CPT-V: A Contrastive Approach to Post-Training Quantization of Vision Transformers

Natalia Frumkin
The University of Texas at Austin
nfrumkin@utexas.edu

Dibakar Gope
Arm Inc.
dibakar.gope@arm.com

Diana Marculescu
The University of Texas at Austin
dianam@utexas.edu

Abstract

When considering post-training quantization, prior work has typically focused on developing a mixed precision scheme or learning the best way to partition a network for quantization. In our work, CPT-V, we look at a general way to improve the accuracy of networks that have already been quantized, simply by perturbing the quantization scales. Borrowing the idea of contrastive loss from self-supervised learning, we find a robust way to jointly minimize a loss function using just $1,000$ *calibration images. In order to determine the best performing quantization scale, CPT-V contrasts the features of quantized and full precision models in a self-supervised fashion.*

Unlike traditional reconstruction-based loss functions, the use of a contrastive loss function not only rewards similarity between the quantized and full precision outputs but also helps in distinguishing the quantized output from other outputs within a given batch. In addition, in contrast to prior works, CPT-V proposes a block-wise evolutionary search to minimize a global contrastive loss objective, allowing for accuracy improvement of existing vision transformer (ViT) quantization schemes. For example, CPT-V improves the top-1 accuracy of a fully quantized ViT-Base by 10.30%, 0.78%, *and* 0.15% *for* 3*-bit,* 4*-bit, and* 8*-bit weight quantization levels. Extensive experiments on a variety of other ViT architectures further demonstrate its robustness in extreme quantization scenarios. Our code is available at* `<link>`.

1. Introduction

Quantization is a widespread technique for efficient neural network inference: reducing data precision from 32 bits to ≤ 8 bits is an effective approach to aggressively reduce model footprint and speed up computation. Network quantization can be used for on-device inference or in cloud scenarios where some accuracy can be sacrificed for an improvement in performance. We expect these devices to have improved hardware support for quantization in the future, where 8-bit, 4-bit, and even 3-bit acceleration is available for mobile devices and cloud applications. In these scenarios, full end-to-end quantization is advantageous because we can physically reduce storage and take full advantage of integer compute units. This paper focuses on post-training quantization of end-to-end neural networks, especially vision transfomers, where weights and activations of all layers are quantized.

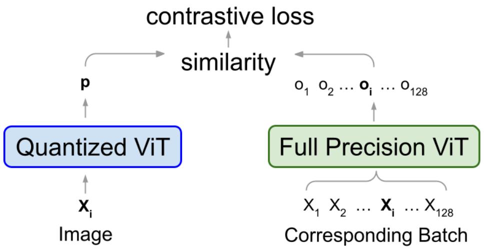
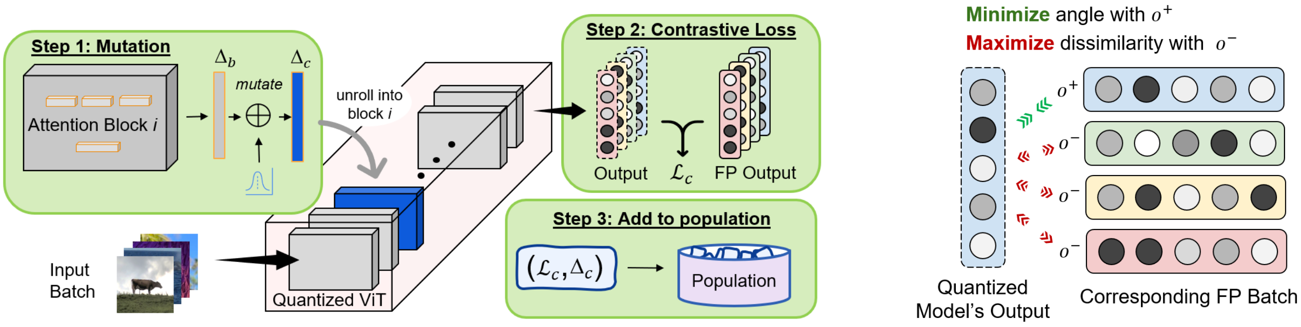


Figure 1. CPT-V uses contrastive loss to measure the similarity between a quantized prediction and the full precision predictions in the corresponding batch. Figure design is heavily inspired by MoCo [12].

