# GenPTQ: Green Post-Training Quantization for Large-Scale ASR Models with Mixed-Precision Bit Allocation *

**Beom Jin Kang[1], Hyun Kim[*†1],**

[1] Department of Electrical and Information Engineering, Research Center for EIT
Seoul National University of Science and Technology, Seoul, Korea
{beomjin,hyunkim}@seoultech.ac.kr

## Abstract

Large-scale models have achieved state-of-the-art performance in automatic speech recognition (ASR), but their high memory and computation demands pose significant challenges for deployment. To address these challenges, weight-only quantization is widely adopted in large-scale models, where weights dominate memory usage, as it enables efficient compression with minimal accuracy degradation compared to activation quantization. Accordingly, most prior quantization studies for ASR models have focused on weights and employed quantization-aware training (QAT) to restore accuracy. However, QAT incurs substantial additional training costs, posing clear limitations for practical application to large-scale models. Moreover, despite the varying quantization sensitivity across layers, mixed-precision quantization (MPQ) remains underexplored in ASR. In this paper, we propose *GenPTQ*, a mixed-precision post-training quantization method that optimizes the trade-off among accuracy, model size, and optimization cost by leveraging gradient-based sensitivity measurement and transforming the search space into a continuous domain for efficient numerical optimization. Applied to Whisper and Conformer models across multiple speech datasets, *GenPTQ* achieves up to 89.1% model size reduction (2.5-bit average precision) with only a 0.8% increase in WER, and completes optimization in just 15 seconds. These results demonstrate its effectiveness for low-resource ASR deployment.

## 1 Introduction

Recently, large-scale end-to-end models (Radford et al., 2023; Gulati et al., 2020; Peng et al., 2022) based on the transformer architecture (Vaswani et al., 2017) have achieved state-of-the-art (SOTA) recognition accuracy in automatic speech recognition (ASR), driving their widespread adoption across various speech processing applications (Zheng et al., 2024; Bhagtani et al., 2024). However, as model size increases, memory usage and computational cost grow substantially, leading to excessive power consumption and inference latency in real-world deployment scenarios (Lee et al., 2024; Wagner et al., 2024). These issues extend beyond efficiency concerns, presenting a critical challenge in terms of environmental impact and degradation of user experience (Aquino-Brítez et al., 2025; Li et al., 2024a). Accordingly, various model compression methods have been actively explored to reduce the size and computational cost of ASR models, while preserving recognition accuracy (Kim and Kim, 2022; Kim et al., 2024; Choi and Kim, 2025). Quantization reduces the bit-precision of weights or activations—originally represented in 32-bit floating-point format—thereby effectively reducing the overall model size (Kim et al., 2022; Gholami et al., 2022). In particular, for large-scale models such as Whisper (Radford et al., 2023), memory bandwidth often emerges as the primary bottleneck rather than computational cost. Consequently, reducing weight bit-precision is crucial for improving practical efficiency (Kim et al., 2023). Moreover, as weights consume over $3.4\times$ more memory than activations, applying lower-bit quantization to weights is more efficient, especially given that activations tend to be more sensitive to accuracy degradation. Accordingly, most ASR quantization studies have primarily focused on weight-only quantization, reflecting this characteristic (Rybakov et al., 2023).

Quantization methods are generally categorized into post-training quantization (PTQ) and quantization-aware training (QAT) (Kim and Kim, 2025; Choi et al., 2024). PTQ applies quantization directly to pre-trained weights without requiring retraining, making it attractive for its simplicity and efficiency. However, the rounding involved in converting floating-point values to lower-precision

*†Corresponding Author



*Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 10704–10718
November 4-9, 2025 

formats can introduce quantization noise, leading to potential performance degradation. To address this, QAT jointly optimizes model weights and quantization parameters (*e.g.,* scaling factors) during training, which allows the model to better adapt to quantization and typically results in higher accuracy (Rybakov et al., 2023; Ding et al., 2022). As a result, most prior ASR quantization studies have focused on QAT. Nevertheless, QAT imposes substantial training overhead, often requiring tens of GPU-hours. For example, applying QAT to the Conformer-large model demands over 84 hours of training (Gulati et al., 2020), which presents a major barrier for practical deployment, especially in user-adaptive or low-resource settings. Furthermore, as model sizes continue to grow, the training cost of QAT is expected to scale accordingly (Guan et al., 2024; Hasan, 2024). These limitations highlight the need for PTQ methods that offer a better trade-off between model size and accuracy, without incurring the high cost of full retraining.

Meanwhile, mixed-precision quantization (MPQ) mitigates accuracy degradation by assigning optimal bit-precision to each layer based on its quantization sensitivity (Ranjan and Savakis, 2025; Dong et al., 2019; Xu et al., 2025). While MPQ has shown strong potential, its application is limited by the prohibitively large search space involved in identifying layer-wise precision configurations. This results in significant computational and optimization overhead, posing a major barrier to practical adoption, especially in large-scale ASR models. Consequently, these challenges have slowed the progress of MPQ research in the ASR domain, despite its theoretical advantages.

To address the limitations of prior ASR quantization methods, we propose *GenPTQ*, a novel mixed-precision PTQ approach that achieves an improved trade-off among accuracy, model size, and optimization efficiency. *GenPTQ* efficiently allocates per-layer bit precision with minimal optimization overhead, achieving robust quantization performance across various ASR models and datasets—all without requiring additional training. The contributions of this work are summarized as follows:

- **Gradient-Based Layer-Wise Sensitivity Measurement:** We quantitatively assess the impact of quantization perturbation on model accuracy by leveraging the gradient information of each layer. This metric guides the allocation of layer-wise optimal bit precision, thereby enabling effective optimization of the trade-off between accuracy and model size.
- **Low-cost Bit-Precision Allocation Method:** We propose a method to minimize the optimization cost associated with bit-precision search. Specifically, we introduce a continuous bit-precision representation (CBPR) based on layer-wise sensitivity values to parameterize bit-precision configurations. By exploiting the differentiability of CBPR, we design a loss function that enforces a target average precision constraint. Additionally, we introduce a sensitivity regularization (SR) to prevent excessive bit-precision reduction in highly sensitive layers. This approach enables the effective derivation of optimal bit-precision configurations with minimal optimization overhead.
- **Evaluation of Generalization and Scalability:** We applied *GenPTQ* to both the Whisper and Conformer models to evaluate its performance. Across various speech datasets, the method maintained an average weight bit-precision of 2.5 bits, reducing model size by 89.1%, with only an average increase of 0.8% in word error rate (WER) and 0.3% in character error rate (CER).

## 2 Related Works

### 2.1 Quantization

**Overview of Quantization:** Quantization reduces the model size by converting weights and activations, originally represented in 32-bit floating-point format, into low-bit precision (*e.g.,* INT2, INT4). In general, min-max uniform quantization is employed, which can be formulated as follows:

$$x_q = clamp(\left\lfloor \frac{x}{s_f} \right\rceil, -2^{b-1}, 2^{b-1}-1) \quad where\ s_f = \frac{\alpha-\beta}{2^b-1} \tag{1}$$

Here, $\alpha$ and $\beta$ denote the maximum and minimum values of a 32-bit floating-point input $x$, respectively. The operator $\lfloor \cdot \rceil$ represents the rounding function, and $b$ indicates the bit-precision. Through this process, $x$ is mapped to a low-bit integer $x_q$ within the range $[-2^{b-1}, 2^{b-1}-1]$. Dequantization is then applied to restore the value to the original floating-point domain, as follows:

$$\tilde{x} = s_f \cdot x_q \tag{2}$$

Quantization inevitably introduces perturbation between the original value $x$ and its quantized representation $\tilde{x}$ due to the rounding operation, which can potentially lead to accuracy degradation. Notably, as bit-precision decreases, model size can be significantly reduced; however, the representational range of the quantized value $x_q$ becomes increasingly limited, leading to greater accuracy degradation. Accordingly, we focus on developing a methodology that effectively improves the trade-off between model size and accuracy.

