MLoRQ: Bridging Low-Rank and Quantization for Transformer Compression

Ofir Gordon
ofir.gordon@sony.com
Ariel Lapid
ariel.lapid@sony.com
Elad Cohen
elad.cohen@sony.com
Yarden Yagil
yarden.yagil@sony.com
Arnon Netzer
arnon.netzer@sony.com
Hai Habi
hai.habi@sony.com

Sony Semiconductor Israel
Hod Hasharon, Israel

Abstract

Deploying transformer-based neural networks on resource-constrained edge devices presents a significant challenge. This is commonly addressed using techniques like low-rank approximation and mixed-precision quantization. In this work, we introduce Mixed Low-Rank and Quantization (MLoRQ), a novel method that integrates both techniques. MLoRQ employs a two-stage optimization process to determine optimal bit-width and rank assignments for each layer, adhering to predefined memory constraints. This process includes: (i) an intra-layer optimization that identifies potentially optimal compression solutions out of all low-rank and quantization combinations; (ii) an inter-layer optimization that assigns bit-width precision and rank to each layer while ensuring the memory constraint is met. An optional final step applies a sequential optimization process using a modified adaptive rounding technique to mitigate compression-induced errors in joint low-rank approximation and quantization. The method is compatible and can be seamlessly integrated with most existing quantization algorithms. MLoRQ shows state-of-the-art results with up to 15% performance improvement, evaluated on Vision Transformers for image classification, object detection, and instance segmentation tasks. The code for this work is available at https://github.com/ofirgo/MLoRQ.

1 Introduction

Transformer-based deep neural networks (DNN) [6, 11, 33, 41, 47, 49] have shown state-of-the-art performance in various domains and tasks, including computer vision [4, 11, 33, 47], weather prediction [41], and natural language processing [6]. However, transformer-based DNNs typically experience considerable challenges in terms of computational complexity, power consumption, and latency. This makes their deployment on edge devices with limited memory and computational power a challenging task. The literature presents various approaches to tackling this challenge, including quantization [13, 15, 36], low-rank approximation [19, 20, 21, 43], and pruning [8, 59], among others.

Quantization emerged as a useful technique to reduce model complexity by converting the weight and activation tensors into low-bit-width representations. Specifically, Post-Training Quantization (PTQ), which aims to do so with minor complexity and only a small calibration dataset, demonstrated advantages in compressing vision transformers using single-bit-width quantization [12, 29, 53, 61]. Other studies in convolution neural network (CNN) [10, 17, 25, 26, 39, 48] as well as transformers [34, 42, 56], demonstrate significant advantages by employing mixed precision quantization, where each layer can be assigned with a different bit-width, out of a set of candidates. Another promising technique for compressing transformers is low-rank approximation, which can exploit the prevalence of fully connected operations in transformer-based DNNs. Early approaches [5] used singular value decomposition (SVD) to obtain low-rank alternatives for the original weights. Recent advances use weighted-SVD, guided by the activation covariance matrix [51] or Fisher approximation [19] of the Hessian, to improve performance.

In this work, we propose a compression framework, targeted mainly for transformer-based models, named MLoRQ, which applies joint Mixed Low-Rank and Quantization Optimization. To the best of our knowledge, this is the first attempt at *joint* low-rank and mixed precision quantization optimization, under a PTQ setup. We present a combined representation of these two techniques for a fully-connected layer. Based on this representation, we introduce the mixed rank and precision problem, aiming to determine the smallest possible decrease in performance while adhering to memory constraints. This presents a significant challenge, as the search space is considerably larger than when considering mixed precision or low-rank individually.

We address this challenge by proposing a two-stage algorithm. First, an intra-layer search identifies a Pareto frontier of compression candidates, balancing compression and accuracy while significantly reducing the search space. To further guide the optimization toward better solutions, we leverage Hessian knowledge, using the Label-Free Hessian approach [15]. Then, an inter-layer optimization assigns bit-width and rank to each layer, selecting only from the filtered set of candidates. MLoRQ can be seamlessly integrated with most existing quantization schemes, enhancing them by incorporating both quantization and low-rank compression options. Specifically, here we utilize the activation quantization enhancement from [29, 61] with MLoRQ. To further mitigate compression-induced error, we suggest an optional step of a modified adaptive rounding procedure [36] that considers both quantization and low-rank to refine the weight quantization rounding.

We evaluated MLoRQ on ImageNet [44] and COCO [30] datasets, showing consistent improvements over existing PTQ methods, especially at high compression rates. For instance, MLoRQ improves ViT-B accuracy by up to 15% when reducing the model to under 12.5% of its original size. We also apply MLoRQ to BERT [6] on the GLUE benchmark [50], demonstrating applicability to other domains. Finally, we conducted an ablation study to analyze the contributions of each component in the proposed method.

Our contributions are summarized as follows:

1. We introduce MLoRQ: a Mixed Low-Rank and Quantization approach enabling a notable reduction in the size and computational complexity of transformer-based DNNs.

2. We propose a modified adaptive rounding technique for combined compression that

includes both quantization and low-rank approximation.

3. We present results for computer vision, including ImageNet classification and COCO object detection, where MLoRQ achieves state-of-the-art performance, demonstrating a notable improvement in accuracy over previous methods.

The appendix section of the paper is included in the online Supplementary Material.

2 Related Works

2.1 Post-training Quantization for Transformers

Neural network post-training quantization (PTQ) emerged as one of the most popular approaches for reducing model memory usage and computational complexity. Many works from the last several years explored the approach of applying a short fine-tuning procedure to learn an adaptive rounding policy to improve PTQ performance, mainly for CNNs [15, 28, 36, 52]. While performing well on convolutional neural network (CNN) models, these methods do not achieve comparable performance on transformer architectures due to their unique structure and different bottlenecks. Consequently, recent works have attempted to design dedicated PTQ methods for transformers [7, 12, 16, 32, 34, 53, 60]. The main focus of these works was addressing the difficulties of quantizing the feature maps of Transformers, which tend to present massive outliers, making quantization a challenging task [2, 55]. An important work in this direction, RepQ-ViT [29], introduced a scale-reparameterized quantizer for post-LayerNorm and implemented a Log$\sqrt{2}$ quantizer to handle post-softmax activations quantization that mitigates outlier effects, leading to improved quantization robustness in Transformers. ERQ [61] extended this work and introduced a two-step PTQ approach to mitigate both the activation and the weight quantization error. Another approach, explored in several works, looks for a mixed-precision quantization solution that optimizes the trade-off between model compression and accuracy [10, 42, 56]. These methods selectively assign lower bit widths to layers less sensitive to quantization.

