BLOB-Q: Boosting Low Bit ViT Quantization via Global Optimization on Model Distortion

Zhe Wang[1,2*], Kaixin Xu[1,2*], Xue Geng[1], Fen Fang[1], Mohamed M. Sabry Aly[2], Xulei Yang[1†], Min Wu[1†], and Weisi Lin[2]

Institute for Infocomm Research (I^{2}R), Agency for Science, Technology and Research (A*STAR), 1 Fusionopolis Way, 138632, Singapore
{wang_zhe,xuk,geng_xue,fang_fen,yang_xulei,wumin}@a-star.edu.sg College of Computing and Data Science, Nanyang Technological University (NTU), Singapore
{msabry,wslin}@ntu.edu.sg

Abstract. In this paper, we present BLOB-Q, a novel Mixed-Precision Post Training Quantization (MPQ) approach for Vision Transformers (ViTs). MPQ aims to assign different bit-widths across layers under a model size budget. Most existing MPQ methods either rely on layerwise sensitivity heuristics or surrogate objectives. These approaches simplify optimization, but they no longer optimize the same model-level objective that determines accuracy. Others solve the global MPQ problem by non-heuristic and non-analytical algorithms, such as Reinforcement Learning, genetic algorithms or gradient-based learning. However, they are inefficient for modern large vision transformers. In this work, we revisit the global MPQ problem and show that this problem can be solved by efficient analytical algorithms while still maintaining its global-optimality. This comes from two key insights. Firstly, through systematic signal analysis on quantization errors, we discover that under 4∼6-bit range, the quantization error on converged ViTs are indeed small perturbations. This encourages that the global MPQ objective is possible to be quadratically decomposed for empirical ViT models. Secondly, motivated by the small perturbation observation, we further discover that the global MPQ objective satisfies an additivity property. Utilizing the additivity property, the NP-hard global MPQ optimization problem can be decomposed into sub-problems and solved practically, while still retain the global optimality. Specifically, we solve the decomposed analytical optimization problem using dynamic programming algorithm, which efficiently find the globally optimal solution with only linear time complexity. Extensive experiments on numerous ViT models demonstrate the effectiveness of our approach. Results show that BLOB-Q significantly improves state-of-the-art and can further reduce the size of ViT models to 4 bits to 6 bits without hurting ImageNet accuracy. Moreover, BLOB-Q is highly efficient, only requiring on average sub-2 minutes for regular ViT-S to ViT-B models.

Keywords: Model Quantization · ViT · Mixed-precision Quantization

* Equal contribution. † Co-corresponding author.

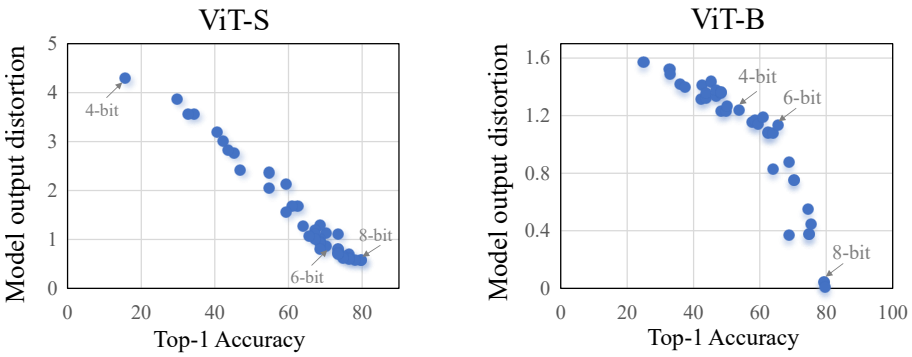

Figure 1: Correlation between the ℓ_2 model-output distortion and final ImageNet accuracy. Varying the budget bit-rate in BLOB-Q yields different mixed-precision solutions. On both ViT-S and ViT-B, the optimization objective tracks the eventual model accuracy closely.

1 Introduction

ViTs [6, 26, 36] achieve state-of-the-art performance across many tasks, including image classification [16], segmentation [11], style transfer [8, 13], and image synthesis [14]. However, this performance comes with substantial computational and memory cost [9, 43]. Modern ViT models typically contain tens or even hundreds of millions of parameters, making the deployment on resource-limited devices such as mobile phones, drones, and autonomous vehicles challenging. Quantization [10, 12, 33, 48, 51–53] is an effective approach to compress deep neural networks. Post-training quantization (PTQ) is a critical technique for compressing large Vision Transformers (ViTs). PTQ does not require retraining or fine-tuning the model after quantization and is more efficient and feasible, since retraining itself is time-consuming, and even impossible in some cases when the training dataset is not available. While the uniform-precision PTQ [3,21,24,28,30] is computationally convenient, it is widely recognized that different layers exhibit drastically distinct sensitivities to quantization. Hence instead of quantizing all layers to the same precision, Mixed-precision quantization (MPQ) [5, 34, 37, 44] allocates different bit-widths across layers, leading to higher model accuracy compared to uniform-precision PTQ.

To achieve the ultimate goal of MPQ, *i.e.* to preserve maximal model accuracy after quantization, one can directly minimize the discrepancy between the full-precision model output and the quantized model output, subject to a target bit budget on the weights and activations. We verified that the ℓ_2 distance of the model output is strongly correlated with the model accuracy under various bit-rates, as shown in Fig. 1, confirming that minimizing this model-level distortion is well aligned with maximizing accuracy.

Main challenges. Despite numerous efforts to tackle this problem to minimize the model accuracy drop in MPQ, there are two major challenges hindering the practical solution to this problem. Firstly, to find the optimal layerwise bit allocation, one challenge for such global objective is the *exponentially increasing search space* of the optimization variable. Given L layers and B bit width options,

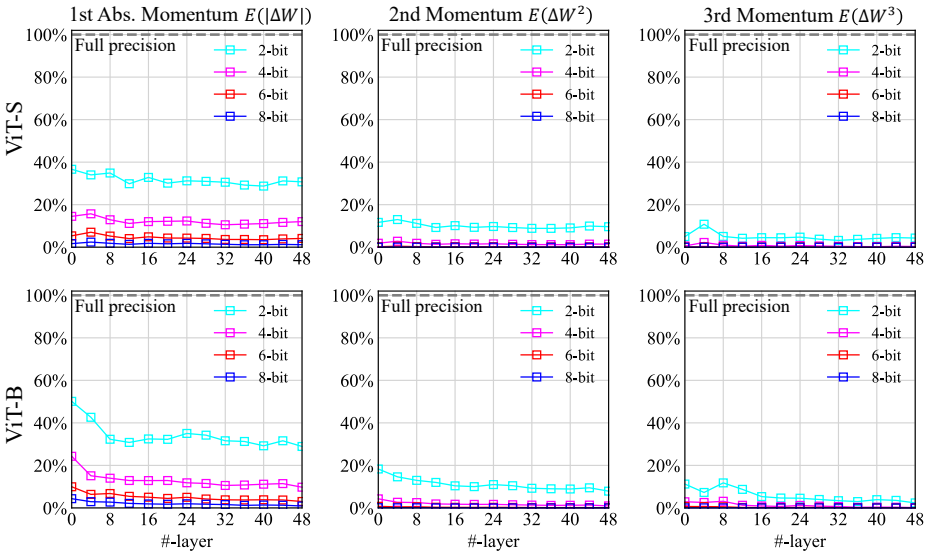

Figure 2: Signal analysis on the magnitude of quantization errors $|\Delta W|$ in ViTs. As the moment order increases, the magnitude of the quantization error vanishes rapidly compared to that of the full-precision weights. From the 3rd moment on, the error becomes negligible across models and layers, except for ultra-low-bit (2-bit).

the search space becomes $O(B^L)$, where in a deep neural network, L can be tens or hundreds of layers. Such a huge search space could make it extremely time-consuming for a heuristic search method, such as reinforcement learning [35]. Another major challenge for minimizing such global objective is the *coupled nonlinearity* of the impact of quantization. Due to the complexity of the model architectures, the quantization errors introduced at intermediate layers interact in complex ways. As a result, it seems impossible to decouple the impact of quantization at individual layers to the global objective.