**Previous Quantization Methods for ASR Models:** To date, most quantization methods for ASR models have been developed within the QAT paradigm. For example, (Ding et al., 2022) applied naive 4-bit QAT to the Conformer, a prominent architecture in ASR, achieving minimal degradation in recognition accuracy. Similarly, (Rybakov et al., 2023) proposed a sub-channel asymmetric, weight-only 2-bit QAT method to further improve recognition accuracy. Furthermore, (Kim et al., 2022) proposed a data-free quantization method that synthesizes mel-spectrograms using batch normalization statistics. Although QAT-based quantization methods have demonstrated high recognition accuracy, their application to large-scale ASR models such as Whisper remains limited due to the substantial training cost, posing challenges for practical deployment. Motivated by the limitations, we propose a PTQ method that minimizes accuracy degradation without requiring additional training, even for large-scale ASR models.

**Previous PTQ Methods for Other Applications:** AdaRound (Nagel et al., 2020) introduced an adaptive rounding method that combines Hessian-based layer-wise task loss approximation with soft relaxation. The method was validated on various convolutional neural networks (CNNs), including ResNet (He et al., 2016) and MobileNetV2 (Sandler et al., 2018), using the ImageNet (Deng et al., 2009) dataset. In addition, BRECQ (Li et al., 2021) proposed a PTQ method based on Hessian-guided block-wise weight reconstruction, targeting various CNNs. Although AdaRound and BRECQ were primarily developed for computer vision tasks, both methods leverage Hessian-based optimization frameworks to precisely control quantization perturbations at the layer or block level. This design plays a critical role in determining the overall performance of PTQ. Nevertheless, these methods typically perform layer- or block-wise optimization and involve complex Hessian-based computations, which lead to substantial optimization costs. Consequently, these methods face practical limitations when directly applied to large-scale ASR models such as Whisper. To emphasize the suitability of the proposed method for ASR tasks, we conduct a comparative evaluation against AdaRound and BRECQ by applying them to ASR models and assessing both efficiency and accuracy.

### 2.2 Mixed Precision Quantization

The primary objective of MPQ is to assign the optimal bit-precision to each layer based on its sensitivity, in order to minimize accuracy degradation under a given target bit-precision constraint. Specifically, higher bit-precisions (*e.g.,* INT6, INT8) are allocated to sensitivity-critical layers to preserve accuracy, while lower bit-precisions (*e.g.,* INT2, INT4) are assigned to less sensitive layers to aggressively reduce model size. This enables effective optimization of the trade-off between accuracy and model size. For example, HTQ (Li et al., 2024b) evaluates the importance of each layer based on connection sensitivity and assigns optimal bit-precision through Pareto frontier exploration. AMPA (Ding et al., 2024) performs sensitivity analysis by integrating gradient magnitude and quantization loss, applying a thresholding mechanism for dynamically bit-precision assignment. However, these methods are specifically tailored for CNNs and face limitations when applied to the structurally more complex transformer-based ASR models. Meanwhile, (Xu et al., 2025) proposed a Gumbel-softmax (Maddison et al., 2016)-based mixed-precision QAT method for wav2vec2. 0(Baevski et al., 2020) and HuBERT-large (Hsu et al., 2021). However, these methods commonly involve complex bit-precision search spaces, resulting in high optimization costs during bit allocation. Such issues become increasingly pronounced for large-scale models, posing a significant barrier to practical deployment in real-world systems. To address this, we propose a mixed-precision PTQ method that efficiently derives an optimal bit-precision configuration with extremely low optimization cost.

## 3 Proposed Methods

### 3.1 Gradient-based Sensitivity Metric

In general, layer-wise sensitivity can be estimated using quantization perturbation based on mean square error (MSE). Specifically, for a layer $L_t$, $where\ t \in \{1, 2, 3, ..., N\}$, with a weight matrix

$w \in \mathbb{R}^{m \times n}$, the MSE-based layer-wise quantization perturbation $\Phi_t$ is defined as follows:

$$\Phi_t = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} (\tilde{w}_{i,j} - w_{i,j})^2 \tag{3}$$

That is, $\Phi_t$ quantitatively measures the perturbation introduced by quantization in layer $L_t$, which guides the assignment of an appropriate bit-precision. However, even with the same level of quantization perturbation, the impact of each layer on the overall accuracy can vary significantly. That is, even when the quantization perturbation is small, certain layers may still have a considerable impact on overall accuracy. This highlights the need for a fine-grained sensitivity metric that can accurately reflect such effects. Therefore, we evaluate the relative importance of each layer to model accuracy by leveraging the mean gradient of the weights with respect to the task loss function $\mathcal{L}_{\text{task}}$, as follows:

$$g_t(w) = \frac{1}{K} \sum_{i=1}^{K} \left| \frac{\partial \mathcal{L}_{task}}{\partial w_i} \right| \tag{4}$$

Here, $g_t(w)$ denotes the average absolute gradient of the weights in layer $L_t$, which consists of $K$ weight elements. The operator $\partial$ represents the partial derivative of the task loss function $\mathcal{L}_{\text{task}}$ with respect to each weight element $w_i$. This metric quantitatively reflects the relative influence of the weights in layer $L_t$ on the task loss $\mathcal{L}_{\text{task}}$ (Chauhan et al., 2023). Based on this gradient $g_t(w)$, we define a more precise sensitivity metric as follows:

$$S_t = \frac{1}{K} \sum_{i=1}^{K} \left( \left| \frac{\partial \mathcal{L}_{task}}{\partial w_i} \right| (\tilde{w}_i - w_i)^2 \right) \tag{5}$$

The proposed sensitivity metric is formulated as the product of the quantization perturbation and $g_t(w)$, enabling it to capture not only the magnitude of quantization error but also the influence of each layer's weights on model accuracy. Specifically, even when the quantization perturbation is small, a high bit-precision can be assigned if the corresponding weights have a significant impact on the task performance. This makes the metric a crucial indicator for guiding the search for bit-precision configurations that achieve optimal accuracy.

### 3.2 Continuous Bit-Precision Representation

The sensitivity $S_t$, derived from Eq. (5), is used to select the optimal bit-precision for each layer from a predefined candidate set $b \in \{\text{INT2}, \text{INT3}, \ldots, \text{INT8}\}$. However, since $b$ is a discrete variable, it is non-differentiable, and the optimal bit-precision is typically determined using search-based algorithms (Kloberdanz and Le, 2023; Tai et al., 2024). To reduce the computational cost associated with such search procedures, we propose CBPR, which transforms layer-wise bit-precision into continuous and differentiable variables, enabling efficient numerical optimization via a loss function. To this end, we normalize $S_t$ and define CBPR within the range bounded by the minimum and maximum bit-precisions, $b_{\min}$ and $b_{\max}$, as follows:

$$P_t = b_{min} + (1 - \frac{S_t - S_{min}}{S_{max} - S_{min}})(b_{max} - b_{min}) \tag{6}$$

Here, $S_{\min}$ and $S_{\max}$ denote the minimum and maximum sensitivity values across all layers, respectively. Accordingly, the initial value of $P_t$ is determined based on the sensitivity distribution, such that layers with higher sensitivity are mapped to higher bit precisions. The computed $P_t$ values are then used to assign bit-precisions to each layer, and the average bit-precision of the model, $b_m$, is defined as follows:

$$b_m = \frac{1}{N} \sum_{t=1}^{N} \lfloor P_t \rceil \tag{7}$$

The compression ratio of a model is commonly evaluated using the average bit-precision $b_m$. However, the initial value of $b_m$ may not satisfy the target bit-precision constraint $b_{tg}$. To address this, we exploit the differentiability of the continuous variable $P_t$ to design an optimizable loss function. This enables the derivation of an optimal bit-precision configuration $\hat{P}_t$ that minimizes performance degradation while adhering to the target bit-precision constraint.