2.2 Low-rank Approximation for Model Compression

Another method for reducing neural networks' memory and computational footprint is low-rank approximation, which decomposes the model's weight matrices into lower-rank representations. Recently, this approach has attracted significant attention for transformer architectures due to their heavy reliance on large matrix multiplications [43, 57, 58]. FWSVD [19] and TFWSVD [20] investigated various weighting strategies to enhance SVD decomposition for ow-rank compression of Transformer-based language models. RankDyna [21] introduced an iterative approach that dynamically assigns ranks based on the importance of the parameters, eliminating the need for a well-trained task-specific model. Other works explored the benefits of low-rank primarily for large language model compression (LLM) and introduced different enhancements for these specific architectures [21, 22, 51]. Previous studies also combined the idea of low-rank and quantization uder the setup of QAT [1, 23, 35].

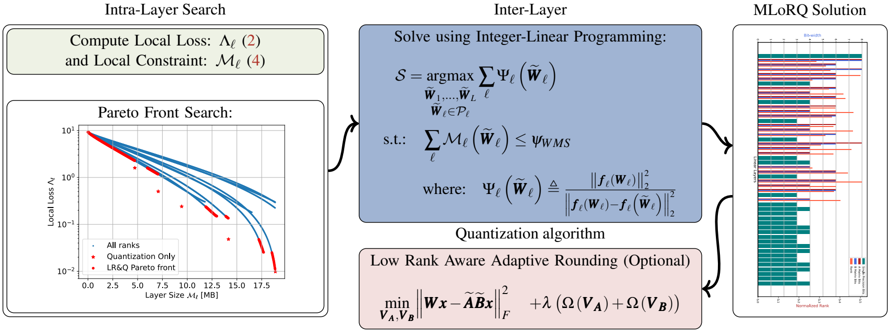


Figure 1: The **MLoRQ** framework for Mixed Low-Rank and Quantization. **(i) Intra-layer** search (5.1) for the Pareto frontier optimal candidates set over the local loss and constraint. **(ii) Inter-layer** search (5.2) to globally assign rank and bit-width to each layer. **(iii)** Apply complementary quantization algorithms with/without **low-rank-aware adaptive rounding**.

3 Background

We provide a short background on low-rank approximation and quantization methods utilized throughout this work. Due to space limitations, a detailed overview of related works is given in Appendix 2.

Low Rank Approximation: Low-rank approximation [45] and tensor decomposition [14, 38] are powerful techniques for reducing neural network size and computational cost. In this study, we focus on compressing transformer-based neural networks, which primarily consist of fully-connected layers and 1×1 convolutions. This structure makes matrix low-rank approximation via singular value decomposition (SVD) a suitable choice for our approach.

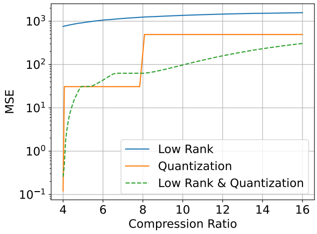
Specifically, let $d_y, d_x \in \mathbb{N}$ and $W \in \mathbb{R}^{d_y \times d_x}$ represent a matrix for which we aim to reduce to rank $r \in \mathbb{N}/\{0\}$; its SVD and low-rank approximation form is given by $W = U\Sigma V^T \approx \widetilde{W} = U^{(r)}\Sigma^{(r)}\left(V^{(r)}\right)^T$, where U, V are unitary matrices, and Σ is a rectangular diagonal matrix containing the non-negative singular values of W. The matrices $U^{(r)}$, $V^{(r)}$, and $\Sigma^{(r)}$ are reduced-rank versions of U, V and Σ, respectively.

It has been demonstrated [19, 51] that this approach does not necessarily yield optimal neural network accuracy. To address this, various methods have introduced additional weighted elements into the SVD. In the remainder of this section, we discuss weighted decomposition with a weighting factor T, where, for a given rank r, we aim to minimize the weighted low rank approximation objective: $\min_{A^{(r)},B^{(r)}} \left\|T\left(W - A^{(r)}B^{(r)}\right)\right\|$, where $A^{(r)} \in \mathbb{R}^{d_x \times r}$ and $B^{(r)} \in \mathbb{R}^{r \times d_y}$ are the low-rank representation of the decomposed matrices of W. A solution to the weighted low-rank approximation objective using SVD is proposed in [19, 45]. Initially, SVD is performed on TW, yielding $U, \Sigma, V = svd(TW)$, where $svd(M)$ represents the singular value decomposition of matrix M. The resulting low-rank factorization is then provided by $A^{(r)} = T^{-1}U^{(r)}\Sigma^{(r)}$ and $B^{(r)} = \left(V^{(r)}\right)^T$.

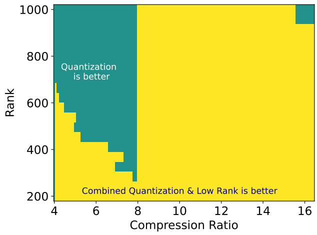
Quantization: Quantization is central to neural network compression, with methods including log-quantization, power-of-two, uniform, symmetric, and more [13]. Let W be a weight tensor to be quantized, with bit-width b, scaling parameter s, and zero point z. The quantized

Problem Statement 1 *Let $\boldsymbol{f}$ be a pre-trained neural network, $\mathcal{S} \triangleq \left(\widetilde{\boldsymbol{W}}_1, \ldots, \widetilde{\boldsymbol{W}}_L\right)$, where $\widetilde{\boldsymbol{W}}_\ell \in \mathcal{W}_\ell$, denotes the compressed weight assignment, $\Delta \boldsymbol{w}$ is the vector representing the difference between the float and compressed weights. Given the constraint of ψ_{WMS} for the weights memory size, we give the following problem formulation:*

$$\min_{\mathcal{S}} \mathcal{L}\left(\boldsymbol{w}+\Delta \boldsymbol{w}(\mathcal{S})\right),$$
$$\text{s.t.}: \text{WeightsMemory}(\mathcal{S}) \leq \psi_{WMS}. \quad (1)$$



(a)



(b)

Figure 2: Problem statement and analysis of different rank and compression rates: (a) Presenting the MSE of a single fully-connected layer under different relative compression ratios; and (b) Shows the best representation for a given rank and compression ratio.

weight is given by $\mathrm{Q}(\boldsymbol{W}, \phi, b) \triangleq s\left(\operatorname{clip}\left(\left\lfloor \frac{\boldsymbol{W}}{s} \right\rceil + z, 0, 2^b - 1\right) - z\right)$, where $\phi = (s, z)$ denote the quantization parameters. Scaling parameters s and zero point z can be chosen using methods like mean square error (MSE) [37], min-max [27], and Hessian-MSE [15]. Recent studies introduced an additional optimization step to minimize rounding errors in weights quantization, specifically by choosing whether to round up or down. The majority of methods employ the soft-quantizer proposed by Nagel et al. [36], detailed in Appendix B.2.