In this setting, we develop a quantization framework that aims to improve accuracy while maintaining the same quantization scheme as the baseline. *Our framework, CPT-V, is the first work to use contrastive loss for neural network quantization*. In CPT-V, we show that a contrastive loss is effective in improving quantization performance, whereas traditional loss functions such as mean squared error, cosine similarity, and the KL divergence tend to overfit the calibration dataset. Using this advantage, we use block-wise evolutionary search to minimize a *single objective global loss*. Ultimately, we find contrastive loss to be very effective in learning quantized representations and robust against representation collapse. The contributions of this paper are three-fold:

1. **Contrastive Loss Function.** We use a contrastive loss to measure the similarity between a quantized prediction and the corresponding batch of full precision pre-



(a) A single cycle of block-wise evolutionary search.

(b) Visualization of Contrastive Loss (Step 2)

Figure 2. An overview of CPT-V. On the left, we show one cycle completed on a single block. Each block has C cycles of evolutionary search, and we perform K passes over all blocks. On the right, we provide intuition for contrastive loss (Step 2), where we encourage similarity between the quantized and corresponding predictions while simultaneously maximizing dissimilarity between unlike predictions.

dictions. The contrastive loss minimizes the distance between the quantized and corresponding full precision predictions while simultaneously maximizing the dissimilarity with other predictions in the batch.

2. **Block-wise Evolutionary Search.** To reduce search complexity, we employ an intuitive block-wise evolutionary search algorithm to adjust scales one block at a time.

3. **Global Search for Quantization Scales.** Our method finds the optimal set of quantization scales that globally minimizes a contrastive loss. By perturbing quantization scales, we find a quantized model that improves accuracy over the baseline method.

2. Related Work

In this paper, we discuss techniques for *post-training quantization* (PTQ). We assume no access to the initial training dataset and use a small, unlabeled calibration dataset ($1,000$ images) to help guide quantization. Typically, the calibration dataset is used for activation quantization to estimate their distributions after each layer and minimize a layer-wise loss [2, 13, 29]. More recently, techniques involving knowledge distillation [6] and loss-aware optimization [25] use the calibration dataset to learn the quantization schemes of all layers jointly. However, these techniques are often victims of overfitting – a set of $1,000$ images for a $1,000$-class dataset like ImageNet is not properly representative of the entire space of images. As a result, while using an MSE objective [15] or a Hessian-Aware objective [16] may achieve reasonable results in a limited setting, they are unlikely to generalize to larger datasets and more extreme quantization where the loss landscape is highly non-smooth [1].

Alternatively, some prior works focus on layer-wise or block-wise quantization, where each layer or block is quantized separately. To learn the best quantization parameters, this work typically employs statistical analysis or a loss with respect to the full precision feature maps. These techniques are typically generalizable to many different architectures; however, quantizing layer-by-layer may not be the best approach since one layer's quantization scheme can influence the information entropy of another layer [20].

Considering the current drawbacks of both global optimization approaches and layer-wise quantization, CPT-V attempts to learn quantization scales that are globally optimal while avoiding the overfitting problem that is common in methods that jointly optimize all quantization layers.

Vision Transformers (ViTs) have shown impressive results on the latest computer vision tasks [28, 29]. However unlike CNNs, ViT layers have extremely unbalanced distributions making quantization choice critical for achieving good predictive performance. To address this, PTQ-for-ViT [23], learns a quantization scheme for each layer by choosing (1) the bit-width of a multi-head attention module using an attention map ranking loss and (2) the bit-width of the MLP layer using cosine similarity. This work focuses on bit-width choice, whereas other works [8,31] focus on optimization of quantization scales. PTQ4ViT [31] employ twin-uniform quantization to improve the quantization of post-GeLU/Softmax activations and use Hessian-guided global optimization to learn the quantization scales of all linear layers. As previously stated, a Hessian-guided metric may perform well in a smooth loss landscape, such as 8-bit quantization with a well-designed dataset, but is likely to degrade in lower bit quantization schemes. Another technique, PSAQ-ViT-V2 [17], uses a student-teacher minmax game to minimize the KL divergence between the full precision and quantized models. See Tab. 1 for a breakdown of closely related prior works.