### 3.3 Mixed-Precision Bit Allocation with Sensitivity Regularization

To ensure that the average bit-precision $b_m$ satisfies the target constraint $b_{tg}$, we formulate the following loss function $\mathcal{L}_p$:

$$\mathcal{L}_p = \left| \left( \frac{1}{N} \sum_{t=1}^{N} \lfloor P_t \rceil \right) - b_{tg} \right| \tag{8}$$

By optimizing $P_t$ with respect to $\mathcal{L}_p$, the average bit-precision of the model can be adjusted to converge toward the target constraint $b_{tg}$. However, for

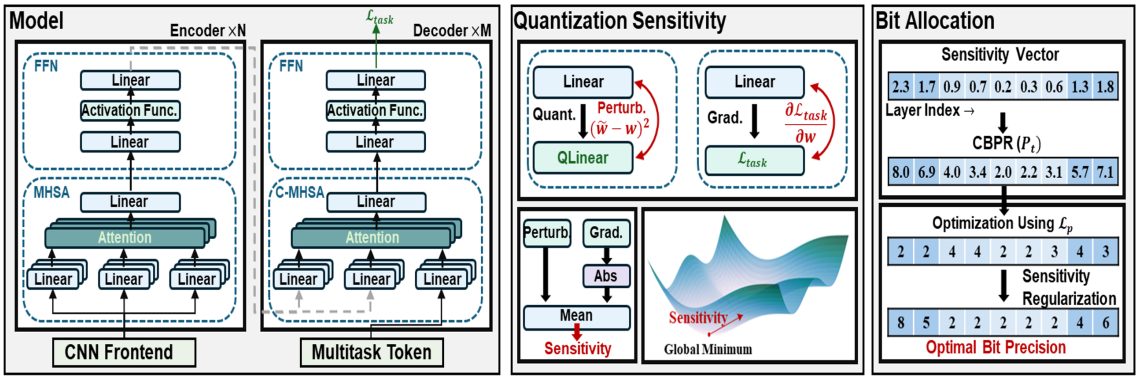


Figure 1: Overview of the Whisper Model Architecture and the Proposed *GenPTQ* Framework.

layers with high sensitivity, the initial value of $P_t$ tends to be large, which may result in excessive reduction during the gradient descent. Consequently, low bit-precision could be assigned to highly sensitive layers, leading to accuracy degradation. To mitigate this, we introduce an SR term as follows:

$$\mathcal{L}_{SR} = \sum_{t=1}^{N} S_t(b_{max} - P_t) \tag{9}$$

The final loss function, including the SR term, is defined as follows:

$$\mathcal{L}_{mpq} = \lambda_1 \mathcal{L}_p + \lambda_2 \mathcal{L}_{SR} \tag{10}$$

The loss function $\mathcal{L}_{mpq}$ mitigates the assignment of excessively low bit-precisions to highly sensitive layers during the optimization of $P_t$, thereby guiding the search toward an optimal layer-wise bit-precision configuration $\hat{P}_t$ that balances the trade-off between accuracy and model size.

### 3.4 The Overall Proposed Framework

Figure 1 presents a visual overview of the overall *GenPTQ* framework, while Algorithm 1 outlines its complete operational procedure. We begin by computing the gradient of the task loss and calculating the layer-wise sensitivity $S_t$ using Eq. (5) (Algorithm1, Line 7). Subsequently, to minimize optimization cost, CBPR is derived from the sensitivity values (Algorithm 1, Line 11), followed by numerical optimization guided by a loss function to obtain the optimal layer-wise bit-precision configuration that satisfies the target bit constraint (Algorithm 1, Lines 13–18). In this process, the final loss function $\mathcal{L}_{\text{mpq}}$, incorporating the SR term, suppresses the assignment of low-precision bits to highly sensitive layers and facilitates the derivation of the optimal bit allocation $\hat{P}t$ that adheres

**Algorithm 1** GenPTQ Framework

1: **Input:** Pre-trained weights $w$, Target bit-precision constraints $b_{tg}$, Initial bit-precision $b$, Scaling factor $s_f$, Optimization iteration $iter$ , Hyper parameter $\lambda_1, \lambda_2$, Total number of layers $N$, Learning rate $\eta$
2: **for** $L_t$ **in** $Layers$ **do**
3: */* Weight quantization & dequantization*/*
4: $w_q = clamp(\left\lfloor \frac{w}{s_f} \right\rceil, -2^b, 2^b - 1)$
5: $\tilde{w} = s_f \cdot w_q$
6: */* Compute gradient-based sensitivity using Eq. (5)*/*
7: $S_t = \frac{1}{K} \sum_{i=1}^{K} \left( \left| \frac{\partial \mathcal{L}_{task}}{\partial w_i} \right| (\tilde{w}_i - w_i)^2 \right)$
8: **end for**
9: **for** $t$ **in** $N$ **do**
10: */* Calculate CBPR using Eq. (6)*/*
11: $P_t = b_{min} + (1 - \frac{S_t - S_{max}}{S_{max} - S_{min}})(b_{max} - b_{min})$
12: **end for**
13: **for** $i$ **in** $iter$ **do**
14: */* Calculate MPQ loss using Eq.(10)*/*
15: $\mathcal{L}_{mpq} = \lambda_1 \mathcal{L}_p + \lambda_2 \mathcal{L}_{SR}$
16: */* Update $P_t$ based on backpropagation*/*
17: $P_t \leftarrow P_t - \eta \cdot \frac{\partial \mathcal{L}_{mpq}}{\partial P_t}$
18: **end for**
19: *# Get optimal bit-precision* $\hat{P}_t$
20: $\hat{P}_t = P_t$
21: **for** $L_t$ **in** $Layers$ **do**
22: */* Assign optimal bit precision per Layer*/*
23: $b_{opt} = \left\lfloor \hat{P}_t \right\rceil$
24: */* Scaling Factor Update/*
25: $s_f = \frac{\alpha - \beta}{2^{b_{opt}} - 1}$
26: */* Weight quantization & dequantization */*
27: $w_q = clamp(\left\lfloor \frac{w}{s_f} \right\rceil, -2^{b_{opt}}, 2^{b_{opt}} - 1)$
28: $\tilde{w} = s_f \cdot w_q$
29: **end for**
30: **Return** $\tilde{w}$

to the target constraint $b_{tg}$. A key advantage of the proposed method is that it requires only a small calibration dataset for gradient computation and achieves efficient bit-precision allocation without relying on combinatorial search, resulting in minimal optimization overhead. This makes it a practical solution for MPQ, effectively reducing optimization cost while achieving a balanced trade-off between accuracy and model size.

## 4 Experimental Results

### 4.1 Experimental Environment

We evaluate the performance of *GenPTQ* on the Whisper-medium and Conformer-large models. The pre-trained weights for Whisper-medium were loaded from Hugging Face (Wolf et al., 2020), while those for Conformer-large were generated by training the model using the ESPnet toolkit (Watanabe et al., 2018). All experiments were conducted on a single H100 GPU. During optimization, the learning rate was fixed at 0.1 with 150 iterations. To simplify the setup, the hyperparameters $\lambda_1$ and $\lambda_2$ in Eq. (10) were both set to 1. Following the configuration in (Rybakov et al., 2023), quantization was implemented at the block-wise granularity.

### 4.2 Performance Evaluation on Various English Speech Datasets

We apply the proposed method to the Whisper-medium model and evaluate its recognition error (*i.e.,* WER and CER) and generalization performance across multiple English speech datasets, including Librispeech test-clean (Panayotov et al., 2015), CommonVoice-test (Ardila et al., 2019), VoxPopuli-test (Wang et al., 2021), Tedlium-test (Rousseau et al., 2012), GigaSpeech-test (Chen et al., 2021), and SPGISpeech-test (O'Neill et al., 2021).

Table 1 compares the performance of the 32-bit floating-point baseline, a standard min-max PTQ method (MinMax), and the proposed *GenPTQ* (Ours). Here, the bit-precision of Ours refers to the average bit-precision across the entire model. As shown in Table 1, our method achieves an average bit-precision of 2.5-bit while maintaining WER and CER increases within 1% across all datasets except CommonVoice. On CommonVoice, relatively higher increases of 2.4% in WER and 1.1% in CER are observed. This can be attributed to the higher level of noise in CommonVoice, which makes the model more susceptible to quantization-induced perturbations compared to cleaner datasets such as Librispeech test-clean. Nevertheless, the proposed method maintains high recognition accuracy on most datasets, demonstrating robust generalization capability across diverse domains. Notably, it achieves a significantly reduced model size of 323.2MB while maintaining WER and CER comparable to MinMax-based INT8 and INT4 PTQ. Furthermore, it consistently outperforms MinMax-based INT2 PTQ across all datasets in terms of recognition accuracy. These results indicate that the proposed method offers a highly effective trade-off between model size and accuracy, making it well-suited for practical deployment in memory-constrained ASR systems.