4 Unified Low-Rank and Quantization Representation

We present a joint low-rank and quantization representation. Specifically, let $\boldsymbol{x}_\ell \in \mathbb{R}^{d_\ell^{\boldsymbol{x}}}$, $\boldsymbol{y}_\ell \in \mathbb{R}^{d_\ell^{\boldsymbol{y}}}$ be the input and output vectors of the ℓ^{th} fully-connected layer such that $\boldsymbol{y}_\ell = \boldsymbol{W}_\ell \boldsymbol{x}_\ell + \boldsymbol{b}_\ell$, where $\boldsymbol{W}_\ell \in \mathbb{R}^{d_\ell^{\boldsymbol{y}} \times d_\ell^{\boldsymbol{x}}}$, $\boldsymbol{b}_\ell \in \mathbb{R}^{d_\ell^{\boldsymbol{y}}}$ are the weight matrix and bias vector of the ℓ^{th} layer. Let $\boldsymbol{W}_\ell = \boldsymbol{A}_\ell \boldsymbol{B}_\ell$ be a decomposition of the layer's weights. Given a rank $r_\ell \in \mathcal{R}_\ell \triangleq \left[1, \min\left(d_\ell^{\boldsymbol{x}}, d_\ell^{\boldsymbol{y}}\right)\right]$ and bit-widths $b_\ell^{\boldsymbol{A}}, b_\ell^{\boldsymbol{B}} \in \mathcal{B}$, for the decomposed matrices $\boldsymbol{A}_\ell$ and $\boldsymbol{B}_\ell$, respectively, where $\mathcal{B}$ is the set of bit-width options. We represent the compressed fully-connected operation as [1] $\tilde{\boldsymbol{y}}_\ell = \widetilde{\boldsymbol{W}}_\ell \tilde{\boldsymbol{x}}_\ell + \boldsymbol{b}_\ell$, where $\widetilde{\boldsymbol{W}}_\ell \in \mathcal{W}_\ell$ is the compressed weight. Here, $\mathcal{W}_\ell = \mathcal{W}_\ell^{Q} \cup \mathcal{W}_\ell^{LQ}$ is the set of compressed options of the ℓ^{th} layer where $\mathcal{W}_\ell^{Q} \triangleq \left\{Q\left(\boldsymbol{W}_\ell, \phi_{\boldsymbol{W}_\ell}\left(b_{\boldsymbol{W}_\ell}\right), b_{\boldsymbol{W}_\ell}\right) \mid b_{\boldsymbol{W}_\ell} \in \mathcal{B}\right\}$ and $\mathcal{W}_\ell^{LQ} \triangleq \left\{\widetilde{\boldsymbol{A}}_\ell^{(r)}\left(b_\ell^{\boldsymbol{A}}\right) \widetilde{\boldsymbol{B}}_\ell^{(r)}\left(b_\ell^{\boldsymbol{B}}\right) \mid b_\ell^{\boldsymbol{A}}, b_\ell^{\boldsymbol{B}} \in \mathcal{B}, r_\ell \in \mathcal{R}_\ell\right\}$ are the sets of compressed weights by quantization only and joint quantization and low-rank, respectively. The compressed

[1] $\tilde{\boldsymbol{x}}_\ell$ is the perturbed input of the ℓ^{th} layer as a result of the compression of preceding layers.

matrices $\widetilde{\boldsymbol{A}}_{\ell}^{(r)}\left(b_{\ell}^{\boldsymbol{A}}\right)=Q\left(\boldsymbol{A}_{\ell}^{(r)},\phi_{\boldsymbol{A}_{\ell}}^{(r)}\left(b_{\ell}^{\boldsymbol{A}}\right),b_{\ell}^{\boldsymbol{A}}\right)$ and $\widetilde{\boldsymbol{B}}_{\ell}^{(r)}\left(b_{\ell}^{\boldsymbol{B}}\right)=Q\left(\boldsymbol{B}_{\ell}^{(r)},\phi_{\boldsymbol{B}_{\ell}}^{(r)}\left(b_{\ell}^{\boldsymbol{B}}\right),b_{\ell}^{\boldsymbol{B}}\right)$ are quantized low-rank decomposition matrices, $\phi_{\boldsymbol{A}_{\ell}}^{(r)},\phi_{\boldsymbol{B}_{\ell}}^{(r)}$ and $b_{\ell}^{\boldsymbol{A}},b_{\ell}^{\boldsymbol{B}}$ are the quantization parameters (scale and zero-point) and bit-width of the reduced rank matrices $\boldsymbol{A}_{\ell}^{(r)}$ and $\boldsymbol{B}_{\ell}^{(r)}$, respectively. In cases of a non-fully-connected operation (e.g., 3×3 convolution) $\mathcal{W}_{\ell}=\mathcal{W}_{\ell}^{Q}$.

Considering this joint representation, our goal is to determine the rank and precision of each linear layer in the neural network while given a small representative dataset. This should result in minimal performance degradation while satisfying specified memory constraints. Specifically, let $\mathcal{D}\in\left\{\boldsymbol{x}_{0}^{(i)}\right\}_{i=1}^{N_{\mathcal{D}}}$ be a representative dataset that contains $N_{\mathcal{D}}$ samples, $\boldsymbol{f}$ be a transformer-based neural network with L fully-connected layers that has undergone pre-training with task loss $\mathcal{L}$. We formalize the problem as depicted in Figure 2.

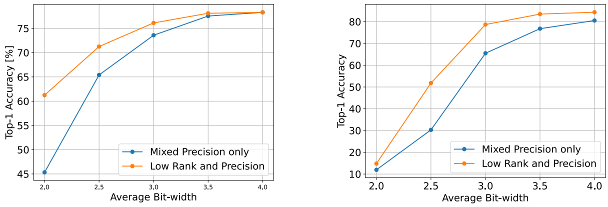
To motivate Problem 1, we analyze the MSE of a fully-connected layer for different ranks and compression ratios, from 25% (8-bit) to 6.25% (2-bit). We evaluate three representations: quantization, low-rank, and the combined approach. Figure 2 (a) shows that none of the representations consistently outperforms the others. Figure 2 (b) shows which representation excels in different ranks and compression rates, highlighting that the combined approach performs best at low ranks and high compression. See details in Appendix A.

5 Mixed Low-Rank and Quantization

We derive the Mixed Low-Rank and Quantization (MLoRQ) method to solve the optimization problem stated in 1. To address the large candidate space (Table 3 in Appendix C), we split the search into two stages: **(i) Intra-layer** filtering using Pareto frontier optimization to find the set of potentially optimal candidates and reduce the search space. **(ii) Inter-layer** search that assigns bit-widths and ranks for a unified compression solution.

After assigning a bit-width and rank to each layer, in the concluding phase, we apply one of two activation quantization methods RepQ [29] or ERQ [61], to correct quantization errors. We wish to highlight that MLoRQ can complement nearly any quantization method. The complete algorithm flow is depicted in Figure 1 and detailed in Algorithm 1 in Appendix B.