Our method's baseline, FQ-ViT [21], is a layer-wise quantization method which incorporates Log2 quantization and an integer softmax function for end-to-end 8-bit

	Mr. BiQ‡ [15]	PTQ4ViT [31]	PSAQ-ViT [6]	PSAQ-ViT-V2 [17]	FQ-ViT [21]	CPT-V [ours]
Weight Quantization	Multi-Bit	Uniform (range: MinMax)				Initial range: MinMax
Activation Quantization	Uniform	Twin Uniform	Asymmetric Uniform		Log2	
Calibration Set	1K+ QAT	128	Data-Free		1K	
Loss Type	Least-Squares	Hessian-based	-	KL	-	Contrastive
Optimization Level	Layer-wise	Global	Layer-wise			Global
Robustness to Bit-Width	2,3,4,6	6,8	8	8	8	3,4,8
Fully Quantized	-	✓	✓	-	✓	✓
Optimized Quantization Scale	-	-	-	-	-	✓

‡ QAT-based initialization.

Table 1. Comparison with Related Work. Optimization level refers to how the quantization parameters are being learned, either with a global or a layer-wise objective. We include robustness to bit-width to illustrate that each method is geared towards a different weight quantization scheme. CPT-V initializes weight quantization with MinMax, and optimizes the quantization scale to obtain the best accuracy.

ViT quantization. In our method, we use FQ-ViT as initialization for a contrastive loss-based global optimization scheme.

Quantization-Aware Training (QAT) has shown impressive results on vision transformers [18–20, 30], yet these methods consider training with the entire dataset rather than an unlabeled calibration dataset. We provide a comparison to PTQ methods using QAT-based methods as an "oracle."

Currently, there is limited work that looks at combining quantization and self-supervised learning. These works [4, 10] use quantization-aware training in conjunction with a self-supervised learning scheme. They claim that quantization allows for regularization during training and is complementary to using augmentation for self-supervised learning. Both contributions [4, 10] combine a quantization-based loss and a contrastive loss into a joint optimization scheme, however, they do not use it in the same way we do in this paper. They apply a contrastive loss with respect to only full precision predictions, whereas we apply the contrastive loss as a reconstruction loss for the quantized predictions (see Fig. 2b).

3. The CPT-V Framework

CPT-V is a post-training technique for end-to-end quantization. We quantize both the weights and activations for all layers, as done in [21]. In particular, we quantize all activations, including those around the Softmax and LayerNorm layers. To improve the accuracy of an existing quantized model, we use a block-wise evolutionary search algorithm to search for the set of scales which are globally optimal.

3.1. Uniform, End-to-End Quantization

We consider uniform quantization, where the quantization range is between α and β, initialized using MinMax [14] for weights and Log2 [3] for the activations. Uniform quantization is formally defined as:

$$Q(\mathbf{x}, \delta, \alpha, \beta) = clip(round(\frac{\mathbf{x}}{\delta}), \alpha, \beta) \tag{1}$$

where $\mathbf{x}$ is a full-precision vector and δ is the quantization scale. In our framework, we learn these initial quantization parameters using a fast, layer-wise framework such as FQ-ViT [21], and then use evolutionary search to adjust the scales of each attention block.

3.2. Global Search for Quantization Scales

When considering *global optimization* of all quantization scales, the dimensionality of the search space becomes extremely high. Each weight matrix and attention map may be assigned many quantization parameters (*e.g.*, token-wise or group-wise quantization), and the number of scales may grow to over $10,000$ per model. In this case, the search space of all quantization scales is typically partitioned in a block-wise or layer-wise fashion. Similar to [16], we partition our search space in a block-wise manner, but evaluate each block using a global fitness metric based on a contrastive loss (see Sec. 3.3 for more details).