Additionally, to demonstrate the portability of the proposed method, we evaluate its performance not only on the Whisper model but also on the Wav2Vec2 (Baevski et al., 2020) model. Specifically, we load the pre-trained Wav2Vec2 weights from Hugging Face, apply *GenPTQ*, and evaluate the performance on the LibriSpeech test-clean dataset. Table 2 presents the performance of *GenPTQ* on the Wav2Vec2-L model. Consistent with the results observed on the Whisper model, *GenPTQ* achieves substantially higher speech recognition accuracy at an effective precision of 2.5 bits, compared to the conventional MinMax 2-bit baseline. These findings demonstrate that the proposed method is not only effective for Whisper but also generalizable to diverse speech recognition models.

### 4.3 Performance Evaluation on Conformer

We apply *GenPTQ* to the Conformer-large model and compare it with SOTA QAT methods on the Librispeech test-clean dataset in terms of training cost, accuracy, and model size trade-offs. Since baseline WER values vary across methods, accuracy is compared based on the relative change in WER. Accordingly, in Table 3, Baseline[1] denotes the baseline performance used in prior works, while Baseline[2] represents the baseline performance of our method. Additionally, the training time for QAT methods is measured over 70 epochs, following the official training configuration of the baseline Conformer-large model.

As shown in Table 3, our method incurs only a 0.3% increase in WER when applying 2-bit precision to the Conformer model, representing a marginal increase compared to the SOTA QAT-based method (Rybakov et al., 2023) at the same bit precision. However, it is noteworthy that QAT requires over 84 hours of training, while our method achieves comparable performance with just 7.8 seconds of optimization—focused solely on CBPR tuning rather than weight updates—demonstrating the minimal overhead of our approach. These results demonstrate that our method achieves a favorable trade-off between optimization cost and accuracy. Moreover, compared to prior methods (Kim et al., 2022; Ding et al., 2022), our method in-

| Method | Prec. | Size(MB) | Librispeech | | CommonVoice | | Voxpopuli | | TedLium | | GigaSpeech | | SPGISpeech | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | WER | CER | WER | CER | WER | CER | WER | CER | WER | CER | WER | CER |
| Baseline | FP32 | 2955.9 | 4.1 | 1.6 | 11.4 | 5.5 | 10.2 | 6.0 | 11.9 | 4.5 | 12.6 | 7.5 | 3.7 | 1.8 |
| MinMax | INT8 | 864.5 | 4.2 | 1.6 | 11.5 | 5.5 | 10.2 | 6.1 | 12.0 | 4.8 | 12.8 | 7.6 | 3.7 | 1.8 |
| | INT4 | 489.2 | 4.2 | 1.7 | 11.7 | 5.6 | 10.3 | 6.1 | 12.2 | 5.0 | 12.7 | 7.6 | 3.8 | 1.9 |
| | INT2 | 301.5 | 15.5 | 9.8 | 28.7 | 16.2 | 28.8 | 19.8 | 27.8 | 18.0 | 43.4 | 37.0 | 19.2 | 13.2 |
| **Ours** | **INT2.5** | **323.2** | **4.3** | **1.7** | **13.8** | **6.6** | **11.0** | **6.2** | **12.3** | **4.7** | **12.8** | **7.6** | **4.4** | **2.1** |

Table 1: Performance comparison of quantization methods across various datasets (WER and CER in %).

| Model | Method | Prec. | WER(↓) | CER(↓) |
|---|---|---|---|---|
| Wav2Vec2-L (Baevski et al., 2020) | Baseline | FP32 | 11.9 | 2.9 |
| | MinMax | INT8 | 12.0 | 2.9 |
| | | INT4 | 12.5 | 2.7 |
| | | INT2 | 21.9 | 5.1 |
| | **Ours** | **INT2.5** | **12.6** | **2.7** |

Table 2: Performance of quantization methods on Wav2Vec2-Large using the LibriSpeech dataset

| Method | Prec. | Training | Training Time | Size (MB) | WER(↓) | Diff. |
|---|---|---|---|---|---|---|
| Baseline[1] | FP32 | - | - | 474.5 | 2.0 | - |
| [1] | INT8 | ✓ | 84h 11m | 123.7 | 3.1 | +1.1 |
| [2] | INT8 | | | 138.1 | 2.0 | - |
| [3] | INT4 | | | 81.9 | 2.0 | - |
| [4] | INT2 | | | 55.3 | 2.0 | - |
| Baseline[2] | FP32 | - | - | 475.2 | 2.2 | - |
| **Ours** | **INT2** | - | **7.8s** | **58.6** | **2.5** | **+0.3** |
| | **INT2.5** | | | **62.6** | **2.2** | **-** |

Table 3: Performance comparison between the proposed method and SOTA QAT methods on Conformer-large using the Librispeech dataset

curs only a slight increase in WER under lower bit-precision settings, despite requiring no additional training. Notably, at the 2.5-bit setting, our method achieves optimal accuracy with no increase in WER, matching the performance of SOTA QAT methods. These results highlight that our method offers a superior trade-off among optimization cost, accuracy, and model size compared to existing QAT-based approaches.

Additionally, we compare the performance of different quantization methods for the Conformer model across multiple datasets. In particular, we load pre-trained weights trained on the VoxPopuli, Multilingual Librispeech (MLS) (Pratap et al., 2020) datasets from Hugging Face, and apply *GenPTQ* to evaluate the performance on the corresponding test sets. As shown in Table 4, the MinMax method exhibits a substantial increase in WER at INT2 precision, whereas our approach effectively mitigates this degradation at a comparable precision. Furthermore, on the MLS-German and MLS-Spanish datasets, our method achieves

| Dataset | Method | Prec. | WER(↓) | CER(↓) |
|---|---|---|---|---|
| Voxpopuli | Baseline | FP32 | 6.9 | 3.8 |
| | MinMax | INT8 | 6.9 | 3.9 |
| | | INT4 | 7.3 | 4.1 |
| | | INT2 | 15.3 | 6.8 |
| | **Ours** | **INT2.5** | **7.8** | **4.3** |
| MLS-German | Baseline | FP32 | 4.9 | 1.6 |
| | MinMax | INT8 | 4.9 | 1.6 |
| | | INT4 | 5.0 | 1.7 |
| | | INT2 | 36.5 | 12.6 |
| | **Ours** | **INT2.5** | **5.7** | **1.8** |
| MLS-Spanish | Baseline | FP32 | 4.4 | 1.8 |
| | MinMax | INT8 | 4.5 | 1.8 |
| | | INT4 | 4.8 | 1.9 |
| | | INT2 | 54.2 | 39.3 |
| | **Ours** | **INT2.5** | **5.4** | **2.1** |

Table 4: Performance of quantization methods on Conformer-Large across various datasets

| Method | Prec. | WER(↓) | CER(↓) | Optimization Time |
|---|---|---|---|---|
| Baseline | FP32 | 4.1 | 1.6 | - |
| AdaRound | INT2 | 5.8 | 2.4 | 8h 10m |
| BRECQ | | 5.5 | 2.5 | 2h 41m |
| **Ours** | **INT2.5** | **4.3** | **1.7** | **15.16s** |

Table 5: Performance comparison between the proposed method and SOTA PTQ methods on Whisper-medium using the Librispeech dataset

markedly higher recognition accuracy than the MinMax 2-bit method, demonstrating its superior generalization capability.