To motivate the MLoRQ approach, we investigate the second-order Taylor approximation of the task loss [9, 28]: $\Delta\mathcal{L}=\boldsymbol{g}^{T}\Delta\boldsymbol{w}+\Delta\boldsymbol{w}^{T}\boldsymbol{H}_{\boldsymbol{w}}\Delta\boldsymbol{w}+\mathcal{O}\left(\Delta\boldsymbol{w}^{3}\right)$, where $\Delta\mathcal{L}\triangleq\mathcal{L}(\boldsymbol{w})-\mathcal{L}(\boldsymbol{w}+\Delta\boldsymbol{w})$ is the change in task loss due to compression, $\Delta\boldsymbol{w}$ is the weights perturbation, $\boldsymbol{g}\triangleq\nabla_{\boldsymbol{w}}\mathcal{L}(\boldsymbol{w})$ is the gradient of the task loss w.r.t. the weight vector $\boldsymbol{w}$ and $\boldsymbol{H}_{\boldsymbol{w}}\triangleq\frac{\partial^{2}\mathcal{L}(\boldsymbol{w})}{\partial\boldsymbol{w}\partial\boldsymbol{w}^{T}}$ is the Hessian matrix of the task loss. Under the assumption that $\boldsymbol{g}\approx0$ as in [15, 28, 36], we get the following: $\Delta\mathcal{L}\approx\Delta\boldsymbol{w}^{T}\boldsymbol{H}_{\boldsymbol{w}}\Delta\boldsymbol{w}$. Using the assumption that the layers' contribution to the error compression-induced error is independent as in [10, 36], we get Equation 3. This illustrates the connection between compression-induced weight perturbations and the task loss, thus forming the proposed solution to improve accuracy. According to (3), the problem can be decomposed into distinct components. We begin with the intra-Layer stage, a phase dedicated to minimizing compression error within each layer. Subsequently, in the latter stage, the objective shifts to minimizing the global error, which we refer to as the inter-layer stage. To obtain the Hessian matrix without labels, we utilize the Label-Free Hessian from [15].



$$\Delta\mathcal{L} \approx \overbrace{\sum_{\ell=1}^{L} \underbrace{\Delta \boldsymbol{w}_\ell^T \boldsymbol{H}_{\boldsymbol{w}_\ell} \Delta \boldsymbol{w}_\ell}_{\text{Intra Layer}}}^{\text{Inter Layer}}, \tag{3}$$

Figure 3: Analysis of joint low-rank and quantization: Comparison between joint optimization to mixed precision only on Deit-B and ViT-L (left). The change in task loss due to compression is presented in (3) (right). $\Delta \boldsymbol{W}_\ell = \boldsymbol{W}_\ell - \widetilde{\boldsymbol{W}}_\ell$ is the weight perturbation in matrix form, $\Delta \boldsymbol{w}_\ell = \mathrm{Vec}(\Delta \boldsymbol{W}_\ell)$ is the vectorization of a matrix $\Delta \boldsymbol{W}_\ell$ and $\boldsymbol{H}_{\boldsymbol{w}_\ell}$ are the weights perturbation and Hessian of the ℓ^{th} layer.

5.1 Intra-Layer Search

First, we perform the intra-layer search to determine a set of candidate solutions for the ℓ^{th} layer. This is achieved by minimizing the inner term in (3), under the added assumption that Hessian $\boldsymbol{H}_{\boldsymbol{w}_\ell}$ is a diagonal matrix. This results in the subsequent optimization problem:

$$\min_{\boldsymbol{A}_\ell^{(r)}, \boldsymbol{B}_\ell^{(r)}, \phi_{\boldsymbol{A}_\ell}, \phi_{\boldsymbol{B}_\ell}} \Lambda_\ell\left(\widetilde{\boldsymbol{W}}_\ell\right) \quad \forall b_\ell^{\boldsymbol{A}}, b_\ell^{\boldsymbol{B}}, r_\ell, \tag{2}$$

where $\Lambda_\ell\left(\widetilde{\boldsymbol{W}}_\ell\right) \triangleq \left\|\boldsymbol{D}_\ell \odot \left(\boldsymbol{W}_\ell - \widetilde{\boldsymbol{W}}_\ell\right)\right\|_F^2$ is the local loss, $\boldsymbol{D}_\ell = \sqrt{[\boldsymbol{H}_{\boldsymbol{w}}]_{i+n\cdot j, i+n\cdot j}}$ represents the square root of the Hessian matrix's diagonal component for $\boldsymbol{w}$, and $\boldsymbol{M}_1 \odot \boldsymbol{M}_2$ denotes the Hadamard product (element-wise multiplication) between matrices $\boldsymbol{M}_1$ and $\boldsymbol{M}_2$. We solve problem (2) using a sequential method, starting with the low-rank decomposition, then selecting the quantization parameters, and ending with a search that provides us with all the potentially optimal solutions.

Hessian-aware SVD: We propose a Hessian-aware SVD applied to each layer's weight matrix $\boldsymbol{W}_\ell$. Specifically, we solve problem (2) without quantization for all ranks $r \in \mathcal{R}_\ell$, resulting in the Hessian-aware SVD objective: $\min_{\boldsymbol{A}_\ell^{(r)}, \boldsymbol{B}_\ell^{(r)}} \Lambda_\ell\left(\boldsymbol{A}_\ell^{(r)} \boldsymbol{B}_\ell^{(r)}\right) \quad \forall r \in \mathcal{R}_\ell$. As noted in [45], a closed-form solution to the Hessian-aware SVD objective is generally not available. Thus, we suggest minimizing an upper bound, which is given by $\Lambda_\ell\left(\boldsymbol{A}_\ell^{(r)} \boldsymbol{B}_\ell^{(r)}\right) \leq \left\|\boldsymbol{Q}_\ell\left(\boldsymbol{W}_\ell - \boldsymbol{A}_\ell^{(r)} \boldsymbol{B}_\ell^{(r)}\right)\right\|_F^2$, where $[\boldsymbol{Q}_\ell]_{i,i} = \sum_j [\boldsymbol{D}_\ell]_{i,j}$. We start by applying an SVD on $\boldsymbol{Q}_\ell \boldsymbol{W}_\ell$ to obtain $\boldsymbol{U}_\ell, \boldsymbol{\Sigma}_\ell, \boldsymbol{V}_\ell = svd(\boldsymbol{Q}_\ell \boldsymbol{W}_\ell)$. Next, we combine the results of the Hessian-weighted SVD into two matrices. Given that weights are quantized using a scale per output channel, scaling individual output channels of the matrix does not affect quantization noise because the scaling is incorporated into the quantization scale parameter. Consequently, we select low-rank matrices so that the scaling can integrate into the output channel scale parameter without introducing additional noise, as given by $\boldsymbol{A}_\ell = \boldsymbol{Q}_\ell^{-1} \boldsymbol{U}_\ell$ and $\boldsymbol{B}_\ell = \boldsymbol{\Sigma}_\ell \boldsymbol{V}_\ell^T$.