Existing workarounds. Due to the above challenges, existing attempts relax this global optimization problems in different ways, which will inevitably suffer from sub-optimality and efficiency issues. They either (1) Avoid the global objective and instead optimize local surrogates [5, 32]; or (2) Rely on Reinforcement Learning (RL), heuristic search gradient-based procedures without global optimality guaranties [32, 34, 37].

Key insight. In this paper, we address this key problem of optimizing the global optimization problem in MPQ in a principled way. By conducting a systematic signal analysis on the quantization errors, we show that it is possible to decouple the global objective into analytical terms w.r.t. the quantization errors. As a result, we are able to employ an analytical algorithm to efficiently solve the global MPQ problem with only linear complexity for the first time.

The first key insight of this work is that the quantization errors are small perturbations around a converged ViT model. We reveal that under normal bit-width ranges, the ViT models are qualified for decomposition of the global distortion up to the second-order at their convergence, where the third moment

of quantization errors is well smaller than the second moment (Fig. 2). This allows us to decompose the global optimization objective of MPQ into a quadratic analytical formulation in majority of the bit search space. Another important finding of this work is that pretrained ViTs satisfy an important additivity property, where the collective impact of all-layer quantization to the global objective can be considered as additive impact of quantizing individual layers.

Building on these two key discoveries, we propose BLOB-Q, a principled, accuracy preserved, and efficient Post-training Mixed-precision Quantization framework for ViTs. BLOB-Q solves the global MPQ problem by structurally decomposing the non-linear global objective into a quadratic analytical expansion on the quantization error. This structural reduction analytically associates the layerwise bit-widths to be optimized with the global objective, and converts the original NP-hard nonlinear formulation into a separable integer optimization problem. The resulting problem can be solved efficiently using dynamic programming, yielding a globally optimal solution with linear complexity against the number of layers. This reformulation preserves alignment with the same global objective, rather than introducing surrogate layerwise criteria seen in prior arts.

As a result, BLOB-Q achieved state-of-the-art quantization performance on ViTs under low-bit quantization under post-training setting. Noticeably, we observe no accuracy drop under W6A6 mixed-precision quantization for DeiT-B model from the full-precision pretrained model, requiring less than 2 minutes to obtain the results without retraining. We summarize our contributions as follows:

- We propose BLOB-Q, a new Mixed-Precision PTQ approach for ViTs, which globally minimizes model distortion caused by quantization. Our approach finds the optimal bit allocation of layers with minimized model distortion and can well maintain the accuracy even when quantized to very low bit widths.
- We propose an efficient method to solve the global MPQ problem by quadratic analytical decomposition on the global objective, via signal analysis and the discovered additivity property. An ultra-fast dynamic programming algorithm is developed to find the global-optimal solution with only linear complexity.
- BLOB-Q significantly outperforms state-of-the-art. For the first time, at 6 bits, we report the PTQ results on ViTs without hurting accuracy (loss $<$ 1%). At 4 bits, our approach noticeably improves the PTQ results up to 11.49% compared with prior arts.

2 Related Works

2.1 Post-Training Quantization.

Model quantization can be generally categorized into Quantization-Aware Training (QAT) and Post-Training Quantization (PTQ), where the latter PTQ performs quantization for a trained model. Recently, PTQ is more preferred as it is more lightweight without requiring to intensive training-from-scratch. Retrain-free [7, 19, 30, 49] approaches have become ones of desired traits of scalable PTQ for modern models. Plenty of PTQ techniques have been explored for

Feature	PTQ	PTQ4ViT	LAPQ	HAWQ	BRECQ, HAQ, P^2-ViT	BLOB-Q
Model-level Objective	✗	✓	✓	✓	✓	✓
Mixed-Precision	✓	✗	✗	✓	✓	✓
Global Solver ***(Joint optimality)***	✗	✗	✓	✗	✓	✓
Analytical Solver ***(Efficiency)***	✓	✓	✗	✓	✗	✓

Figure 3: Comparisons of different PTQ techniques. For MPQ methods, we compare only their mixed-precision strategies if different from other parts of their optimization.

ViTs [3, 4, 20, 21, 23, 25, 28], mostly at 8 bits. Few others [46] explored low bit PTQ down to 4 bits with noticeable accuracy drops. We achieve 4-bits PTQ with vastly shrinked accuracy gaps, and 6-bits PTQ with full-precision level accuracies.

2.2 Mixed-Precision Quantization (MPQ)

Adapting to different sensitivity to quantization on different layers of the model, MPQ aims to allocate varying bit widths within the model (commonly in layerwise granularity). Existing methods for MPQ mainly fall into two categories.

Local-level Optimization. Some PTQ works discussed mixed-precision scheme using locally defined (layerwise) proxy metrics to allocate bit-widths. VT-PTQ [28] measures quantization sensitivity using the nuclear norm of attention maps, then greedily solve the bit-widths. HAWQ [5] initially introduces model-level objectives but eventually applied heavy relaxations into layerwise surrogates to solve by lagrangian-based or ranking algorithms. Since these approaches rely on locally defined proxies, they do not directly optimize a model-level objective.

Model-level Optimization. To make the quantization strategy aware of task loss, several works formulate MPQ using model-level objectives. Some methods solve the resulting problem through heuristic search. Reinforcement-learning based approaches such as HAQ [38] and AutoQ [29] search for mixed-precision policies, while BRECQ [19] and P^2-ViT [34] refines bit allocations using genetic algorithms and PTMQ [44] and FracBits [45] learns differentiable bit variables. These approaches treat bit allocation as a black-box search problem and therefore require expensive search procedures. Other works optimize model-level distortion through gradient-based reconstruction. LAPQ [32] optimizes on a global loss by the iterative Power's algorithm similar to gradient method. PTQ4ViT [46] similarly use global loss to obtain Hessian guided proxy at each layer to locally search quantization stepsizes. However, these approaches rely on gradient-based optimization or multi-stage procedures and do not exploit the analytic structure between bit-width and model distortion. Fig. 3 illustrates a comprehensive comparison between our method and these related approaches.

3 Preliminaries and Problem Statements

Following prior works on Mixed-Precision Quantization (MPQ), *e.g.*, [2, 5, 28, 37], we explore the MPQ problem for ViTs. Compared to Uniform-Precision Quantization (UPQ) which uses the same bit width throughout the model, MPQ

works have shown better accuracy via unevenly allocating bit widths in the model. The general goal of MPQ is to solve the best quantization configuration, *a.k.a.* searching a set of bit widths $B_l^W \in \mathbb{R}^L$ and $B_l^A \in \mathbb{R}^L$ for all L layer weights and activations respectively in a model.