### 4.4 Accuracy-optimization cost trade-off Evaluation

We compare SOTA PTQ methods *(e.g.,* AdaRound (Nagel et al., 2020), BRECQ (Li et al., 2021)) with the proposed *GenPTQ* on Whisper-medium model, focusing on the trade-off between optimization cost and accuracy. Table 5 presents the optimization time, WER, and CER of each PTQ method applied to the Whisper-medium model on the Librispeech test-clean dataset. We directly applied AdaRound and BRECQ to Whisper-medium using their official

| Method | Prec. | Size(MB) | MLS-German | | MLS-Portuguese | | MLS-Spanish | | MLS-Polish | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | | WER | CER | WER | CER | WER | CER | WER | CER |
| Baseline | FP32 | 2955.9 | 6.8 | 2.4 | 8.7 | 3.1 | 5.1 | 1.6 | 6.5 | 1.4 |
| MinMax | INT8 | 864.5 | 6.7 | 2.3 | 8.9 | 3.2 | 5.1 | 1.6 | 6.6 | 1.4 |
| | INT4 | 489.2 | 6.9 | 2.4 | 9.2 | 3.6 | 5.2 | 1.7 | 6.8 | 1.6 |
| | INT2 | 301.5 | 51.7 | 39.8 | 64.5 | 38.4 | 30.4 | 16.4 | 69.7 | 54.5 |
| **Ours** | **INT2.5** | **323.2** | **7.9** | **3.1** | **10.3** | **3.6** | **6.3** | **2.0** | **7.9** | **1.9** |

Table 6: Performance comparison of quantization methods across various multilingual datasets

implementations and evaluated performance under consistent experimental conditions. In terms of optimization cost, AdaRound and BRECQ involve computationally expensive layer-wise and block-wise procedures, requiring approximately 8 hours and 2 hours, respectively. In contrast, our method completes optimization in only 15.16 seconds, demonstrating a substantial reduction in optimization cost. From the accuracy perspective, *GenPTQ* also achieves the best performance, achieving a WER of 4.3 under the 2.5-bit precision setting. Meanwhile, directly comparing the proposed method under the 2.5-bit precision setting with existing SOTA PTQ methods under the 2-bit precision setting may appear somewhat unfair. However, as shown in Table 1, the proposed method results in a model size that is only about 22.3MB larger than 2-bit quantization, yet achieves much better recognition performance *(i.e.,* **WER = 4.3 vs. WER = 5.8**) with significantly lower optimization cost *(i.e.,* **15.16s vs. 8h10m**). Considering these factors together, the proposed method offers a highly practical advantage in terms of the trade-off between optimization cost and accuracy, and can be regarded as an efficient PTQ solution for large-scale models.

### 4.5 Ablation Study

**Performance Evaluation on Multilingual Datasets:** Unlike conventional ASR models, Whisper achieves high multilingual recognition accuracy without additional fine-tuning. Accordingly, to assess generalization, we evaluate the proposed method on the MLS dataset, which includes German, Spanish, Portuguese, and Polish. As shown in Table 6, the proposed method achieves significantly lower WER and CER compared to MinMax-based INT2 PTQ. These results demonstrate that the proposed method maintains strong performance even in multilingual scenarios, highlighting its potential for language generalization on Whisper.

| Dataset | SR | WER(↓) | WER Diff. | CER(↓) | CER Diff. |
|---|---|---|---|---|---|
| Librispeech | × | 4.9 | +0.8 | 2.2 | +0.6 |
| | ✓ | **4.3** | **+0.2** | **1.7** | **+0.1** |
| Voxpopuli | × | 11.5 | +1.3 | 6.7 | +0.7 |
| | ✓ | **11.0** | **+0.8** | **6.2** | **+0.2** |
| TedLium | × | 13.0 | +1.1 | 5.2 | +0.7 |
| | ✓ | **12.3** | **+0.4** | **4.7** | **+0.2** |

Table 7: Analysis of the impact of SR on WER and CER for Whisper-medium across Librispeech, Voxpopuli, and TedLium Datasets

**Analysis of the Effectiveness of Sensitivity Regularization:** We further analyze the performance contribution of the proposed SR. Specifically, we assess its impact by comparing results with and without SR in Eq. (10), using the Whisper-medium model across three datasets: LibriSpeech, VoxPopuli, and Tedlium. As shown in Table 7, the application of SR leads to reductions in WER and CER by approximately $0.2 \sim 0.8$ and $0.1 \sim 0.2$, respectively, across all datasets. These results indicate that SR effectively mitigates performance degradation by preventing excessive bit-precision reduction in highly sensitive layers, even under a fixed target bit-precision constraint.

**A Comprehensive Analysis of Sensitivity Regularization:** To more comprehensively analyze the effectiveness of the proposed SR, we conduct a visual analysis of the relationship between layer-wise sensitivity and the assigned bit-precision. Figure 2(a) and (b) illustrate the bit-precision allocation without and with SR, respectively. As shown in Figure 2, the application of SR leads to a clear difference in bit-precision assignment, particularly for highly sensitive layers. In Figure 2(a), where SR is not applied, we observe that layers with high sensitivity *(e.g.,* the first and third layers) are assigned excessively low bit-precisions *(i.e.,*2-bit, 3-bit) to satisfy the global bit constraint $b_{tg}$. This suggests that although higher bit-precisions may be initially assigned during the CBPR computation, they are aggressively reduced during optimization, compromising the quantization expressiveness of sensitive

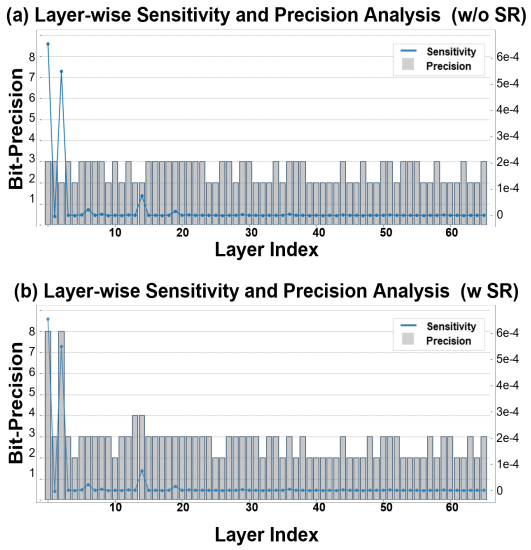


Figure 2: Visualization of bit allocation patterns for Layers 0–70 of the Whisper-medium model under SR-applied and SR-free conditions. The x-axis denotes the layer index, the y-axis indicates the assigned bit-precision, and the secondary axis represents sensitivity.

layers and potentially degrading recognition performance. In contrast, Figure 2(b) shows that the application of SR stabilizes bit allocation, preserving relatively high bit-precision *(i.e.,* 8-bit) for layers with high sensitivity. This highlights the role of SR in preserving bit-precision where it is most needed and in stabilizing bit-precision allocation during CBPR-based optimization.

**Analysis of performance variation with respect to the size of the calibration dataset:** We further provide a detailed analysis of *GenPTQ* performance on the Whisper model with varying sizes of calibration data. As shown in Table 8, our method requires only a very small number of calibration samples (*i.e.*, 1 or 2 samples), which highlights the advantage of *GenPTQ* in incurring extremely low optimization cost. This is because, unlike conventional approaches that utilize calibration datasets to update specific quantization parameters through optimization, the *GenPTQ* method employs them solely for gradient computation to measure sensitivity. As a result, even a single sample is sufficient for effective gradient estimation, rendering *GenPTQ* a highly practical method with very low dependence on calibration data.

## 5 Conclusion

We presented *GenPTQ*, a mixed-precision post-training quantization framework that effectively balances recognition accuracy, model size, and optimization cost for large-scale ASR models. By leveraging a gradient-based sensitivity metric and a CBPR, *GenPTQ* enables efficient layer-wise bit allocation without requiring any retraining. Experimental results on both the Whisper and Conformer models demonstrate that *GenPTQ* achieves an average of 2.5-bit weight quantization while incurring less than one percent degradation in WER. Moreover, it completes the entire optimization process in just 15 seconds, significantly outperforming existing PTQ and QAT methods in terms of both accuracy and efficiency. These findings highlight the practicality and scalability of *GenPTQ* as a quantization solution for real-world ASR deployment.

| Calibration Samples | Dataset | WER(↓) | CER(↓) |
|---|---|---|---|
| 1 | Librispeech | 4.3 | 1.7 |
| 2 | | 4.3 | 1.7 |
| 4 | | OOM | OOM |
| 1 | Voxpopuli | 11.0 | 6.2 |
| 2 | | 11.0 | 6.2 |
| 4 | | OOM | OOM |

Table 8: Performance analysis of *GenPTQ* on the Whisper model with respect to the number of calibration samples

## Limitations

The proposed method demonstrates strong performance across various speech recognition datasets, significantly reducing optimization cost while maintaining high recognition accuracy. However, it follows a gradient-based sensitivity metric to guide bit-precision allocation, which relies on the assumption that the metric accurately reflects the actual impact on recognition accuracy. This assumption may not always hold in models with complex inter-layer dependencies or strong long-range interactions, potentially leading to suboptimal bit allocation.