Quantization Parameters Selection: Given $\boldsymbol{W}_\ell$, $\boldsymbol{A}_\ell$, and $\boldsymbol{B}_\ell$, we aim to find the quantization parameters ϕ (scale and zero-point) that minimize the quantization error for each matrix. For $\boldsymbol{W}_\ell$, the quantization parameters are determined for every bit-width using the Hessian-MSE method proposed in [15], denoted by $\phi_{\boldsymbol{W}_\ell}$, which completes the construction of the set $\mathcal{W}_\ell^Q$.

Next, we address the quantization parameters of the decomposed matrices $\boldsymbol{A}_\ell$ and $\boldsymbol{B}_\ell$ to construct $\mathcal{W}_\ell^{LQ}$. We aim to minimize while considering both the SVD and Hessian factorization to assess their impact on the task loss. We find quantization parameters $\phi_{\boldsymbol{A}_\ell}$ and $\phi_{\boldsymbol{B}_\ell}$ by minimizing $\min_{\phi_{\boldsymbol{A}_\ell},\phi_{\boldsymbol{B}_\ell}} \Lambda_\ell\left(\hat{\boldsymbol{W}}_\ell\right) \quad \forall b_\ell^{\boldsymbol{A}}, b_\ell^{\boldsymbol{B}} \in \mathcal{B}$, where $\hat{\boldsymbol{W}}_\ell = Q\left(\boldsymbol{A}_\ell, \phi_{\boldsymbol{A}_\ell}, b_\ell^{\boldsymbol{A}}\right) Q\left(\boldsymbol{B}_\ell, \phi_{\boldsymbol{B}_\ell}, b_\ell^{\boldsymbol{B}}\right)$ is the quantized SVD without rank reduction. This means that we determine the quantization parameters once for all ranks. Given the large number of candidates, we propose solving the quantization parameters optimization separately for $\boldsymbol{A}_\ell$ and $\boldsymbol{B}_\ell$. For $\boldsymbol{A}_\ell$, we solve the following problem: $\phi_{\boldsymbol{A}_\ell}(b) = \arg\min_\phi \Lambda_\ell\left(Q\left(\boldsymbol{A}_\ell, \phi, b\right)\boldsymbol{B}_\ell\right)$. As for $\boldsymbol{B}_\ell$, we reduce the search complexity using the bound $\Lambda_\ell\left(\boldsymbol{A}_\ell Q\left(\boldsymbol{B}_\ell, \phi, b\right)\right) \leq \left\|\boldsymbol{B}_\ell - Q\left(\boldsymbol{B}_\ell, \phi, b\right)\right\|_F^2$. This results in the following minimization problem: $\phi_{\boldsymbol{B}_\ell}(b) = \arg\min_\phi \left\|\boldsymbol{B}_\ell - Q\left(\boldsymbol{B}_\ell, \phi, b\right)\right\|_F^2$. The minimization in $\phi_{\boldsymbol{A}_\ell}(b)$ and $\phi_{\boldsymbol{B}_\ell}(b)$ is performed on a fixed grid of points using a line search.

Pareto Set Search: In the final step of this phase, we extract only relevant potentially optimal solutions for the ℓ^{th} layer by identifying the set of Pareto optimal solutions. Specifically, let $\Lambda_\ell\left(\widetilde{\boldsymbol{W}}_\ell\right)$ denote the loss (2), and $\mathcal{M}_\ell\left(\widetilde{\boldsymbol{W}}_\ell\right)$ represents the memory footprint of the compressed weight, given by

$$\mathcal{M}_\ell\left(\widetilde{\boldsymbol{W}}_\ell\right) \triangleq \begin{cases} r_\ell\left(d_\ell^{\boldsymbol{y}} \cdot b_\ell^{\boldsymbol{A}} + d_\ell^{\boldsymbol{x}} \cdot b_\ell^{\boldsymbol{B}}\right) & \widetilde{\boldsymbol{W}}_\ell \in \mathcal{W}_\ell^{LQ} \\ d_\ell^{\boldsymbol{y}} \cdot d_\ell^{\boldsymbol{x}} \cdot b_{\boldsymbol{W}_\ell} & \widetilde{\boldsymbol{W}}_\ell \in \mathcal{W}_\ell^{Q}. \end{cases} \tag{4}$$

Let us define that a compression option $\widetilde{\boldsymbol{W}}_\ell'$ *dominates* ($\succ$) another option $\widetilde{\boldsymbol{W}}_\ell$ if both its local error and memory footprint are better and at least one of them is strictly better. Then, a compression option is *Pareto optimal* if no other compression option dominates it. Formally, the set of Pareto optimal solutions is denoted by $\mathcal{P}_\ell = \left\{\widetilde{\boldsymbol{W}}_\ell \in \mathcal{W}_\ell | \nexists \widetilde{\boldsymbol{W}}_\ell' \in \mathcal{W}_\ell : \widetilde{\boldsymbol{W}}_\ell' \succ \widetilde{\boldsymbol{W}}_\ell\right\}$. After obtaining Λ_ℓ and $\mathcal{M}_\ell$ for all ranks and bit-widths, we select a set of candidate solutions on the Pareto frontier as shown in the Pareto search image in Figure 1, and in Appendix F.

5.2 Inter-Layer Search

After acquiring the compression candidates $\mathcal{P}_\ell$ of each layer, We search for a mixed rank and quantization solution $\mathcal{S}$ that assigns an optimal compression setup to each layer, minimizing the total error in the compressed model. While Hessian-based metrics are common in global optimization [9, 10, 59], recent studies suggest instead to maximize the model's signal-to-quantization noise ratio (SQNR) [24, 39]. Our ablation study in Appendix E.4.2 supports this approach. We therefore formulate the problem as follows:

$$\mathcal{S} = \underset{\widetilde{\boldsymbol{W}}_1,\ldots,\widetilde{\boldsymbol{W}}_L}{\operatorname{argmax}} \sum_\ell \Psi_\ell\left(\widetilde{\boldsymbol{W}}_\ell\right) \quad \text{s.t.:} \quad \sum_\ell \mathcal{M}_\ell\left(\widetilde{\boldsymbol{W}}_\ell\right) \leq \psi_{WMS}, \quad \text{and} \quad \widetilde{\boldsymbol{W}}_\ell \in \mathcal{P}_\ell \quad \forall \ell, \tag{5}$$

where $\Psi_\ell\left(\widetilde{\boldsymbol{W}}_\ell\right) \triangleq \frac{\left\|\boldsymbol{f}_\ell(\boldsymbol{W}_\ell)\right\|_2^2}{\left\|\boldsymbol{f}_\ell(\boldsymbol{W}_\ell) - \boldsymbol{f}_\ell\left(\widetilde{\boldsymbol{W}}_\ell\right)\right\|_2^2}$ is the SQNR of the ℓ^{th} layer perturbation, $\boldsymbol{f}_\ell$ is the output of a neural network as a function of the ℓ^{th} weight tensor, while the rest of the model is in floating point. For the cases where $b_{\boldsymbol{A}_\ell}$ and $b_{\boldsymbol{B}_\ell}$ have a large number of candidates for different ranks, we apply a linear interpolation instead of running a full model inference, to reduce the computational cost (Appendix D). We translate the problem in (5) into an Integer Linear Program and use an off-the-shelf [40] algorithm, detailed in the Appendix B.1.