Q: Why not use gradient descent?
The quantization function is non-differentiable. Normal quantization-aware training typically circumvents this by training the weights as opposed to the quantization scales and using a straight-through estimator. In order to adjust the scales directly, we choose to use evolutionary search since it converges fast enough in this setup.

In order to find the optimal scales for all layers, we employ block-wise evolutionary search, as shown in Algorithm 1. We represent all scales in an attention block as a stacked 1-D vector $\mathbf{\Delta}_b$, and perturb this vector during search to find the best scales with respect to the fitness function (evaluation metric). Once we have completed our

Algorithm 1 Block-wise Evolutionary Search

```
Input: Calibration Dataset $D_C$
    Quantized Model $M_Q$, Full Precision Model $M_F$
    Number of passes $K$, cycles $C$
    Population size $P$, Sample size $S$
▷ traverse all attention blocks
for pass in 0 : $K$ do
    for $b$ in AttentionBlocks($M_Q$) do
        ▷ begin search for attention block $b$
        pop ← [ ]
        while | pop | < $P$ do                    ▷ initialize population
            fitness ← Fitness($b$, $M_Q$, $M_F$, $D_C$)
            pop.insert(($\boldsymbol{\Delta}_{\texttt{b}}$, fitness))
        for cycle in 0 : $C$ do
            ▷ choose parent scales from sampled set
            samples ← [ ]
            while | samples | < $S$ do
                samples ← RandomElement(pop)
            parent ← BestFitness(samples)

            ▷ evaluate child and add to population
            $\boldsymbol{\Delta}_{\texttt{child}}$ ← Mutate($\boldsymbol{\Delta}_{\texttt{parent}}$)
            fitness ← Fitness($D_C$, $M_Q$, $M_F$ )
            pop.insert( ($\boldsymbol{\Delta}_{\texttt{child}}$, fitness ) )
            ▷ remove candidate with lowest fitness
            pop.DeleteDead()
        $\boldsymbol{\Delta}_{\texttt{b}}$ ← BestFitness(pop)
return $M_Q$                              ▷ model with updated scales
```

search for one block, we go on to the next block and repeat this process.

For each attention block, we apply a small evolutionary search procedure to learn the best quantization scales within a neighborhood of the current scales. This sub-problem consists of C cycles of evolutionary search, where the scales are mutated once per cycle in the following manner:

$$\boldsymbol{\Delta}_{\texttt{child}} \sim \mathcal{U}(\boldsymbol{\Delta}_{\texttt{parent}}, -\gamma, \gamma) \tag{2}$$

Our mutation function, Eq. (2), generates a new child scale, $\boldsymbol{\Delta}_{\texttt{child}}$, from a uniform distribution parameterized by γ. This child scale is assigned to the current block b and evaluated using our fitness function.

We apply this small evolutionary search sub-problem to a given block b and traverse through all blocks to jointly learn all quantization parameters. P passes are performed over all attention blocks (*i.e.*, K passes means search is applied to each block K times). Through trial and error, we find that $P >> C$ is most effective, demonstrating that we want many smaller evolutionary search problems rather than fewer large search problems.

3.3. Contrastive Loss

Contrastive loss is commonly used in self-supervised learning to prevent representation collapse [11] and to encourage discrimination between a target representation and a set of negative examples. We find a contrastive loss to be very effective in preventing the collapse of a *quantized network's* representation in the same way that it is used to develop richer representations in a self-supervised setting. Inspired by [5], we use the infoNCE [26] loss:

$$\mathcal{L}_c = -\log \frac{\exp(p \cdot o^+/\tau)}{\exp(p \cdot o^+/\tau) + \sum_{o^-} \exp(p \cdot o^-/\tau)} \tag{3}$$

where p is the prediction of the quantized model, o^+ is the corresponding prediction of the full precision model, and o^- is a prediction of another image in the same batch. We use infoNCE in a novel way: to improve the reconstruction of o^+ while enforcing that the quantized model prediction is dissimilar from other images in a batch.