3.1 Layer Distortion Minimization

Previously, most PTQ approaches [2, 5, 19, 28, 31] avoid optimizing on a model-level objective as it is NP-hard for non-linear DNNs. They generally can be summarized into the following framework, that individually solves a series of layer-level problems capturing the local impact of quantization on current layer output, under a generic compression constraint $\mathcal{R}(\widehat{\boldsymbol{W}}) \leqslant R$ that the total size $\mathcal{R}$ of the quantized model parametrized by all layer weights $\widehat{\boldsymbol{W}} = \{\widehat{\boldsymbol{W}_1}, \widehat{\boldsymbol{W}_2}, ..., \widehat{\boldsymbol{W}_L}\}$ is under a budget size R:

$$\arg\min_{\widehat{\boldsymbol{W}}_l} \sum_l \Gamma(\boldsymbol{O}_l, \widehat{\boldsymbol{O}}_l), \quad \text{s.t.} \quad \mathcal{R}(\widehat{\boldsymbol{W}}) \leqslant R. \tag{1}$$

Here $\boldsymbol{O}_l$ denotes the output of the layer and $\widehat{\boldsymbol{O}_l}$ denotes the output of the quantized layer. Such layer distortion minimization framework works well for previous Uniform-Precision Quantization (UPQ) works [19, 31] that reconstruct the quantized weight locally (layerwisely or blockwisely) via adjusting rounding parameters in the specific form of $\arg\min_{\widehat{\boldsymbol{W}}_l} \mathbb{E}_X(\|\boldsymbol{O}_l - \widehat{\boldsymbol{O}}_l\|_2^2)$. Most recent Mixed-Precision Quantization (MPQ) attempts [2, 5, 28] also deduct from this layer distortion minimization to search for optimal bit allocation resulting to ranking sensitivity by *e.g.*, nuclear forms [1] and Hessian terms [5].

3.2 Model Distortion Minimization.

Despite most previous PTQ works circumvent the intimidating global optimization and resort to exploring local effect of quantization for efficient solution, in this paper, we aim to tackle the more difficult model-level optimization. Since the effect of quantization on model distortion directly links to the eventual model accuracy, theoretically it leads to a more accurate quantization solution (see Fig. 1). Formally, model distortion minimization refers to the following framework:

$$\arg\min_{\widehat{\boldsymbol{W}}_l, \widehat{\boldsymbol{A}}_l} \Gamma(\boldsymbol{O}, \widehat{\boldsymbol{O}}), \quad \text{s.t.} \quad \mathcal{R}(\boldsymbol{W}, \boldsymbol{A}) \leqslant R, \tag{2}$$

where it directly optimizes the best quantization that leads to minimal model distortion in the output of the last layer $\boldsymbol{O} = \boldsymbol{O}_L$. Unlike many MPQ works which only search for weight quantization configurations, we include both weights and activations $\widehat{\boldsymbol{W}}_l, \widehat{\boldsymbol{A}}_l$ in the MPQ optimization. We define distortion Γ as the expected ℓ_2 distance: $\Gamma(\boldsymbol{O}, \widehat{\boldsymbol{O}}) = \mathbb{E}_X(\|\boldsymbol{O} - \widehat{\boldsymbol{O}}\|_2)$.

The obvious challenge to tackle the above model-level optimization framework practically is the non-trivial task to find out close form analytical relationship

between perturbation on layer weights and activations of non-linear models. Heuristic search-based method like HAQ [37] and P^2-ViT [34] optimize similar global task criteria (*e.g.* cross entropy), but rely on Reinforcement Learning (RL) or evolutionary serach which can be extremely time-consuming to search a solution especially when model sizes scale up in modern models like ViTs.

4 Small Signal Analysis on Quantization Error

In the global MPQ optimization problem Eq. 2, the effect of quantizing different layers interact nonlinearly. Therefore, in order to solve it through analytical method, one way is to directly measure the global effect of every bit configuration by brute force. However, building such a look-up-table requires populating B^L different combinations of bit configurations jointly, which is impractical for PTQ setting. BRECQ [19] circumvents this challenge by aggressively assuming higher bits ($\geqslant$4-bit) almost equally contributes to the objective, and narrows down the search space to only 2-bit versus others. To avoid such expensive process, another way is to approximate the objective into analytical combination of the bit-width of all the layers. One possibility is to use Taylor expansion under perturbations on weights and activations around the working point at the converged model, we can approximate the model output $\boldsymbol{O}$ as: (first order term is typically ignored for converged models where first-order gradient vanishes [32])

$$\hat{\boldsymbol{O}} \approx \boldsymbol{O} + \sum_{l=1}^{L} \frac{1}{2} \Delta \boldsymbol{W}_l^\top \boldsymbol{H}_l^w \Delta \boldsymbol{W}_l + \sum_{l=1}^{L} \frac{1}{2} \Delta \boldsymbol{A}_l^\top \boldsymbol{H}_l^a \Delta \boldsymbol{A}_l + \mathcal{O}(\|\Delta \boldsymbol{W}_l\|^3) + \mathcal{O}(\|\Delta \boldsymbol{A}_l\|^3), \tag{3}$$

where $\boldsymbol{H}^w$ and $\boldsymbol{H}^a$ are Hessian of weights and activations. However, such mathematical approximation may not be directly applicable for quantizing empirical ViT models, where the higher-order terms $\mathcal{O}(\|\Delta \boldsymbol{W}_l\|^3) + \mathcal{O}(\|\Delta \boldsymbol{A}_l\|^3)$ may be required to accurately approximate it.

We show that this is possible *when the quantization error is small perturbation on the converged models*. In Fig. 2, we systematically examine the distribution of the magnitude of quantization errors on various pretrained ViTs, using symmetrical linear quantizer on weight tensors. For each layer, we analyze the first, second, to the third moments of the magnitude of quantization errors under different bit-widths, where N-th moment $E(|\Delta W|^N)$ captures different characteristics of the distribution. Empirically, we observe that the third moment is already very close to zero and can be ignored, across most of the bit ranges ($\geqslant$4-bit). This imply that to approximate the global objective (model distortion), it is practically safe to include up to quadratic terms w.r.t. the quantization error, and the higher-order information can be discarded.

5 Analytical Approximation of Model Distortion

Based on the conclusion above, we can then conduct analytical approximation on the global MPQ objective on the model distortion. The ℓ_2 distortion when all

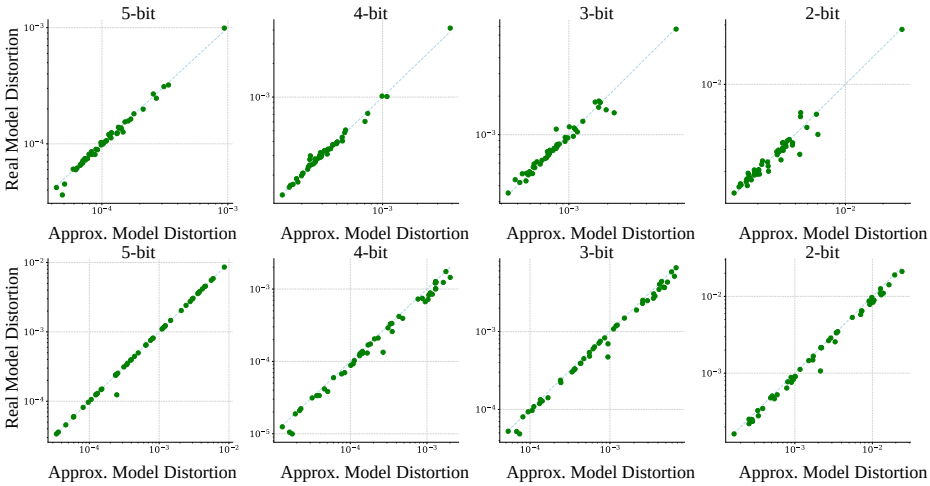

Figure 4: Empirical Evidences of Additivity property on ViT models. The first row evaluates on the weights of ViT-S model, and the second row evaluates on the activations of the DeiT-B. The X-axis represents the model distortion when approximating via additivity property (Finding 1) when quantizing all layer **activations**. The Y-axis represents the real model distortion by directly measuring the ℓ_2 distortion on model output. Data points inclines to the diagonal lines very well, showing that the additivity property holds.