## Acknowledgement

This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (RS-2025-02304537, Development of Intelligent Home On-Device AI Matter Hub System Technology) and K-CHIPS (Korea Collaborative & High-tech Initiative for Prospective Semiconductor Research) (RS-2025-02305531) funded by the Ministry of Trade, Industry & Energy (MOTIE, Korea).

## References


Sergio Aquino-Brítez, Pablo García-Sánchez, Andrés Ortiz, and Diego Aquino-Brítez. 2025. Towards an energy consumption index for deep learning models: A comparative analysis of architectures, gpus, and measurement tools. *Sensors*, 25(3):846.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M Tyers, and Gregor Weber. 2019. Common voice: A massively-multilingual speech corpus. *arXiv preprint arXiv:1912.06670*.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

Kratika Bhagtani, Amit Kumar Singh Yadav, Paolo Bestagini, and Edward J Delp. 2024. Sslct: A convolutional transformer for synthetic speech localization. In *2024 IEEE 7th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 134–140. IEEE.

Arun Chauhan, Utsav Tiwari, and 1 others. 2023. Post training mixed precision quantization of neural networks using first-order information. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1343–1352.

Guoguo Chen, Shuzhou Chai, Guanbo Wang, Jiayu Du, Wei-Qiang Zhang, Chao Weng, Dan Su, Daniel Povey, Jan Trmal, Junbo Zhang, and 1 others. 2021. Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio. *arXiv preprint arXiv:2106.06909*.

Dahun Choi and Hyun Kim. 2025. Gradq-vit: Robust and efficient gradient quantization for vision transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 16019–16027.

Dahun Choi, Juntae Park, and Hyun Kim. 2024. Hlq: Hardware-friendly logarithmic quantization aware training for power-efficient low-precision cnn models. *IEEE Access*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Li Ding, Wen Fei, Yuyang Huang, Shuangrui Ding, Wenrui Dai, Chenglin Li, Junni Zou, and Hongkai Xiong. 2024. Ampa: Adaptive mixed precision allocation for low-bit integer training. In *Forty-first International Conference on Machine Learning*.

Shaojin Ding, Phoenix Meadowlark, Yanzhang He, Lukasz Lew, Shivani Agrawal, and Oleg Rybakov. 2022. 4-bit conformer with native quantization aware training for speech recognition. *arXiv preprint arXiv:2203.15952*.

Zhen Dong, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. 2019. Hawq: Hessian aware quantization of neural networks with mixed-precision. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 293–302.

Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2022. A survey of quantization methods for efficient neural network inference. In *Low-power computer vision*, pages 291–326. Chapman and Hall/CRC.

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

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and 1 others. 2020. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*.

Jahid Hasan. 2024. Optimizing large language models through quantization: A comparative analysis of ptq and qat techniques. *arXiv preprint arXiv:2411.06084*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM transactions on audio, speech, and language processing*, 29:3451–3460.

Nam Joon Kim and Hyun Kim. 2022. Fp-agl: Filter pruning with adaptive gradient learning for accelerating deep convolutional neural networks. *IEEE Transactions on Multimedia*, 25:5279–5290.

Nam Joon Kim and Hyun Kim. 2025. Hardware-friendly quantization via outlier scaling in convolution-attention-based hybrid networks. In *2025 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–3. IEEE.

Nam Joon Kim, Jongho Lee, and Hyun Kim. 2024. Hyq: hardware-friendly post-training quantization for cnn-transformer hybrid networks. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence*, pages 4291–4299.

Sehoon Kim, Amir Gholami, Zhewei Yao, Nicholas Lee, Patrick Wang, Aniruddha Nrusimha, Bohan Zhai, Tianren Gao, Michael W Mahoney, and Kurt Keutzer. 2022. Integer-only zero-shot quantization for efficient speech recognition. In *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4288–4292. IEEE.

Sehoon Kim, Coleman Hooper, Amir Gholami, Zhen Dong, Xiuyu Li, Sheng Shen, Michael W Mahoney, and Kurt Keutzer. 2023. Squeezellm: Dense-and-sparse quantization. *arXiv preprint arXiv:2306.07629*.

Eliska Kloberdanz and Wei Le. 2023. Mixquant: Mixed precision quantization with a bit-width optimization search. *arXiv preprint arXiv:2309.17341*.

Seung Il Lee, Kwanghyun Koo, Jong Ho Lee, Gilha Lee, Sangbeom Jeong, Seongjun O, and Hyun Kim. 2024. Vision transformer models for mobile/edge devices: a survey. *Multimedia Systems*, 30(2):109.

Haiguang Li, Usama Pervaiz, Joseph Antognini, MichaŁ Matuszak, Lawrence Au, Gilles Roux, and Trausti Thormundsson. 2024a. Enhancing user experience in on-device machine learning with gated compression layers. *arXiv preprint arXiv:2405.01739*.

Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. 2021. Brecq: Pushing the limit of post-training quantization by block reconstruction. *arXiv preprint arXiv:2102.05426*.

Zhikai Li, Xianlei Long, Junrui Xiao, and Qingyi Gu. 2024b. Htq: Exploring the high-dimensional tradeoff of mixed-precision quantization. *Pattern Recognition*, 156:110788.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. 2016. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. 2020. Up or down? adaptive rounding for post-training quantization. In *International conference on machine learning*, pages 7197–7206. PMLR.

Patrick K O'Neill, Vitaly Lavrukhin, Somshubra Majumdar, Vahid Noroozi, Yuekai Zhang, Oleksii Kuchaiev, Jagadeesh Balam, Yuliya Dovzhenko, Keenan Freyberg, Michael D Shulman, and 1 others. 2021. Spgispeech: 5,000 hours of transcribed financial audio for fully formatted end-to-end speech recognition. *arXiv preprint arXiv:2104.02014*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE.

Yifan Peng, Siddharth Dalmia, Ian Lane, and Shinji Watanabe. 2022. Branchformer: Parallel mlp-attention architectures to capture local and global context for speech recognition and understanding. In *International Conference on Machine Learning*, pages 17627–17643. PMLR.

Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert. 2020. Mls: A large-scale multilingual dataset for speech research. *arXiv preprint arXiv:2012.03411*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *International conference on machine learning*, pages 28492–28518. PMLR.

Navin Ranjan and Andreas Savakis. 2025. Mixqvit: Mixed-precision vision transformer quantization driven by layer importance and quantization sensitivity. *arXiv preprint arXiv:2501.06357*.

Anthony Rousseau, Paul Deléglise, and Yannick Esteve. 2012. Ted-lium: an automatic speech recognition dedicated corpus. In *LREC*, pages 125–129.

Oleg Rybakov, Phoenix Meadowlark, Shaojin Ding, David Qiu, Jian Li, David Rim, and Yanzhang He. 2023. 2-bit conformer quantization for automatic speech recognition. *arXiv preprint arXiv:2305.16619*.

Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520.

Yu-Shan Tai and 1 others. 2024. Mptq-vit: Mixed-precision post-training quantization for vision transformer. *arXiv preprint arXiv:2401.14895*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Dominik Wagner, Ilja Baumann, Korbinian Riedhammer, and Tobias Bocklet. 2024. Outlier reduction with gated attention for improved post-training quantization in large sequence-to-sequence speech foundation models. *arXiv preprint arXiv:2406.11022*.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. 2021. Voxpopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. *arXiv preprint arXiv:2101.00390*.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa

Ochiai. 2018. ESPnet: End-to-end speech processing toolkit. In *Proceedings of Interspeech*, pages 2207–2211.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, and 3 others. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Haoning Xu, Zhaoqing Li, Zengrui Jin, Huimeng Wang, Youjun Chen, Guinan Li, Mengzhe Geng, Shujie Hu, Jiajun Deng, and Xunying Liu. 2025. Effective and efficient mixed precision quantization of speech foundation models. *arXiv preprint arXiv:2501.03643*.

Qiuyu Zheng, Zengzhao Chen, Zhifeng Wang, Hai Liu, and Mengting Lin. 2024. Meconformer: Highly representative embedding extractor for speaker verification via incorporating selective convolution into deep speaker encoder. *Expert Systems with Applications*, 244:123004.