Algorithm 2 Fitness Function

```
Input: calibration dataset $D_C$
    quantized model $M_Q$, full precision model $M_F$
for batch in $D_C$ do
    p = $M_Q$(batch)
    o = $M_F$(batch)
    score += ContrastiveLoss(p, o)        ▷ Eq. (3)
return score / size($D_C$)
```

When evaluating a block's fitness, we take the average contrastive loss with respect to the calibration dataset as shown in Algorithm 2. We emphasize that fitness is a *global evaluation metric* to evaluate how well $\boldsymbol{\Delta}_b$ affects the final prediction, and is not evaluated on any intermediate feature map individually.

4. Results

In the following section, we present results on a variety of vision transformers and show the consistency of our method under standard 8-bit quantization and in extreme quantization schemes (3-bit and 4-bit weights). We present results for end-to-end quantization, where all weights and activations are quantized. For all experiments, we consider 8-bit activation quantization and vary weight quantization.

4.1. Setup

We conduct experiments using ImageNet (ILSVRC2012) and evaluate our post-training techniques on a variety of vision transformer model families. The calibration dataset is $1,000$ randomly sampled images from the ImageNet training set (without labels). We note that this is a different setup from quantization-aware training, where the entire training dataset is used. The initial quantized model (M_Q in Algorithm 1) is generated using FQ-ViT, and our method does not adjust any quantization settings. It only perturbs the quantization scales. FQ-ViT is an end-to-end quantization framework that uses MinMax [14] for weight quantization and Log2 [3] for activation quantization. We refer to FQ-ViT and our code for all other quantization settings.

For block-wise evolutionary search, we specify all search settings in Tab. 2 below.

passes	K	10
population size	P	15
cycles	C	3
samples	S	10
mutation range	γ	$10^{-3}/10^{-4}$

Table 2. Block-wise evolutionary search settings. The mutation range is 10^{-3} for 8W8A, and 10^{-4} for 4W8A and 3W8A.

4.2. 8-bit Quantization

We compare our standard 8-bit quantization with state-of-the-art methods in Tab. 3. We improve over existing *end-to-end* quantization techniques by at least 0.1%, 1.2%, and 0.15% for DeiT-Small, DeiT-Base, and ViT-Base, respectively.

8-bit weights, 8-bit activations (8W8A)				
Method	DeiT-T	DeiT-S	DeiT-B	ViT-B
PSAQ-ViT	71.56	76.92	79.10	37.36
PTQ4ViT	-	79.47	81.48	84.25
FQ-ViT	71.61	79.17	81.20	83.31
PSAQ-ViT-V2†	**72.17**	79.56	81.52	-
CPT-V (ours)	71.63	**79.57**	**82.67**	**84.40**

† Does not quantize Softmax/GELU layers

Table 3. Top-1 Accuracy on ImageNet using 8W8A quantization. -T, -S, & -B refer to Tiny, Small, and Base models respectively.

We also compare with PSAQ-ViT-V2 [17] and find that it outperforms our method by 0.5% for DeiT-Tiny. However, PSAQ-ViT-V2 is not a full end-to-end quantization method since it *does not quantize the activations* following the Softmax and GELU layers. These activations are typically very sensitive to quantization and are often maintained at full precision for this purpose. In our work, we consider end-to-end quantization, so we are forced to quantize the post-Softmax/GELU activations. We leave it to future work to apply our technique on top of PSAQ-ViT-V2, but we expect similar improvements to what we achieved with FQ-ViT.

4.3. 4-bit Quantization

Moving from 8-bit to 4-bit weight quantization, we see an accuracy degradation of about $2-5\%$ across all models. In Tab. 4, we find that our method performs similarly to what is shown for 8-bit quantization. In particular, we still see improvement for DeiT-Small, DeiT-Base, and ViT-Base, but now the top-1 accuracy improvement is 0.9%, 0.65%, and 0.8%, respectively.