the layer weights are perturbated can be written in quadratic form:

$$\Gamma(\boldsymbol{O}, \hat{\boldsymbol{O}}) = \mathbb{E}_X \left[\|\boldsymbol{O} - \hat{\boldsymbol{O}}\|_2^2 \right] \approx \mathbb{E} \left[\left\| \sum_{l=1}^{L} \frac{1}{2} \Delta \boldsymbol{W}_l^\top \boldsymbol{H}_l^w \Delta \boldsymbol{W}_l + \sum_{l=1}^{L} \frac{1}{2} \Delta \boldsymbol{A}_l^\top \boldsymbol{H}_l^a \Delta \boldsymbol{A}_l \right\|_2^2 \right] \tag{4}$$

Based on this analytical approximation, we discover the following finding when doing arbitrary quantization:

Finding 1 (Additivity Property)**.** *The expected model distortion $\Gamma(\boldsymbol{O}, \hat{\boldsymbol{O}})$ between the output of a pretrained model $\boldsymbol{O}$ and a quantized one $\hat{\boldsymbol{O}}$ on dataset X can be approximated by the sum of model distortion due to the quantization of individual layers:*

$$\Gamma(\boldsymbol{O}, \hat{\boldsymbol{O}}) = \mathbb{E}_X(\|\boldsymbol{O} - \hat{\boldsymbol{O}}\|_2^2) \approx \sum_l \Gamma_{\widehat{\boldsymbol{W}}_l}(\boldsymbol{O}, \hat{\boldsymbol{O}}) + \sum_l \Gamma_{\hat{\boldsymbol{A}}_l}(\boldsymbol{O}, \hat{\boldsymbol{O}}), \tag{5}$$

We provide a mathematical derivation for Finding 1 in the supplementary material and empirical evidences in Fig. 4. This finding also tells us that instead of optimizing the original model distortion which is NP-hard, we can instead solve an integer programming problem on layerwise contributions to the model distortion as in below Corollary.

Corollary 1 (Model Distortion Optimization)**.** *Given Eq. 5, the model distortion minimization problem in Eq. 2 can be turned into additive global objective:*

$$\arg\min_{\widehat{\boldsymbol{W}}_l, \hat{\boldsymbol{A}}_l} \sum_l \Gamma_{\widehat{\boldsymbol{W}}_l}(\boldsymbol{O}, \hat{\boldsymbol{O}}) + \sum_l \Gamma_{\hat{\boldsymbol{A}}_l}(\boldsymbol{O}, \hat{\boldsymbol{O}}), \quad \text{s.t. } \mathcal{R}(\widehat{\boldsymbol{W}}, \boldsymbol{A}) \leqslant R. \tag{6}$$

Since bit-widths B_l^W, B_l^A only control the quantization of single layer weights $\widehat{\boldsymbol{W}}_l$ and activations $\widehat{\boldsymbol{A}}_l$, they also only affect $\Gamma_{\widehat{\boldsymbol{W}}_l}$ and $\Gamma_{\widehat{\boldsymbol{A}}_l}$ of current layer. Furthermore, as the size constraint $\mathcal{R}$ can be instantiated as the linear function of bit-widths *i.e.*, $\mathcal{R}(\boldsymbol{W}, \boldsymbol{A}) = \sum_l B_l^W \|\boldsymbol{W}_l\|_0 + B_l^A \|\boldsymbol{A}_l\|_0$, The model distortion minimization problem in Eq. 6 is equivalent to an integer programming problem:

$$\arg\min_{B_l^W, B_l^A} \sum_l \Gamma_{\widehat{\boldsymbol{W}}_l}(\boldsymbol{O}, \widehat{\boldsymbol{O}}) + \sum_l \Gamma_{\widehat{\boldsymbol{A}}_l}(\boldsymbol{O}, \widehat{\boldsymbol{O}}), \quad \text{s.t. } \sum_l B_l^W \|\boldsymbol{W}_l\|_0 + B_l^A \|\boldsymbol{A}_l\|_0 \leqslant R, \tag{7}$$

with only the concrete values of $\Gamma_{\widehat{\boldsymbol{W}}_l}$ and $\Gamma_{\widehat{\boldsymbol{A}}_l}$ under different bit widths needed to be obtained via calibration, detailed in Section. 5. Supported by the additivity property in Eq. 5, we can guarantee to attain the global-optimal of the Model-level objective in Eq. 2 by solving the integer programming problem. We eventually devise a non-greedy solver for problem Eq. 7 detailed in Section. 6.

Efficient Calibration of Model Distortion To efficiently compute the second-order taylor expansion for ViTs, we improve scalability of the empirical Fisher [17] approximation method for model Hessians on ViTs by introducing matrix level parallelism along data dimension. For weight with the dimension of $\boldsymbol{W} \in \mathbb{R}^{d_{row} \times d_{col}}$ ($d = d_{row} d_{col}$), we first define the Jacobian matrix $\boldsymbol{J}_n(\boldsymbol{W})$ of the n-th calibration data out of calibration dataset of N samples. Then we concatenate all N samples of Jacobians into $\mathbf{J}_N(\boldsymbol{W}) \in \mathbb{R}^{N \times d}$ and obtain a N-dimensional variable $\mathbf{P}^w = \mathbf{J}_N^\top(\boldsymbol{W}) \Delta \boldsymbol{W} \in \mathbb{R}^N$. We use the stored intermediate matrix $\mathbf{P}_l^w$ to obtain the distortion under weight quantization at l-th layer as:

$$\Gamma_{\widehat{\boldsymbol{W}}_l}(\boldsymbol{O}, \widehat{\boldsymbol{O}}) \approx \frac{\kappa}{2} \|\Delta \boldsymbol{W}_l\|_2^2 + \frac{1}{2N} \text{trace}(\mathbf{P}_l^w \mathbf{P}_l^{w\top}). \tag{8}$$

As for activations, since there are N copies of activations $\boldsymbol{A}^N = \{\boldsymbol{A}^{(1)}, \boldsymbol{A}^{(2)}, ..., \boldsymbol{A}^{(N)}\} \in \mathbb{R}^{N \times dT}$ (T is the number of visual tokens in ViTs), we similarly define $\boldsymbol{P}^a = \text{diag}(\boldsymbol{J}_N(\boldsymbol{A}^N) \Delta \boldsymbol{A}^{N\top}) \in \mathbb{R}^N$. Then we can substitude $\Delta \boldsymbol{A}$ and $\boldsymbol{P}^a$ into Eq. 8 to calibrate the distortion for activation quantization:

$$\Gamma_{\widehat{\boldsymbol{A}}_l}(\boldsymbol{O}, \widehat{\boldsymbol{O}}) \approx \frac{\kappa}{2} \|\Delta \boldsymbol{A}_l\|_2^2 + \frac{1}{2N} \text{trace}(\boldsymbol{P}_l^a \boldsymbol{P}_l^{a\top}). \tag{9}$$

With the efficient computation above, the holistic calibration process is streamlined to a one-time forward+backward pass on calibration set, followed by using Eq. 8 and Eq. 9 to rapidly get the distortions $\Gamma_{\boldsymbol{W}_l}, \Gamma_{\boldsymbol{A}_l}$ for each layers. The step-by-step process are summarized in Line 1-10 in Algo. 1. We provide detailed asymptotic time and space complexity studies of the calibration in the supplementary material.