# A Appendix

## A.1 Detailed Hyperparameter Setting

Table 9 summarizes the hyperparameter settings used in Algorithm 1. Specifically, the target bit-precision constraint $b_{tg}$ is set to 2.5, and the initial bit-precision $b$ is initialized to 4. Optimization-related hyperparameters (*e.g.,* learning rate, number of iterations) follow the configuration described in Section 4.1.

| Hyperparameter | Value |
|---|---|
| $b_{tg}$ | 2.5 |
| $b$ | 4 |
| $\lambda_1$ | 1 |
| $\lambda_2$ | 1 |
| $\eta$ | 0.1 |
| Iteration | 150 |

Table 9: GenPTQ Hyperparameter Setting

## A.2 Detailed Bit Allocation per Sensitivity in Whisper with *GenPTQ*

In Table 10, we present the bit-precision and corresponding sensitivity values of the fully-connected layer weights in the encoder and decoder blocks (up to the 11th block) of the Whisper-medium model with *GenPTQ* applied. The results are shown under a bit budget constraint of $b_{tq} = 2.5$. Due to the large number of layers, encoder and decoder blocks beyond the 11th are omitted for brevity. Notably, layers exhibiting high sensitivity—such as the first and third layers (*i.e.,* $encoder.layers.0.self\_attn.k\_proj$ and $encoder.layers.0.self\_attn.q\_proj$)— are assigned relatively high bit-precision (*e.g.,*INT8), whereas most of the remaining layers are allocated low bit-precision values (*e.g.,* INT2 or INT3). Given that this setting leads to only minimal increases in WER and CER, the results suggest that aggressively reducing the bit-precision of low-sensitivity layers while preserving high-precision in sensitive layers is an effective strategy for mitigating recognition performance degradation under tight precision constraints.

## A.3 Recognition Example

We present recognition examples obtained by applying both *GenPTQ* and a MinMax-based PTQ method to the Whisper-medium model across various datasets. As shown in Table 11, we report the prediction results on multiple benchmarks, including LibriSpeech, VoxPopuli, Tedlium, CommonVoice, GigaSpeech, MLS-German, MLS-Spanish, and MLS-Polish. For each method, we compare the predicted transcription (Pred) against the ground truth reference (Ref) in terms of WER and CER.

| Encoder Layer | Prec. | Sensitivity | Decoder Layer | Prec. | Sensitivity |
|---|---|---|---|---|---|
| encoder.layers.0.self_attn.k_proj | INT8 | 6.547e-04 | decoder.layers.0.self_attn.k_proj | INT3 | 1.068e-07 |
| encoder.layers.0.self_attn.v_proj | INT3 | 3.891e-06 | decoder.layers.0.self_attn.v_proj | INT2 | 2.012e-08 |
| encoder.layers.0.self_attn.q_proj | INT8 | 5.493e-04 | decoder.layers.0.self_attn.q_proj | INT2 | 1.705e-13 |
| encoder.layers.0.self_attn.out_proj | INT3 | 1.148e-06 | decoder.layers.0.self_attn.out_proj | INT3 | 7.046e-07 |
| encoder.layers.0.fc1 | INT2 | 2.784e-12 | decoder.layers.0.fc1 | INT2 | 8.527e-14 |
| encoder.layers.0.fc2 | INT3 | 1.935e-06 | decoder.layers.0.fc2 | INT2 | 5.698e-08 |
| encoder.layers.1.self_attn.k_proj | INT3 | 2.164e-05 | decoder.layers.1.self_attn.k_proj | INT2 | 1.347e-08 |
| encoder.layers.1.self_attn.v_proj | INT3 | 1.441e-07 | decoder.layers.1.self_attn.v_proj | INT2 | 5.193e-08 |
| encoder.layers.1.self_attn.q_proj | INT3 | 4.118e-06 | decoder.layers.1.self_attn.q_proj | INT3 | 1.685e-07 |
| encoder.layers.1.self_attn.out_proj | INT2 | 2.266e-09 | decoder.layers.1.self_attn.out_proj | INT2 | 3.979e-08 |
| encoder.layers.1.fc1 | INT3 | 1.930e-07 | decoder.layers.1.fc1 | INT3 | 9.779e-07 |
| encoder.layers.1.fc2 | INT2 | 7.034e-13 | decoder.layers.1.fc2 | INT2 | 3.386e-08 |
| encoder.layers.2.self_attn.k_proj | INT3 | 2.087e-06 | decoder.layers.2.self_attn.k_proj | INT3 | 1.600e-06 |
| encoder.layers.2.self_attn.v_proj | INT4 | 6.465e-07 | decoder.layers.2.self_attn.v_proj | INT2 | 3.644e-08 |
| encoder.layers.2.self_attn.q_proj | INT4 | 7.497e-05 | decoder.layers.2.self_attn.q_proj | INT2 | 8.798e-09 |
| encoder.layers.2.self_attn.out_proj | INT3 | 3.103e-07 | decoder.layers.2.self_attn.out_proj | INT2 | 3.553e-15 |
| encoder.layers.2.fc1 | INT3 | 1.842e-07 | decoder.layers.2.fc1 | INT3 | 1.885e-07 |
| encoder.layers.2.fc2 | INT2 | 4.761e-13 | decoder.layers.2.fc2 | INT3 | 9.683e-08 |
| encoder.layers.3.self_attn.k_proj | INT3 | 1.008e-06 | decoder.layers.3.self_attn.k_proj | INT2 | 2.185e-08 |
| encoder.layers.3.self_attn.v_proj | INT3 | 1.586e-07 | decoder.layers.3.self_attn.v_proj | INT2 | 1.221e-09 |
| encoder.layers.3.self_attn.q_proj | INT3 | 3.700e-07 | decoder.layers.3.self_attn.q_proj | INT2 | 2.842e-14 |
| encoder.layers.3.self_attn.out_proj | INT3 | 1.868e-06 | decoder.layers.3.self_attn.out_proj | INT3 | 1.022e-07 |
| encoder.layers.3.fc1 | INT3 | 2.692e-07 | decoder.layers.3.fc1 | INT2 | 0.000e+00 |
| encoder.layers.3.fc2 | INT3 | 6.765e-07 | decoder.layers.3.fc2 | INT3 | 1.332e-07 |
| encoder.layers.4.self_attn.k_proj | INT3 | 6.858e-08 | decoder.layers.4.self_attn.k_proj | INT2 | 4.251e-08 |
| encoder.layers.4.self_attn.v_proj | INT2 | 1.984e-08 | decoder.layers.4.self_attn.v_proj | INT3 | 9.400e-08 |
| encoder.layers.4.self_attn.q_proj | INT2 | 6.253e-13 | decoder.layers.4.self_attn.q_proj | INT3 | 1.735e-07 |
| encoder.layers.4.self_attn.out_proj | INT3 | 1.818e-07 | decoder.layers.4.self_attn.out_proj | INT2 | 9.603e-10 |
| encoder.layers.4.fc1 | INT3 | 6.583e-08 | decoder.layers.4.fc1 | INT3 | 2.469e-07 |
| encoder.layers.4.fc2 | INT3 | 2.810e-06 | decoder.layers.4.fc2 | INT2 | 1.810e-08 |
| encoder.layers.5.self_attn.k_proj | INT3 | 1.684e-07 | decoder.layers.5.self_attn.k_proj | INT3 | 9.476e-06 |
| encoder.layers.5.self_attn.v_proj | INT3 | 1.126e-07 | decoder.layers.5.self_attn.v_proj | INT3 | 1.057e-07 |
| encoder.layers.5.self_attn.q_proj | INT2 | 2.842e-13 | decoder.layers.5.self_attn.q_proj | INT2 | 3.596e-08 |
| encoder.layers.5.self_attn.out_proj | INT3 | 4.630e-07 | decoder.layers.5.self_attn.out_proj | INT2 | 4.974e-14 |
| encoder.layers.5.fc1 | INT3 | 1.493e-07 | decoder.layers.5.fc1 | INT3 | 1.114e-07 |
| encoder.layers.5.fc2 | INT2 | 2.224e-08 | decoder.layers.5.fc2 | INT3 | 1.411e-07 |
| encoder.layers.6.self_attn.k_proj | INT3 | 4.897e-06 | decoder.layers.6.self_attn.k_proj | INT2 | 3.834e-08 |
| encoder.layers.6.self_attn.v_proj | INT2 | 2.309e-08 | decoder.layers.6.self_attn.v_proj | INT2 | 9.836e-09 |
| encoder.layers.6.self_attn.q_proj | INT3 | 2.297e-07 | decoder.layers.6.self_attn.q_proj | INT2 | 8.882e-14 |
| encoder.layers.6.self_attn.out_proj | INT2 | 9.653e-09 | decoder.layers.6.self_attn.out_proj | INT3 | 1.151e-07 |
| encoder.layers.6.fc1 | INT2 | 5.761e-08 | decoder.layers.6.fc1 | INT2 | 9.948e-14 |
| encoder.layers.6.fc2 | INT2 | 2.132e-13 | decoder.layers.6.fc2 | INT2 | 4.219e-08 |
| encoder.layers.7.self_attn.k_proj | INT2 | 2.658e-08 | decoder.layers.7.self_attn.k_proj | INT2 | 1.019e-08 |
| encoder.layers.7.self_attn.v_proj | INT2 | 9.815e-10 | decoder.layers.7.self_attn.v_proj | INT3 | 1.126e-07 |
| encoder.layers.7.self_attn.q_proj | INT3 | 2.658e-06 | decoder.layers.7.self_attn.q_proj | INT2 | 5.779e-09 |
| encoder.layers.7.self_attn.out_proj | INT2 | 5.088e-08 | decoder.layers.7.self_attn.out_proj | INT3 | 1.191e-07 |
| encoder.layers.7.fc1 | INT2 | 2.226e-08 | decoder.layers.7.fc1 | INT3 | 2.236e-07 |
| encoder.layers.7.fc2 | INT2 | 2.700e-13 | decoder.layers.7.fc2 | INT2 | 1.407e-08 |
| encoder.layers.8.self_attn.k_proj | INT3 | 2.861e-07 | decoder.layers.8.self_attn.k_proj | INT3 | 7.417e-06 |
| encoder.layers.8.self_attn.v_proj | INT2 | 3.497e-08 | decoder.layers.8.self_attn.v_proj | INT2 | 5.029e-08 |
| encoder.layers.8.self_attn.q_proj | INT3 | 6.827e-07 | decoder.layers.8.self_attn.q_proj | INT2 | 3.796e-08 |
| encoder.layers.8.self_attn.out_proj | INT3 | 1.654e-06 | decoder.layers.8.self_attn.out_proj | INT2 | 9.592e-14 |
| encoder.layers.8.fc1 | INT3 | 1.026e-07 | decoder.layers.8.fc1 | INT3 | 2.081e-07 |
| encoder.layers.8.fc2 | INT2 | 4.553e-08 | decoder.layers.8.fc2 | INT3 | 4.428e-07 |
| encoder.layers.9.self_attn.k_proj | INT2 | 5.095e-08 | decoder.layers.9.self_attn.k_proj | INT2 | 4.474e-08 |
| encoder.layers.9.self_attn.v_proj | INT2 | 2.995e-08 | decoder.layers.9.self_attn.v_proj | INT2 | 5.124e-08 |
| encoder.layers.9.self_attn.q_proj | INT2 | 2.345e-13 | decoder.layers.9.self_attn.q_proj | INT2 | 9.948e-14 |
| encoder.layers.9.self_attn.out_proj | INT3 | 2.033e-07 | decoder.layers.9.self_attn.out_proj | INT2 | 1.969e-08 |
| encoder.layers.9.fc1 | INT2 | 2.079e-08 | decoder.layers.9.fc1 | INT2 | 1.279e-13 |
| encoder.layers.9.fc2 | INT3 | 1.513e-06 | decoder.layers.9.fc2 | INT2 | 3.613e-08 |
| encoder.layers.10.self_attn.k_proj | INT3 | 9.569e-08 | decoder.layers.10.self_attn.k_proj | INT2 | 1.901e-08 |
| encoder.layers.10.self_attn.v_proj | INT2 | 2.481e-08 | decoder.layers.10.self_attn.v_proj | INT3 | 1.222e-07 |
| encoder.layers.10.self_attn.q_proj | INT2 | 1.634e-13 | decoder.layers.10.self_attn.q_proj | INT3 | 1.156e-06 |
| encoder.layers.10.self_attn.out_proj | INT3 | 9.149e-08 | decoder.layers.10.self_attn.out_proj | INT3 | 1.043e-07 |
| encoder.layers.10.fc1 | INT2 | 2.838e-08 | decoder.layers.10.fc1 | INT2 | 2.810e-09 |
| encoder.layers.10.fc2 | INT3 | 9.042e-07 | decoder.layers.10.fc2 | INT3 | 8.704e-08 |