4-bit weights, 8-bit activations (4W8A)				
Method	DeiT-T	DeiT-S	DeiT-B	ViT-B
PSAQ-ViT	65.57	73.23	77.05	25.34
PTQ4ViT	-	-	64.39	-
FQ-ViT	66.91	76.93	79.99	78.73
PSAQ-ViT-V2†	**68.61**	76.36	79.49	-
CPT-V (ours)	67.29	**77.06**	**80.15**	**79.50**

† Does not quantize Softmax/GELU layers

Table 4. Top-1 Accuracy on ImageNet using 4W8A quantization.

4.4. 3-bit Quantization

We report 3-bit quantization results in Tab. 5 to show that CPT-V extends to more extreme quantization scenarios. In particular, CPT-V improves accuracy over FQ-ViT by 10.3% for ViT-Base and 3.7% for DeiT-Tiny. Since other PTQ papers do not consider 3-bit quantization for ViTs, we compare with Mr.BiQ [15], a post-QAT method (see Sec. 5.6 for more details). By reducing the precision from 32 to 3 bits, we achieve a >10X reduction in memory footprint while still maintaining a reasonable accuracy for DeiT-Base. While DeiT-Tiny and DeiT-Small achieve low accuracy in this regime, their improvement may be drastically improved with other techniques such as OMSE [7], bias correction [2], or group-wise quantization [27].

3-bit weights, 8-bit activations (3W8A)				
Method	DeiT-T	DeiT-S	DeiT-B	ViT-B
FQ-ViT	35.79	60.58	72.11	55.33
CPT-V (ours)	**39.45**	61.16	72.41	65.63
Mr. BiQ‡	-	**78.09**	**81.10**	**77.41**

‡ QAT-based initialization.

Table 5. Top-1 Accuracy on ImageNet using 3W8A quantization.

4.5. Extending to Swin & LeViT Models

We run our experiments on additional model families to ensure that our method is applicable to different types of transformer blocks. Swin transformers [22] have the same macro-architecture as DeiT and ViT models, with the exception that the swin transformer block is a windowed attention. We see results for 4-bit Swin transformers in Tab. 6.

Prior quantization techniques do not consider LeViT models, a family of vision transformers that improve the

Method	Swin-T	Swin-S	Swin-B
PSAQ-ViT	71.79	75.14	-
PSAQ-ViT-V2[†]	76.28	78.86	-
FQ-ViT	**80.73**	82.13	82.73
CPT-V (ours)	80.43	**82.63**	**83.07**

[†] Does not quantize Softmax/GELU layers

Table 6. Top-1 Accuracy for Swin Models using 4W8A.

inference speed of existing transformers. Using a longer convolutional backbone, they can achieve similar results to classic vision transformers while also reducing the complexity of the transformer blocks in favor of ResNet-like stages. We include the LeViT family in our experiments to illustrate how our method can be extended beyond the standard block size. We can see 4W8A results for the LeViT model family in Tab. 7. Across the board, we see a significant improvement in LeViT quantization as compared to FQ-ViT, our baseline.

Model	FQ-ViT	CPT-V (ours)
LeViT-128S	14.90	**29.20**
LeViT-192	17.00	**30.37**
LeViT-256	61.33	**64.57**
LeViT-384	64.60	**69.50**

Table 7. Top-1 Accuracy for LeViT models using 4W8A.

In fairness, we have not applied any other techniques to boost LeViT's accuracy (doing so may inflate our method's improvement), so we leave it to future work to incorporate other quantization techniques on top of our framework.

5. Analysis

In the section below, we provide a discussion on runtime, various ablations, and explainability using attention maps. Our goal is to motivate the need for a fast, yet reliable technique to improve end-to-end quantization of ViTs.

5.1. Loss Function Choice

We compare contrastive loss with other common loss functions in Fig. 3. We find mean-squared error (MSE) to be equally (if not more) effective in the initial iterations of CPT-V. However, as the number of passes grows, MSE does not perform as well as the contrastive loss. Both cosine similarity and the Kullback–Leibler divergence (KL) fail to improve performance as the number of iterations increases. We postulate that the poor performance of these traditional loss functions is due to overfitting to the calibration dataset. On the other hand, contrastive loss is naturally regularized by the negative samples in the batch, allowing for the contrastive loss to preserve the quantization parameters that help discriminate between classes.