6 Global Optimal Solver via Dynamic Programming

We adopt Dynamic Programming (DP) algorithm to recursively solve the integer problem by decomposing the whole problem into multiple sub-problems and then processing each sub-problem separately. Defining the state function required by

DP as $S(l, r) \in \mathbb{R}$ is a function of discrete values layer $l \in [1, L]$ and remaining model total bits $r \in [1, R]$, this indicates the the minimal achievable model distortion at current intermediate state. Then the DP algorithm can be completed by the following state transferring function to recursively solve the original problem at state $S(L, R)$:

$$S(l, r) = \min_{1 \leqslant b_1 \leqslant B} \{\Gamma_{q(\boldsymbol{Z}_l, b_1)}(\boldsymbol{O}, \widehat{\boldsymbol{O}}) + S(l-1, r - b_1 d_l)\}, \tag{10}$$

where $\boldsymbol{Z}_l$ represents the either one between $\boldsymbol{W}_l$ and $\boldsymbol{A}_l$ and we treat states for weights and activates the same as they both contribute to the total bit budget R. $q(\boldsymbol{Z}_l, b)$ is the optimal b-bit quantizer for input $\boldsymbol{Z}_l$, B is the maximum allowed bit-width for each layer (we set to $B = 10$). We set the boundary condition of $S(0, :) = 0$. As we populate through the above states $S(l, r)$, we use another function $G(l, r) \in \mathbb{Z}_+$ to record all intermediate selections of b_1 made:

$$G(l, r) = \arg\min_{1 \leqslant b_1 \leqslant B} \{\Gamma_{q(\boldsymbol{Z}_l, b_1)}(\boldsymbol{O}, \widehat{\boldsymbol{O}}) + S(l-1, r - b_1 d_l)\}. \tag{11}$$

Using $G(l, r)$, we can iterate through layers to find the optimal bit-width B_l at layer l.

The theoretical time complexity of the DP-based solver is $O(2LRB^2)$, as we have $2L \times R$ different states since we consider two variable bit-widths *i.e.* weight and activation for each layer. It shows the algorithm scales linearly with the model size. Moreover, the solution found by dynamic programming is the globally optimal solution. Taking advantages of modern parallel compute like GPU, the DP-solver can be further accelerated towards practical usages. We provide more details of our implemented DP-solver in the supplementary materials, incorporating strategies like vectorized updating and Run-length coding, with source code attached. Empirically, we find that DP-solver only requires to perform in few seconds as we show in Tab. 4. We provide the end-to-end pseudocode of our proposed BLOB-Q algorithm in Algo. 1.

7 Experiments

Implementation Details. We randomly select 512 samples from the ImageNet training set as the calibration dataset without any augmentations, and evaluate the quantized ViTs on the full validation set. We conduct all experiments on Nvidia 4-L40 with 48GB of VRAM on each GPU. For bias correction, we adopt the strategies same as [28] to rectify the layerwise output distribution. We quantize all the Conv2D and Linear layers in ViTs, including the first and last layer and excluding LayerNorm and Embedding layers. For the underlying quantizer, we integrate BLOB-Q on top of a RepQ-ViT [21] weight quantizer together with a log-2 activation quantizer, and adopt block-wise reconstruction to refine the quantized weights. We provide more details in the supplementary materials.

Main Results. We compare our approach with prior Post-Training Quantization (PTQ) methods for ViTs on six ViT models (ViT-S, ViT-B, ViT-B/384, DeiT-S, DeiT-B, and DeiT-B/384) on the ImageNet validation dataset, including

Algorithm 1 BLOB-Q: Model-level Global-optimal Mixed-precision Quantization Algorithm.

Require: Calibration dataset X_{cal} with N samples, pretrained model with L layers, maxmimum allowed bit-width B, target total model size R.
Ensure: Optimal bit-widths for layer Weights and activations $B_l^W, B_l^A \in \mathbb{R}, 1 \leqslant l \leqslant L$.

A forward+backward pass on X_{cal} to get $\boldsymbol{A}_N^{(l)}, J_N(\boldsymbol{W}_l), J_N(\boldsymbol{A}_l)$ ▷ Start calibration.
for each layer $l \in [1, L]$ **do**
  for each bit $b \in [1, B]$ **do**
    $\widehat{\boldsymbol{W}}_l = q(\boldsymbol{W}_l, b),\ \widehat{\boldsymbol{A}}_l = q(\boldsymbol{A}_l, b)$ ▷ Quantize W and A to b-bit.
    $\Delta\boldsymbol{W}_l = \boldsymbol{W}_l - \widehat{\boldsymbol{W}}_l,\ \Delta\boldsymbol{A}_l = \boldsymbol{A}_l - \widehat{\boldsymbol{A}}_l$ ▷ Quant. errors.
    $\boldsymbol{P}_l^w = \mathbf{J}_N^\top(\boldsymbol{W}_l)\Delta\boldsymbol{W}_l,\ \boldsymbol{P}_l^a = \mathrm{diag}(\boldsymbol{J}_N(\boldsymbol{A}_l^N)\Delta\boldsymbol{A}_l^{N\top})$
    $\Gamma_{q(\boldsymbol{W}_l,b)}(\boldsymbol{O}, \widehat{\boldsymbol{O}}) = \frac{\kappa}{2}\|\Delta\boldsymbol{W}_l\|_2^2 + \frac{1}{2N}\mathrm{trace}(\mathbf{P}_l^w\mathbf{P}_l^{w\top})$. ▷ Distortion w.r.t. W-quant. (Eq. 8)
    $\Gamma_{q(\boldsymbol{A}_l,b)}(\boldsymbol{O}, \widehat{\boldsymbol{O}}) = \frac{\kappa}{2}\|\Delta\boldsymbol{A}_l\|_2^2 + \frac{1}{2N}\mathrm{trace}(\boldsymbol{P}_l^a\boldsymbol{P}_l^{a\top})$. ▷ Distortion w.r.t. A-quant. (Eq. 9)
  end for each
end for each
Create matrices `S`, `G` $\in \mathbb{R}^{2L+1,R}$ ▷ Start DP-solver (Sec.6).
Initialize `S[1,r]`$\leftarrow 0,\ \forall r \in [1, R]$ if $l = 0$ else ∞.
for each layer $l \in [1, 2L]$ **do** ▷ (have $2L$ for both weights and activations)
  for each size $r \in [1, R]$ **do**
    Update `S[1,r]` using Eq. 10, Update `G[1,r]` using Eq. 11 ▷ Recursively populate states.
  end for each
end for each
B_L^W = `G[2L,R]` ▷ Last layer weight bit-width.
for each layer l reversely from L to 1 **do** ▷ Recursively obtain layerwise bit-widths.
  B_l^A = `G[2l, R -` $B_l^W\|\boldsymbol{W}_l\|_0$`]` ▷ optimal activation bit-width.
  $R \leftarrow R - B_l^W\|\boldsymbol{W}_l\|_0$
  B_l^W = `G[2l-1, R -` $B_l^A\|\boldsymbol{A}_l\|_0$`]` ▷ layerwise weight bit-width.
  $R \leftarrow R - B_l^A\|\boldsymbol{A}_l\|_0$
end for each

VT-PTQ [28], PTQ4ViT [47], Percentile [18], EasyQuant [39], APQ-ViT [4], NoisyQuant [24], I-ViT [20], FQ-ViT [22], PMQ [42], P^2-ViT [34], PTMQ [44], and the most recent FIMA-Q [40], APHQ-ViT [41] and LAMPQ [15]. Tab. 1 illustrates the results. Our approach consistently outperforms the baseline methods on almost all configurations, especially at low bit-widths. At 4-bit, BLOB-Q surpasses the strongest competitor by 0.69%, 0.45%, 0.74% and 2.82% on ViT-S, DeiT-S, DeiT-B and ViT-B/384 respectively, and by 2.54% on DeiT-B/384. On ViT-B at 4-bit, FIMA-Q is marginally higher than us by 0.10%, but it requires 7.3× more calibration runtime (3.4 hours), whereas BLOB-Q finishes in under one second. At 6-bit, BLOB-Q achieves the best accuracy on all six models and can quantize the ViT models without hurting accuracy (loss $< 1\%$).