Table 1: Top-1 accuracy (%) on ImageNet classification. **"+ RepQ"** and **"+ ERQ"** refer to combining MLoRQ with RepQ's activation parametrization and ERQ's activation error reduction techniques, respectively. All referenced results were taken from [61].

Model	Model Size (%)	ViT-S	ViT-B	DeiT-T	DeiT-S	DeiT-B	Swin-S	Swin-B
Full Prec.	100%	**81.39**	**84.54**	**72.21**	**79.85**	**81.8**	**83.23**	**85.27**
GPTQ [12]		23.32	44.63	42.25	48.95	61.75	66.71	71.43
PD-Quant [31]		4.56	21.81	41.87	41.65	53.63	70.07	56.48
RepQ-ViT [29]		15.65	26.98	29.34	45.82	58.92	59.83	44.17
ERQ [61]	9.3%	45.68	53.88	44.09	57.63	70.33	75.08	75.78
MLoRQ + RepQ (Ours)	(~3-bit)	17.65	48.16	44.38	52.32	68.41	77.90	77.01
MLoRQ + ERQ (Ours)		42.64	60.16	46.74	58.32	71.61	78.02	78.90
MLoRQ + ERQ + LoRAda (Ours)		**46.55**	**69.56**	**52.17**	**64.44**	**76.12**	**79.27**	**79.06**
GPTQ [12]		67.59	75.12	58.96	70.85	76.10	80.17	81.08
PD-Quant [31]		32.64	34.86	58.50	64.85	60.06	77.04	75.84
RepQ-ViT [29]		65.05	68.48	57.43	69.03	75.61	79.45	78.32
ERQ [61]	12.5%	68.91	76.63	60.29	72.56	78.23	80.74	82.44
MLoRQ + RepQ (Ours)	(~4-bit)	64.61	75.98	59.11	70.44	76.78	80.15	81.96
MLoRQ + ERQ (Ours)		68.11	78.43	60.26	72.18	77.99	80.61	82.61
MLoRQ + ERQ + LoRAda (Ours)		**69.12**	**79.13**	**61.55**	**72.80**	**78.61**	**80.76**	**82.63**

Table 2: Results on COCO dataset. "AP^{box}" denotes the box average precision for object detection, and "AP^{mask}" denotes the mask average precision for instance segmentation. All referenced results were taken from [62].

Method	Model Size (%)	Mask R-CNN				Cascade Mask R-CNN			
		w. Swin-T		w. Swin-S		w. Swin-T		w. Swin-S	
		AP^{box}	AP^{mask}	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}	AP^{box}	AP^{mask}
Full-Precision	**100%**	**46.0**	**41.6**	**48.5**	**43.3**	**50.4**	**43.7**	**51.9**	**45.0**
GPTQ [12]		22.9	25.0	31.7	32.5	39.8	35.8	44.6	39.6
RepQ-ViT [29]	9.3%	22.2	24.1	27.9	29.9	40.2	35.7	43.7	38.8
BRECQ [28]	(~3-bit)	14.7	18.2	15.3	19.5	29.9	28.0	32.3	29.3
ERQ [62]		27.2	28.9	30.6	33.0	45.1	40.0	47.3	42.0
MLoRQ + ERQ + LoRAda (Ours)		**33.7**	**32.9**	**36.8**	**35.5**	**46.1**	**40.8**	**48.1**	**42.4**

6 Low-Rank Aware Adaptive Rounding

Given the structure of compressed weights, as a result of the combination of low-rank and quantization, we propose a modified version of Adaptive Rounding [36], referred to as **LoRAda**, tailored for scenarios involving low-rank and quantization considerations. The procedure encompasses a sequence of optimizations in which, for each layer, we aim to minimize both the error caused by weight perturbations and the propagation errors throughout the model. Suppose $\boldsymbol{A} \in \mathbb{R}^{d_y \times r}$ and $\boldsymbol{B} \in \mathbb{R}^{r \times d_x}$ represent the low-rank decomposition of a certain layer whose original weight is denoted by $\boldsymbol{W} \in \mathbb{R}^{d_y \times d_x}$. We define two continuous variables, $\boldsymbol{V_A} \in \mathbb{R}^{d_y \times r}$ and $\boldsymbol{V_B} \in \mathbb{R}^{r \times d_x}$, which are optimized as part of the sequence-mean mechanism given by $\min_{\boldsymbol{V_A},\boldsymbol{V_B}} \left\|\boldsymbol{Wx} - \widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{B}}\boldsymbol{x}\right\|_F^2 + \lambda\left(\Omega\left(\boldsymbol{V_A}\right) + \Omega\left(\boldsymbol{V_B}\right)\right)$, where $\widetilde{\boldsymbol{A}} = Q_S\left(\boldsymbol{A}, \boldsymbol{V_A}, \phi_{\boldsymbol{A}}, b_{\boldsymbol{A}}\right)$ and $\widetilde{\boldsymbol{B}} = Q_S\left(\boldsymbol{B}, \boldsymbol{V_B}, \phi_{\boldsymbol{B}}, b_{\boldsymbol{B}}\right)$ are the soft-quantized (B.2) low-rank decomposition matrices.

7 Experiments

We conducted experiments to demonstrate the advantages of MLoRQ. We first evaluated MLoRQ on the ImageNet dataset [44] using Vision Transformers (taken from timm [54]), and on COCO with Mask R-CNN [18] and Cascade Mask R-CNN [3], comparing it to leading compression methods. We conducted an ablation study to support the choice for unifying quantization and low-rank. Additional ablation experiments investigating different aspects of MLoRQ are provided in Appendix E.4. In addition, results on BERT [6] using the GLUE benchmark [50] appear in Appendix E.3 due to space limitations.

In our experiments, we applied the same setup unless otherwise specified. To ensure a fair comparison, we adopted quantization settings from prior work [29, 61], using a uniform quantizer for weight and activation and a $log2$ quantizer for the softmax function. A set of 1,024 images was used for the optimization process. We report results in terms of the compressed weights memory size relative to the floating-point model weights memory (%), where the compressed size accounts for both low-rank and quantization, as defined in (4). Activations are quantized to uniform 4-bit in all experiments unless stated otherwise.