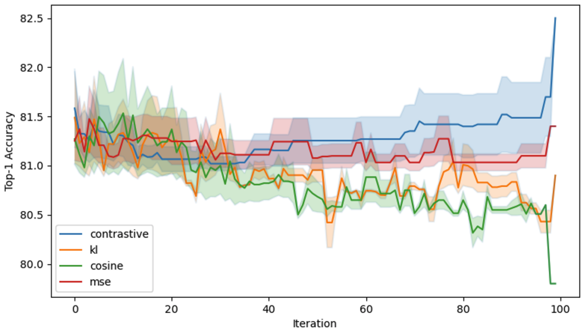


Figure 3. Comparing performance of CPT-V for the four loss functions. We find that Contrastive loss prevents overfitting to the calibration dataset, whereas all other loss functions cannot improve accuracy beyond the initialized quantization scheme.

5.2. Layer-wise Weight Distributions

In Fig. 4, we compare the weight distributions of the full precision, FQ-ViT, and CPT-V quantization schemes. CPT-V's quantized weight distribution is similar to FQ-ViT, yet CPT-V has a 0.8% improvement over FQ-ViT (note: $\%$ improvement is for this run only, Tab. 3 is averaged over three seeds).

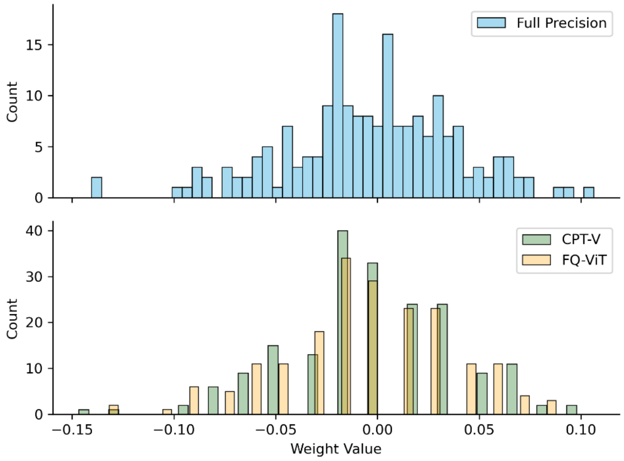


Figure 4. The weight distribution of the projection layer for attention block #1 of DeiT-Tiny. The top plot shows the full precision weight distribution and the bottom plot shows the 8-bit weights using both FQ-ViT and CPT-V.

Interestingly, this graph illustrates that a small perturbation in the scales can significantly improve performance. This is consistent with the conclusion in AdaRound [24], where they show that a perturbation in weights may yield no difference in quantized outputs. AdaRound claims that round-to-nearest is not the most effective rounding scheme since it may round away weights which are important for deciphering classes. By adjusting quantization scales, we can overcome the aforementioned rounding issue by en-