Ablation Studies. Both model distortion minimization and global-optimal solution contribute to the results. We conduct an ablation study to evaluate the effectiveness of each of them, which is shown in Tab. 2. We compare three variants: (a) minimize layer distortion with a local solution, (b) minimize model distortion with a local solution, and (c) minimize model distortion with our global-optimal solution. As Tab. 2 shows, both ingredients contribute substantially to the final accuracy. For example, at 4 bits, switching from layer distortion to model distortion lifts ViT-S accuracy from 0.11% to 20.52%, and replacing the local solver with our global-optimal one further pushes it to **76.85**%. Similar gains are observed on the other models and bit-widths.

Impact of the Calibration Set. We evaluate the impact of the calibration set size on our approach by sweeping from 32 to 1024 images on DeiT-S, as

Table 1: Results on the ImageNet validation dataset when ViT models are quantized to 4 bits and 6 bits. Top-1 classification accuracy (%) is evaluated. MP denotes quantization with Mixed-Precision. For a fair comparison, we report the results with the same average bit widths. For example, 4 MP means that we quantize the ViT models into 4 bits on average across the layers. The best result of each group is highlighted in bold. New baselines added during the camera-ready revision are highlighted in blue.

Method	Size (Bit)	ViT-S	ViT-B	ViT-B/384	DeiT-S	DeiT-B	DeiT-B/384
Full Precision	32	81.39%	84.54%	86.05%	79.87%	81.85%	83.12%
PTQ4ViT [47]	4	42.57%	30.69%	-	34.08%	64.39%	-
APQ-ViT [4]	4	47.95%	41.41%	-	43.55%	67.48%	-
RepQ-ViT [21]	4	65.60%	68.48%	-	69.03%	75.61%	-
P^2-ViT [34]	4 MP	64.24%	79.93%	-	75.26%	79.37%	-
FIMA-Q [40]	4	76.16%	82.94%	77.48%	76.62%	80.20%	79.10%
APHQ-ViT [41]	4	76.07%	82.41%	-	76.40%	80.21%	-
LAMPQ [15]	4 MP	74.02%	81.91%	-	75.40%	79.24%	-
BLOB-Q	4 MP	**76.85%**	**82.84%**	**80.30%**	**77.07%**	**80.95%**	**81.64%**
VT-PTQ [28]	6 MP	70.24%	75.26%	46.88%	75.10%	77.47%	68.44%
PTQ4ViT [47]	6	78.63%	81.65%	83.34%	76.28%	80.25%	81.55%
Percentile [18]	6 MP	67.74%	77.63%	77.60%	70.49%	73.99%	78.24%
EasyQuant [39]	6	75.13%	81.42%	82.02%	73.26%	75.86%	81.26%
APQ-ViT [4]	6	79.10%	82.21%	-	77.76%	80.42%	-
RepQ-ViT [21]	6	80.43%	83.62%	-	77.76%	80.42%	-
NoisyQuant [24]	6	78.65%	82.32%	83.22%	77.43%	80.70%	81.65%
PMQ [42]	6 MP	-	73.33%	-	76.68%	79.64%	-
P^2-ViT [34]	6 MP	32.05%	82.1%	-	77.56%	80.59%	-
PTMQ [44]	6 MP	76.09%	77.7%	-	78.74%	80.81%	-
FIMA-Q [40]	6	80.64%	84.82%	-	79.52%	81.74%	-
APHQ-ViT [41]	6	79.98%	84.83%	85.62%	79.17%	81.67%	82.83%
BLOB-Q	6 MP	**80.76%**	**85.07%**	**85.66%**	**79.65%**	**81.83%**	**82.97%**
PTQ4ViT [47]	8	81.00%	84.25%	85.82%	79.47%	81.48%	82.97%
APQ-ViT [4]	8	81.25%	84.26%	-	79.78%	81.72%	-
NoisyQuant [24]	8	81.15%	84.22%	85.86%	79.51%	81.45%	82.49%
I-ViT [20]	8	81.27%	84.76%	-	80.12%	81.74%	-
P^2-ViT [34]	8 MP	68.14%	83.00%	-	78.41%	80.93%	-
BLOB-Q	8 MP	**81.54%**	**84.89%**	**85.93%**	**81.25%**	**81.75%**	**82.85%**

shown in Tab. 3. The accuracy is not monotonic in the calibration size: a small number of samples is insufficient to capture the layerwise distortion statistics, while overly large calibration sets may cause calibration overfitting [**?**], leading to a slight degradation at 1024 images. We find that 512 images yields the best overall trade-off, with both 4-bit (76.13%) and 6-bit (79.65%) peaking near this point. We therefore use 512 calibration images in all of our main experiments.

Executive Time. We also evaluate the execution time of the optimization method and compare it with other baselines, including PTQ4ViT [47], REPQ-ViT [21], EasyQuant [39], and NoisyQuant [24]. Our method is much faster than others by orders of magnitute. As shown in Tab. 4, our method is 4.2× faster than PTQ4ViT [47], 4.9× faster than REPQ-ViT [21], 8.4× faster than EasyQuant [39], and 31.2× faster than NoisyQuant [24]. Different with prior works which need to search the solution from a huge space with exponential

Table 2: Ablation studies of our approach. The effectiveness of model distortion minimization and global-optimal solution are evaluated. We use a greedy algorithm to find a local solution in this table. For layer distortion, we use the output error of current layer in the learning objective. The supplementary materials provide more details about the local solution.

Comparison	Size	ViT-S	ViT-B	ViT-B/384	DeiT-S	DeiT-B	DeiT-B/384
Layer Distortion + Local	4 Bits	0.11%	0.10%	0.09%	0.10%	0.12%	0.12%
Model Distortion + Local	4 Bits	20.52%	1.25%	0.60%	0.16%	16.51%	49.92%
Model Distortion + Global	4 Bits	**76.85%**	**82.84%**	**80.30%**	**77.07%**	**80.95%**	**81.64%**
Layer Distortion + Local	6 Bits	0.10%	0.12%	0.11%	0.17%	14.01%	50.80%
Model Distortion + Local	6 Bits	62.87%	15.27%	11.04%	40.25%	44.08%	74.70%
Model Distortion + Global	6 Bits	**80.76%**	**85.07%**	**85.66%**	**79.65%**	**81.83%**	**82.97%**
Layer Distortion + Local	8 Bits	39.81%	1.25%	14.43%	0.12%	45.81%	76.08%
Model Distortion + Local	8 Bits	67.78%	52.34%	61.23%	74.88%	80.81%	81.12%
Model Distortion + Global	8 Bits	**81.54%**	**84.89%**	**85.93%**	**81.25%**	**81.75%**	**82.85%**

Table 3: Impact of the calibration size on DeiT-S. The best setting (512 images) is highlighted in bold.

No. images	32	64	128	256	**512**	1024
4 Bits	75.33%	75.18%	75.75%	76.19%	**76.13%**	76.09%
6 Bits	79.45%	79.39%	79.26%	79.56%	**79.65%**	79.42%

complexity, our method utilizes the additivity property and adopts dynamic programming find the global-optimal solution. As a result, our optimization algorithm has only linear time complexity and is much faster than others. We anaylzed the time and memory efficiency in more detail in the supplementary material.

Table 4: Executive time of optimization (seconds, averaged over 4/6/8-bit settings) on Nvidia L40. Full per-bit-width breakdown is provided in the supplementary material.

Method	ViT-S	ViT-B	DeiT-S	DeiT-B
PTQ4ViT [47]	102	190	101	201
REPQ-ViT [21]	123	249	117	250
EasyQuant [39]	211	464	212	464
NoisyQuant [24]	782	1849	781	1847
BLOB-Q	**33**	**121**	**34**	**129**

Plug-and-play Integration with Other PTQ Pipelines. BLOB-Q is a general bit-allocation strategy. Besides RepQ-ViT [21], which we use in the main results, we also apply BLOB-Q on top of ERQ [50], a recent PTQ method that performs error reduction on quantized weights and activations. Tab. 5 reports the results at 4-bit and 6-bit. BLOB-Q improves the accuracy of ERQ on all four models. For example, at 4-bit, BLOB-Q gives +3.51% on ViT-B and +5.31% on DeiT-S. These results show that BLOB-Q works with different underlying quantizers.

