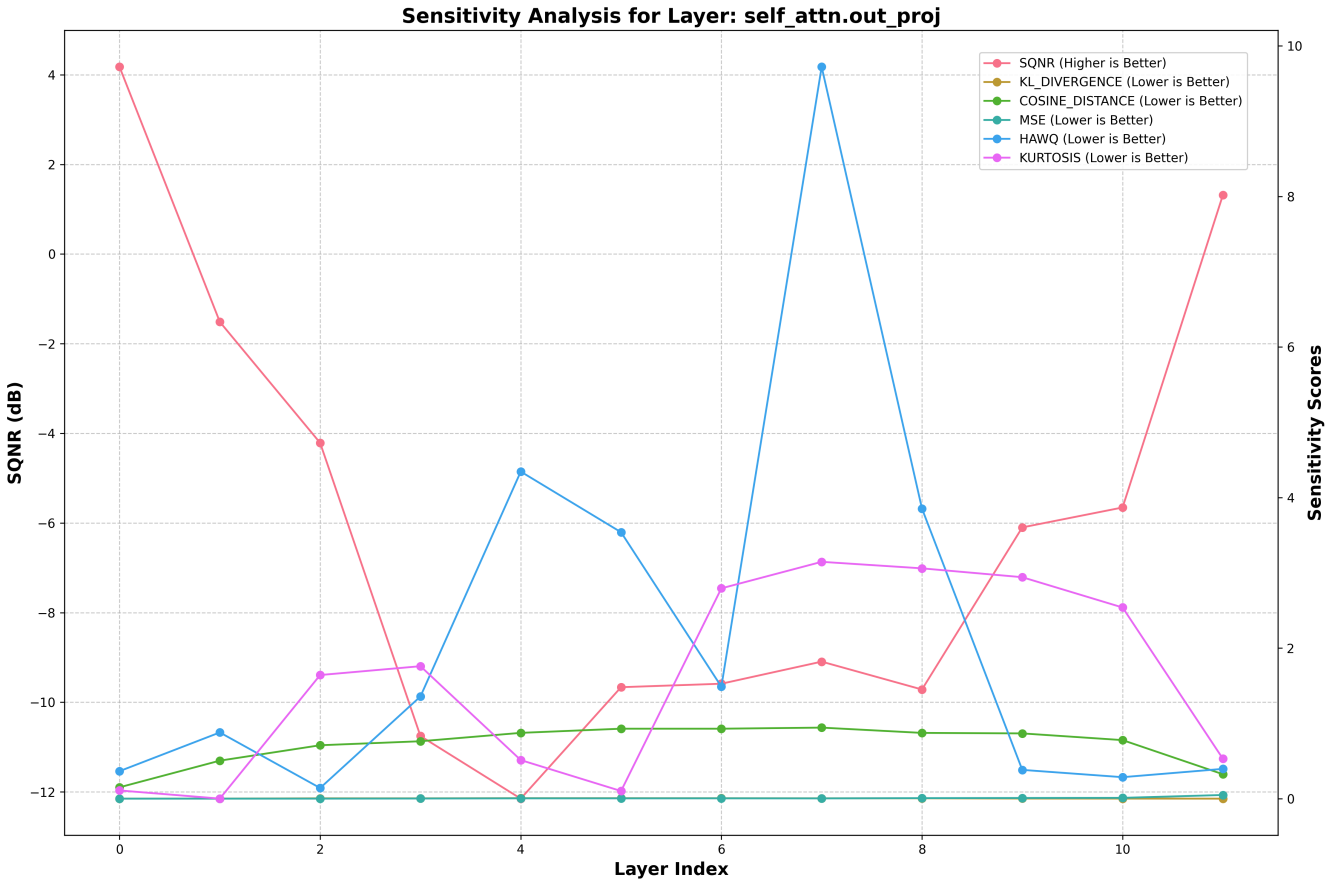

Fig. 11: Quantization sensitivity on the Attention Output Projection in OPT-125M