Table 10: Layer-wise sensitivity and assigned bit-precision in the first 11 encoder/decoder blocks of the Whisper-medium model under the 2.5-bit constraint.

| Dataset | Method | Prec. | | Results | WER | CER |
|---|---|---|---|---|---|---|
| Libirspeech | MinMax | INT4 | Ref : | out in the woods stood a nice little fir tree | 0.0 | 0.0 |
| | | | Pred : | out in the woods stood a nice little fir tree | | |
| | | INT2 | Ref : | out in the woods stood a nice little fir tree | 30.0 | 13.3 |
| | | | Pred : | out in the woods **to the nice** little **fur** tree | | |
| | **GenPTQ** | **INT2.5** | Ref : | out in the woods stood a nice little fir tree | 0.0 | 0.0 |
| | | | Pred : | out in the woods stood a nice little fir tree | | |
| Voxpopuli | MinMax | INT4 | Ref : | i know it means as much to them as it means to me | 0.0 | 0.0 |
| | | | Pred : | i know it means as much to them as it means to me | | |
| | | INT2 | Ref : | i know it means as much to them as it means to me | 30.8 | 30.7 |
| | | | Pred : | it means as much as it means to me | | |
| | **GenPTQ** | **INT2.5** | Ref : | i know it means as much to them as it means to me | 0.0 | 0.0 |
| | | | Pred : | i know it means as much to them as it means to me | | |
| Tedlium | MinMax | INT4 | Ref : | senile decrepit laid up done up done for done in cracked up counted out | 0.0 | 0.0 |
| | | | Pred : | senile decrepit laid up done up done for done in cracked up counted out | | |
| | | INT2 | Ref : | senile decrepit laid up done up done for done in cracked up counted out | 42.9 | 16.9 |
| | | | Pred : | **seen off the crap it** laid up done up done for done in crapped up counted out | | |
| | **GenPTQ** | **INT2.5** | Ref : | senile decrepit laid up done up done for done in cracked up counted out | 0.0 | 0.0 |
| | | | Pred : | senile decrepit laid up done up done for done in cracked up counted out | | |
| CommonVoice | MinMax | INT4 | Ref : | it is a busy market town that serves a large surrounding area | 8.3 | 4.9 |
| | | | Pred : | it is a busy market town that serves a large **surrounded** area | | |
| | | INT2 | Ref : | it is a busy market town that serves a large surrounding area | 91.7 | 63.9 |
| | | | Pred : | **its a piece of** market **and thats around it here yeah** | | |
| | **GenPTQ** | **INT2.5** | Ref : | it is a busy market town that serves a large surrounding area | 8.3 | 4.9 |
| | | | Pred : | it is a busy market town that serves a large **surrounded** area | | |
| GigaSpeech | MinMax | INT4 | Ref : | i mean i showed you a short clip and you may have seen about half of them | 5.9 | 2.7 |
| | | | Pred : | **mean** i showed you a short clip and you may have seen about half of them | | |
| | | INT2 | Ref : | i mean i showed you a short clip and you may have seen about half of them | 11.8 | 13.7 |
| | | | Pred : | i mean i showed you **a clip** and you may have seen about half of **it** | | |
| | **GenPTQ** | **INT2.5** | Ref : | i mean i showed you a short clip and you may have seen about half of them | 0.0 | 0.0 |
| | | | Pred : | i mean i showed you a short clip and you may have seen about half of them | | |

Table 11: Recognition examples generated by the Whisper-medium model quantized with *GenPTQ* across multiple speech datasets.