Table 5: Plug-and-play integration of BLOB-Q on top of ERQ [50]. We report top-1 ImageNet accuracy (%) at 4-bit and 6-bit. The best result of each group is highlighted in bold.

Method	4-bit				6-bit			
	ViT-S	ViT-B	DeiT-S	DeiT-B	ViT-S	ViT-B	DeiT-S	DeiT-B
ERQ [50]	68.91	76.63	72.56	78.23	80.48	83.89	79.03	81.41
ERQ + BLOB-Q (MP)	**69.58**	**80.14**	**77.87**	**80.74**	**81.04**	**84.81**	**79.92**	**82.03**

Generalization to More Datasets and Tasks. Our main experiments use ImageNet with ViT/DeiT backbones. We further test BLOB-Q on two more settings. The first setting is CIFAR-10 classification with ViT-S/B and DeiT-S/B. The second setting is COCO 2017 detection and instance segmentation, with Swin-T/S [27] as the backbone of Mask R-CNN. Tab. 6 summarizes the results. On CIFAR-10, BLOB-Q at 6-bit drops at most 0.31% on ViT-B and gains +0.05% on DeiT-S. ViT-S shows a larger drop of 4.43% because it is the smallest model and is more sensitive to quantization. At 4-bit, three of the four models stay within 2.67% drop. On COCO, BLOB-Q at 6-bit loses at most 0.6 bbox mAP and 0.5 mask mAP. At 4-bit, the loss is at most 1.4 bbox mAP and 1.3 mask mAP. BLOB-Q therefore also works on detection and segmentation at low bit-widths.

Table 6: Generalization of BLOB-Q to more datasets and tasks. We report top-1 accuracy (%) on CIFAR-10 and bbox / mask mAP on COCO 2017. Values in parentheses denote the drop from FP32.

Bit-width	CIFAR-10 Top-1 Accuracy (%)				COCO bbox mAP / mask mAP	
	ViT-S	ViT-B	DeiT-S	DeiT-B	Swin-T	Swin-S
6-bit	94.05 (-4.43)	98.49 (-0.31)	97.44 (+0.05)	97.94 (-0.29)	42.0 (-0.6) / 38.8 (-0.5)	47.8 (-0.4) / 42.7 (-0.5)
4-bit	88.62 (-9.86)	97.48 (-1.32)	94.72 (-2.67)	96.57 (-1.66)	41.2 (-1.4) / 38.0 (-1.3)	47.0 (-1.2) / 42.2 (-1.0)

Stability across Calibration Sets PTQ methods use a small calibration set, so the result may vary with different samples. We repeat the 4-bit mixed-precision experiment with three independently sampled calibration sets. Tab. 7 reports the mean and standard deviation of the top-1 accuracy. The standard deviation is at most $\pm 0.53\%$ on DeiT-S and below $\pm 0.15\%$ on the other three models. BLOB-Q is therefore stable at 4-bit when the calibration set changes.

Table 7: Stability of BLOB-Q's 4-bit mixed-precision results across 3 independently sampled calibration sets. We report mean$\pm$std of top-1 ImageNet accuracy (%).

Bit-width	ViT-S	ViT-B	DeiT-S	DeiT-B
4 MP	76.85 ± 0.08	82.65 ± 0.13	77.07 ± 0.53	80.95 ± 0.08

Hardware-Friendly Bit-Width Configurations. Our main results use bit-widths in the full range $[1, 10]$ to show the upper bound of BLOB-Q. In practice, most GPU kernels only support INT4 and INT8. We therefore restrict the per-layer bit-width to $\{4, 8\}$, so that a 6-bit average is realized by a mixture of INT4 and INT8 tensors. We compare BLOB-Q with P^2-ViT [34] under the same constraint. Tab. 8 reports the results. The constraint costs BLOB-Q 0.81% on ViT-B and 0.40% on DeiT-B compared to the full-range setting. The constrained

BLOB-Q still outperforms P^2-ViT by +2.16% on ViT-B and +0.84% on DeiT-B. BLOB-Q therefore works on mainstream GPU kernels with little accuracy loss.

Table 8: BLOB-Q under hardware-friendly bit-width constraints ($\{4, 8\}$, supported by mainstream GPU INT4/INT8 GEMM kernels), compared with P^2-ViT [34] under the same constraint. We report top-1 ImageNet accuracy (%). Δ is the gain of BLOB-Q over P^2-ViT under the constrained setting.

Model	Avg. Bit-width	**BLOB-Q** (bit$\in [1, 10]$)	**BLOB-Q** (bit=$\{4,8\}$)	P^2-ViT [34] (bit=$\{4,8\}$)	Δ
ViT-B	6 MP	85.07	84.26	82.10	+2.16
DeiT-B	6 MP	81.83	81.43	80.59	+0.84

We provide more experimental discussions in the supplementary materials, such as method explainability studies and hardware benchmark results, as well as implementation details and source codes in the supplementary materials to facilitate reproducibility.

8 Conclusion

In this work, we revisit the global mixed-precision post-training quantization (MPQ) problem for Vision Transformers. We show that the seemingly intractable global MPQ objective can be solved by an efficient analytical formulation while preserving its global optimality. Our approach is built on two key observations. First, quantization errors in converged ViTs behave as small perturbations under practical low-bit regimes, which allows the global objective to be approximated by a quadratic form. Second, this formulation reveals an additivity property of the global distortion, enabling the MPQ problem to be decomposed into independent sub-problems. Based on this structure, we solve the resulting optimization efficiently using dynamic programming with only linear complexity. Experiments on multiple ViT models demonstrate that BLOB-Q consistently achieves state-of-the-art post-training quantization performance, while remaining computationally efficient.

Acknowledgment

This research is supported by the Agency for Science, Technology and Research (A*STAR) under its MTC Programmatic Funds (Grant No. M23L7b0021) and MI Chapter Core Fund (Cost Center T30-JAC0002-A22-MI01-EOM, WBS element A22005). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of the A*STAR.

References

1. Banner, R., Nahshan, Y., Hoffer, E., Soudry, D.: Post-training 4-bit quantization of convolution networks for rapid-deployment. arXiv preprint arXiv:1810.05723 (2018)
2. Chen, J., Bai, S., Huang, T., Wang, M., Tian, G., Liu, Y.: Data-free quantization via mixed-precision compensation without fine-tuning. Pattern Recognition **143**, 109780 (2023)
3. Di Wu, Qi Tang, Y.Z.M.Z.D.Z.Y.F.: Easyquant: Post-training quantization via scale optimization (2020)
4. Ding, Y., Qin, H., Yan, Q., Chai, Z., Liu, J., Wei, X., Liu, X.: Towards accurate post-training quantization for vision transformer. In: Proceedings of the 30th ACM international conference on multimedia. pp. 5380–5388 (2022)
5. Dong, Z., Yao, Z., Gholami, A., Mahoney, M.W., Keutzer, K.: Hawq: Hessian aware quantization of neural networks with mixed-precision. In: arXiv (2019)
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021), `https://openreview.net/forum?id=YicbFdNTTy`
7. Frantar, E., Alistarh, D.: Optimal brain compression: A framework for accurate post-training quantization and pruning. Advances in Neural Information Processing Systems **35**, 4475–4488 (2022)
8. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: CVPR (2016)
9. Geng, X., Wang, Z., Chen, C., Xu, Q., Xu, K., Jin, C., Gupta, M., Yang, X., Chen, Z., Aly, M.M.S., et al.: From algorithm to hardware: A survey on efficient and safe deployment of deep neural networks. IEEE Transactions on Neural Networks and Learning Systems (2024)
10. Han, S., Pool, J., Tran, J., Dally, W.J.: Learning both weights and connections for efficient neural networks. arXiv preprint arXiv:1506.02626 (2015)
11. He, K., Gkioxari, G., Dollar, P., Girshick, R.: Mask r-cnn. In: CVPR (2017)
12. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R., Bengio, Y.: Quantized neural networks: Training neural networks with low precision weights and activations. In: arXiv:1609.07061 (2016)
13. Johnson, J., Alahi, A., Li, F.F.: Perceptual losses for real-time style transfer and super-resolution. In: ECCV (2016)
14. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: ICLR (2018)
15. Kim, M., Lee, J., Kim, J., Yun, J., Kwon, Y., Kang, U.: Lampq: Towards accurate layer-wise mixed precision quantization for vision transformers. In: The 40th Annual AAAI Conference on Artificial Intelligence (2026)
16. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NeurIPS (2012)
17. Kurtic, E., Campos, D., Nguyen, T., Frantar, E., Kurtz, M., Fineran, B., Goin, M., Alistarh, D.: The optimal bert surgeon: Scalable and accurate second-order pruning for large language models. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing. pp. 4163–4181 (2022)
18. Li, R., Wang, Y., Liang, F., Qin, H., Yan, J., Fan, R.: Fully quantized network for object detection. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2810–2819 (2019)