ImageNet Classification The results for the ViT-based classification task are presented in Table 1. In this experiment, the model's weights are compressed using our MLoRQ method. We incorporate two activation quantization enhancements into MLoRQ: RepQ's reparameterization [29] (**MLoRQ + RepQ**) and ERQ's activation error reduction [61] (**MLoRQ + ERQ**), highlighting MLoRQ's standalone effectiveness, independent of dedicated rounding refinement. Finally, we report results of our complete method, including our low-rank-aware adaptive rounding (**+ LoRAda**).

As discussed in this paper, MLoRQ is highly effective for compressing vision transformers due to their architecture, which leverages the advantages of low-rank compression. As shown in Table 1, MLoRQ achieves state-of-the-art results across architectures, with up to 15% improvement, especially in low-bit settings. Full comparisons with additional methods and activation mixed-precision results appear in Appendices E.1 and E.2, respectively.

COCO Object Detection We evaluated MLoRQ on the COCO dataset for object-detection and instance segmentation on the Mask-RCNN [18] and Cascade Mask R-CNN [3] models. The results are presented in Table 2. MLoRQ shows improvement over state-of-the-art methods in different settings with over 6 percentage points in “AP bbox” for Mask R-CNN.

Joint Low-Rank and Quantization Ablation We evaluate the contribution of incorporating low-rank approximation with mixed precision. We compare the results of MLoRQ to the same algorithmic flow using only mixed precision options at various weight compression rates. Figure 3 (left) presents this comparison for the DeiT-B model and the larger ViT-L model, with results for additional models in Appendix E.4.1. The results highlight the impact of integrating low-rank options into the optimization, particularly at high compression rates, where accuracy improvements are substantial.

8 Conclusions

In this study, we tackle joint low-rank approximation and mixed precision quantization by introducing MLoRQ, a novel algorithm that assigns rank and bit-width per layer. MLoRQ comprises two stages: (i) intra-layer candidates reduction via Pareto optimization; (ii) inter-layer rank and bit-width allocation. The process is followed by an adaptive rounding refinement for mixed representations. It integrates easily into existing quantization schemes, unifying low-rank and quantization strategies. Experiments in vision tasks show notable improvement over state-of-the-art. This work highlights the potential of combining quantization and low-rank approximation across tasks, with future directions including: (i) extending MLoRQ to convolutional neural networks (CNNs), large language models (LLMs), and vision-language models (VLMs) with high memory demands, and (ii) exploring additional compression techniques like pruning and one-bit quantization.

Acknowledgments

We would like to thank Idit Diamant, Idan Ben-Ami, Reuven Peretz, Lior Dikshtein, Elad Cohen, Irena Byzalov, Itai Berman, and Nir Elkayam for their helpful insights and comments, which contributed significantly to the development of this paper.

References

[1] Yelysei Bondarenko, Riccardo Del Chiaro, and Markus Nagel. Low-rank quantization-aware training for llms. *arXiv preprint arXiv:2406.06385*, 2024.

[2] Yelysei Bondarenko, Markus Nagel, and Tijmen Blankevoort. Quantizable transformers: Removing outliers by helping attention heads do nothing. *NeurIPS*, 36, 2024.

[3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, pages 6154–6162, 2018.

[4] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *ECCV*, pages 213–229. Springer, 2020.

[5] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. *NeurIPS*, 27, 2014.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pages 4171–4186. Association for Computational Linguistics, 2019.

[7] Yifu Ding, Haotong Qin, Qinghua Yan, Zhenhua Chai, Junjie Liu, Xiaolin Wei, and Xianglong Liu. Towards accurate post-training quantization for vision transformer. In *ACM MM*, pages 5380–5388, 2022.

[8] Xin Dong, Shangyu Chen, and Sinno Pan. Learning to prune deep neural networks via layer-wise optimal brain surgeon. *NeurIPS*, 30, 2017.

[9] Zhen Dong, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. HAWQ: Hessian aware quantization of neural networks with mixed-precision. In *ICCV*, pages 293–302, 2019.

[10] Zhen Dong, Zhewei Yao, Daiyaan Arfeen, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Hawq-v2: Hessian aware trace-weighted quantization of neural networks. In *NeurIPS*, pages 18518–18529, 2020.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[12] Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. Optq: Accurate quantization for generative pre-trained transformers. In *ICLR*, 2022.

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

[14] Gene H Golub and Christian Reinsch. Singular value decomposition and least squares solutions. In *Handbook for automatic computation: volume II: linear algebra*, pages 134–151. Springer, 1971.

[15] Ofir Gordon, Elad Cohen, Hai Victor Habi, and Arnon Netzer. Eptq: Enhanced post-training quantization via hessian-guided network-wise optimization. *arXiv preprint arXiv:2309.11531*, 2024.

[16] Cong Guo, Jiaming Tang, Weiming Hu, Jingwen Leng, Chen Zhang, Fan Yang, Yunxin Liu, Minyi Guo, and Yuhao Zhu. Olive: Accelerating large language models via hardware-friendly outlier-victim pair quantization. In *Proceedings of the 50th Annual International Symposium on Computer Architecture*, pages 1–15, 2023.

[17] Hai Victor Habi, Roy H Jennings, and Arnon Netzer. Hmq: Hardware friendly mixed precision quantization block for cnns. In *ECCV*, pages 448–463, 2020.

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[19] Yen-Chang Hsu, Ting Hua, Sungen Chang, Qian Lou, Yilin Shen, and Hongxia Jin. Language model compression with weighted low-rank factorization. In *ICLR*, 2022.

[20] Ting Hua, Yen-Chang Hsu, Felicity Wang, Qian Lou, Yilin Shen, and Hongxia Jin. Numerical optimizations for weighted low-rank estimation on language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1404–1416, 2022".

[21] Ting Hua, Xiao Li, Shangqian Gao, Yen-Chang Hsu, Yilin Shen, and Hongxia Jin. Dynamic low-rank estimation for transformer-based language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[22] Wei Huang, Haotong Qin, Yangdong Liu, Yawei Li, Qinshuo Liu, Xianglong Liu, Luca Benini, Michele Magno, Shiming Zhang, and XIAOJUAN QI. Slim-LLM: Salience-driven mixed-precision quantization for large language models. In *ICML*, 2025.

[23] Wenjin Ke, Zhe Li, Dong Li, Lu Tian, and Emad Barsoum. Dl-qat: Weight-decomposed low-rank quantization-aware training for large language models. *arXiv preprint arXiv:2504.09223*, 2025.