19. Li, Y., Gong, R., Tan, X., Yang, Y., Hu, P., Zhang, Q., Yu, F., Wang, W., Gu, S.: Brecq: Pushing the limit of post-training quantization by block reconstruction. In: International Conference on Learning Representations (2020)
20. Li, Z., Gu, Q.: I-vit: Integer-only quantization for efficient vision transformer inference. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17065–17075 (2023)
21. Li, Z., Xiao, J., Yang, L., Gu, Q.: Repq-vit: Scale reparameterization for post-training quantization of vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 17227–17236 (2023)
22. Lin, Y., Zhang, T., Sun, P., Li, Z., Zhou, S.: Fq-vit: Post-training quantization for fully quantized vision transformer. arXiv preprint arXiv:2111.13824 (2021)
23. Lin, Y., Zhang, T., Sun, P., Li, Z., Zhou, S.: Fq-vit: Post-training quantization for fully quantized vision transformer. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22. pp. 1173–1179 (2022)
24. Liu, Y., Yang, H., Dong, Z., Keutzer, K., Du, L., Zhang, S.: Noisyquant: Noisy bias-enhanced post-training activation quantization for vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 20321–20330 (2023)
25. Liu, Y., Yang, H., Dong, Z., Keutzer, K., Du, L., Zhang, S.: Noisyquant: Noisy bias-enhanced post-training activation quantization for vision transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 20321–20330 (June 2023)
26. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021)
27. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 10012–10022 (2021)
28. Liu, Z., Wang, Y., Han, K., Zhang, W., Ma, S., Gao, W.: Post-training quantization for vision transformer. Advances in Neural Information Processing Systems **34**, 28092–28103 (2021)
29. Lou, Q., Guo, F., Kim, M., Liu, L., Jiang, L.: Autoq: Automated kernel-wise neural network quantizations. In: ICLR (2020)
30. Nagel, M., Amjad, R.A., Baalen, M.V., Louizos, C., Blankevoort, T.: Up or down? adaptive rounding for post-training quantization. PMLR (2020)
31. Nagel, M., Amjad, R.A., Van Baalen, M., Louizos, C., Blankevoort, T.: Up or down? adaptive rounding for post-training quantization. In: International Conference on Machine Learning. pp. 7197–7206. PMLR (2020)
32. Nahshan, Y., Chmiel, B., Baskin, C., Zheltonozhskii, E., Banner, R., Bronstein, A.M., Mendelson, A.: Loss aware post-training quantization. Machine Learning **110**(11-12), 3245–3262 (2021)
33. Rastegari, M., Ordonez, V., Redmon, J., Farhadi, A.: XNOR-NET: Imagenet classification using binary convolutional neural networks. In: arXiv preprint arXiv:1603.05279 (2016)
34. Shi, H., Cheng, X., Mao, W., Wang, Z.: P^2-vit: Power-of-two post-training quantization and acceleration for fully quantized vision transformer. IEEE Transactions on Very Large Scale Integration (VLSI) Systems (2024)
35. Sutton, R.S., Barto, A.G.: Reinforcement learning: An introduction. MIT press (2018)

36. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International conference on machine learning. pp. 10347–10357. PMLR (2021)
37. Wang, K., Liu, Z., Lin, Y., Lin, J., Han, S.: Haq: Hardware-aware automated quantization with mixed precision. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 8612–8620 (2019)
38. Wang, K., Liu, Z., Lin, Y., Lin, J., Han, S.: HAQ: Hardware-aware automated quantization with mixed precision. In: CVPR (2019)
39. Wu, D., Tang, Q., Zhao, Y., Zhang, M., Fu, Y., Zhang, D.: Easyquant: Post-training quantization via scale optimization. arXiv preprint arXiv:2006.16669 (2020)
40. Wu, Z., Wang, S., Zhang, J., Chen, J., Wang, Y.: Fima-q: Post-training quantization for vision transformers by fisher information matrix approximation. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
41. Wu, Z., Zhang, J., Chen, J., Guo, J., Huang, D., Wang, Y.: Aphq-vit: Post-training quantization with average perturbation hessian based reconstruction for vision transformers. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2025)
42. Xiao, J., Li, Z., Yang, L., Gu, Q.: Patch-wise mixed-precision quantization of vision transformer. In: 2023 International Joint Conference on Neural Networks (IJCNN). pp. 1–7. IEEE (2023)
43. Xu, K., Wang, Z., Chen, C., Geng, X., Lin, J., Yang, X., Wu, M., Li, X., Lin, W.: Lpvit: Low-power semi-structured pruning for vision transformers. In: European Conference on Computer Vision. pp. 269–287. Springer (2024)
44. Xu, K., Li, Z., Wang, S., Zhang, X.: Ptmq: Post-training multi-bit quantization of neural networks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 38, pp. 16193–16201 (2024)
45. Yang, L., Jin, Q.: Fracbits: Mixed precision quantization via fractional bit-widths. arXiv preprint arXiv:2007.02017 (2020)
46. Yuan, Z., Xue, C., Chen, Y., Wu, Q., Sun, G.: Ptq4vit: Post-training quantization framework for vision transformers with twin uniform quantization. arXiv preprint arXiv:2111.12293 (2021)
47. Yuan, Z., Xue, C., Chen, Y., Wu, Q., Sun, G.: Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In: European conference on computer vision. pp. 191–207. Springer (2022)
48. Zhang, D., Yang, J., Ye, D., Hua, G.: Lq-nets: learned quantization for highly accurate and compact deep neural networks. In: ECCV (2018)
49. Zhao, R., Hu, Y., Dotzel, J., Sa, C.D., Zhang, Z.: Improving neural network quantization without retraining using outlier channel splitting. PMLR (2019)
50. Zhong, Y., Hu, J., Huang, Y., Zhang, Y., Ji, R.: Erq: Error reduction for post-training quantization of vision transformers. In: Proceedings of the International Conference on Machine Learning (ICML) (2024)
51. Zhou, A., Yao, A., Guo, Y., Xu, L., Chen, Y.: Incremental network quantization: Towards lossless cnns with low-precision weights. In: arXiv preprint arXiv:1702.03044 (2017)
52. Zhou, S., Wu, Y., Ni, Z., Zhou, X., Wen, H., Zou, Y.: Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. In: arXiv preprint arXiv:1606.06160 (2016)
53. Zhu, C., Han, S., Mao, H., Dally, W.: Ternary weight networks. In: ICLR (2017)