[24] Hana Kim, Hyun Eun, Jung Hwan Choi, and Ji-Hoon Kim. Complexity-aware layer-wise mixed-precision schemes with sqnr-based fast analysis. *IEEE Access*, 11:117800–117809, 2023.

[25] Ivan Koryakovskiy, Alexandra Yakovleva, Valentin Buchnev, Temur Isaev, and Gleb Odinokikh. One-shot model for mixed-precision quantization. In *CVPR*, pages 7939–7949, 2023.

[26] Ivan Koryakovskiy, Alexandra Yakovleva, Valentin Buchnev, Temur Isaev, and Gleb Odinokikh. One-shot model for mixed-precision quantization. In *CVPR*, pages 7939–7949, 2023.

[27] Raghuraman Krishnamoorthi. Quantizing deep convolutional networks for efficient inference: A whitepaper. *arXiv preprint arXiv:1806.08342*, 2018.

[28] Yuhang Li, Ruihao Gong, Xu Tan, Yang Yang, Peng Hu, Qi Zhang, Fengwei Yu, Wei Wang, and Shi Gu. BRECQ: Pushing the limit of post-training quantization by block reconstruction. In *ICLR*, 2021.

[29] Zhikai Li, Junrui Xiao, Lianwei Yang, and Qingyi Gu. Repq-vit: Scale reparameterization for post-training quantization of vision transformers. In *ICCV*, pages 17227–17236, 2023.

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.

[31] Jiawei Liu, Lin Niu, Zhihang Yuan, Dawei Yang, Xinggang Wang, and Wenyu Liu. Pd-quant: Post-training quantization based on prediction difference metric. In *CVPR*, pages 24427–24437, 2023.

[32] Yijiang Liu, Huanrui Yang, Zhen Dong, Kurt Keutzer, Li Du, and Shanghang Zhang. Noisyquant: Noisy bias-enhanced post-training activation quantization for vision transformers. In *CVPR*, pages 20321–20330, 2023.

[33] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021.

[34] Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. *NeurIPS*, 34:28092–28103, 2021.

[35] Sebastian Loeschcke, Mads Toftrup, Michael Kastoryano, Serge Belongie, and Vésteinn Snæbjarnarson. Loqt: Low-rank adapters for quantized pretraining. *NeurIPS*, pages 115282–115308, 2024.

[36] Markus Nagel, Rana Ali Amjad, Mart Van Baalen, Christos Louizos, and Tijmen Blankevoort. Up or down? adaptive rounding for post-training quantization. In *ICML*, pages 7197–7206. PMLR, 2020.

[37] Yury Nahshan, Brian Chmiel, Chaim Baskin, Evgenii Zheltonozhskii, Ron Banner, Alex M Bronstein, and Avi Mendelson. Loss aware post-training quantization. *Machine Learning*, 110(11):3245–3262, 2021.

[38] Matan Ben Noach and Yoav Goldberg. Compressing pre-trained language models by matrix decomposition. In *AACL-IJCNLP*, pages 884–889, 2020.

[39] Nilesh Prasad Pandey, Markus Nagel, Mart van Baalen, Yin Huang, Chirag Patel, and Tijmen Blankevoort. A practical mixed precision algorithm for post-training quantization. *arXiv preprint arXiv:2302.05397*, 2023.

[40] Laurent Perron and Vincent Furnon. OR-Tools, 2024. URL https://developers.google.com/optimization/. Version v9.11. by Google, Available at: https://developers.google.com/optimization/.

[41] Ilan Price, Alvaro Sanchez-Gonzalez, Ferran Alet, Tom R Andersson, Andrew El-Kadi, Dominic Masters, Timo Ewalds, Jacklynn Stott, Shakir Mohamed, Peter Battaglia, et al. Probabilistic weather forecasting with machine learning. *Nature*, 637(8044): 84–90, 2025.

[42] Navin Ranjan and Andreas Savakis. Lrp-qvit: Mixed-precision vision transformer quantization via layer-wise relevance propagation. *arXiv preprint arXiv:2401.11243*, 2024.

[43] Siyu Ren and Kenny Q. Zhu. Low-rank prune-and-factorize for language model compression. In *LREC-COLING*, pages 10822–10832, 2024.

[44] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252, 2015.

[45] Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *ICML*, pages 720–727, 2003.

[46] Yu-Shan Tai and An-Yeu Wu. Amp-vit: Optimizing vision transformer efficiency with adaptive mixed-precision post-training quantization. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pages 6828–6837, 2025.

[47] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357. PMLR, 2021.

[48] Stefan Uhlich, Lukas Mauch, Fabien Cardinaux, Kazuki Yoshiyama, Javier Alonso Garcia, Stephen Tiedemann, Thomas Kemp, and Akira Nakamura. Mixed precision dnns: All you need is a good parametrization. In *ICLR*, 2020.

[49] A Vaswani. Attention is all you need. *NeurIPS*, 2017.

[50] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, 2018.

[51] Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. SVD-LLM: Truncation-aware singular value decomposition for large language model compression. In *ICLR*, 2025.

[52] Xiuying Wei, Ruihao Gong, Yuhang Li, Xianglong Liu, and Fengwei Yu. QDrop: Randomly dropping quantization for extremely low-bit post-training quantization. In *ICLR*, 2021.

[53] Xiuying Wei, Yunchen Zhang, Yuhang Li, Xiangguo Zhang, Ruihao Gong, Jinyang Guo, and Xianglong Liu. Outlier suppression+: Accurate quantization of large language models by equivalent and effective shifting and scaling. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

[54] Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

[55] Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. Smoothquant: Accurate and efficient post-training quantization for large language models. In *ICML*, pages 38087–38099. PMLR, 2023.

[56] Junrui Xiao, Zhikai Li, Lianwei Yang, and Qingyi Gu. Patch-wise mixed-precision quantization of vision transformer. In *2023 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2023.

[57] Yuedong Yang, Hung-Yueh Chiang, Guihong Li, Diana Marculescu, and Radu Marculescu. Efficient low-rank backpropagation for vision transformer adaptation. *NeurIPS*, 36:14725–14736, 2023.

[58] Hao Yu and Jianxin Wu. Compressing transformers: features are low-rank, but weights are not! In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11007–11015, 2023.

[59] Shixing Yu, Zhewei Yao, Amir Gholami, Zhen Dong, Sehoon Kim, Michael W Mahoney, and Kurt Keutzer. Hessian-aware pruning and optimal neural implant. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3880–3891, 2022.

[60] Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *ECCV*, pages 191–207, 2022.

[61] Yunshan Zhong, Jiawei Hu, You Huang, Yuxin Zhang, and Rongrong Ji. ERQ: Error reduction for post-training quantization of vision transformers. In *ICML*, volume 235, pages 61664–61680, 2024.

[62] Yunshan Zhong, You Huang, Jiawei Hu, Yuxin Zhang, and Rongrong Ji. Towards accurate post-training quantization of vision transformers via error reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025.