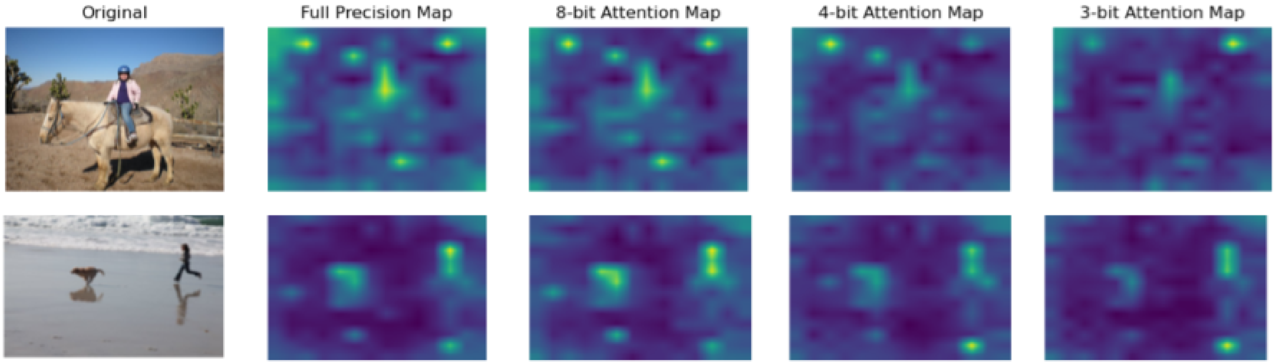


Figure 5. Attention maps for different quantization levels. CPT-V's quantized models preserve the spacial locality of the full precision feature map. As the quantization level becomes more extreme, the attention map becomes subject to decreased resolution.

couraging dissimilarity between two unlike predictions.

Fig. 4 is just one block, and we refer to the supplementary materials for a more complete discussion of how these perturbations change across different layers.

5.3. Impact on Attention Maps

We find that CPT-V preserves the spatial integrity of the full precision feature maps even as quantization forces discretization of the attention mechanism. In Fig. 5, as quantization becomes more severe from 8-bit to 3-bit, the resolution of the feature map degrades, as is expected when only a finite number of values can be expressed in the quantized scheme. This attention map visualization is averaged over all blocks, and serves as qualitative inspection of how the network's attention mechanism is performing. All in all, Fig. 5 provides confidence that CPT-V's quantized attention maps learn reasonable representations of the original full precision network.

5.4. Ablation: Passes vs. Cycles

In Fig. 6, we ablate the number of passes, K, from 1 to 35. As we can see, a majority of the accuracy improvement occurs in the first 10 passes, so we choose $K = 10$ for all experiments above. This allows for our method to run in less than one hour. However, we note that an additional accuracy boost may be enjoyed with more passes.

We also ablate the number of cycles, C, to determine how many mutations should occur per block (one cycle is shown in Fig. 2a). We use $C = 3$ even though we see $C = 7$ is optimal in our ablation study. In practice, we find that the choice of C is random seed and model dependent. We find that for some runs, the best choice is simply 1 cycle, but in others it is 3, 5 or 7. Ultimately, we choose $C = 3$ for consistency across experiments.

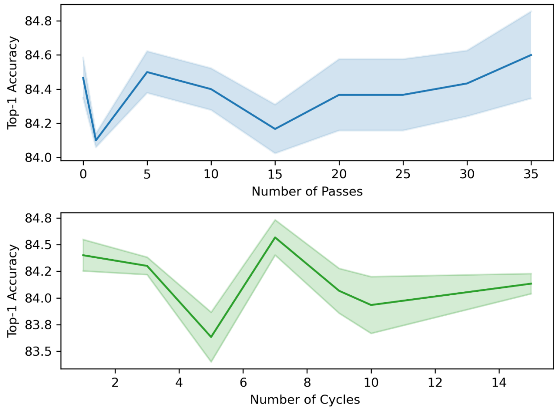


Figure 6. Ablation on number of cycles and passes

5.5. Runtime

We run our method on an Nvidia A100-PCIE-40GB Tensor Core GPU and find that all experiments take less than one hour to run. The average runtime is shown in Tab. 8. We use Pytorch 1.9.1, built with the CUDA 11.1 toolkit.

	DeiT-T	DeiT-S	DeiT-B	ViT-B
Runtime (mins)	41.5	46.3	41.6	43.2

Table 8. CPT-V Runtime (in minutes) on one Nvidia A100 GPU

In Fig. 7, we make a comparison with our runtime and other ViT quantization methods and demonstrate that our method achieves superior accuracy on ViT-Base network while being faster than QAT and post-QAT methods. Fig. 7 only captures the Top-1 Accuracy of ViT-Base (or, alternatively, DeiT-base if ViT-Base is unavailable). We note that there would be a different plot if we used DeiT-Tiny, since PSAQ-ViT-V2 outperforms CPT-V in this scenario. We refer to the supplementary material for a wider discussion on how this plot changes with different models.

5.6. Comparison with QAT Methods

In Fig. 7, methods are categorized as either post-training quantization (PTQ), quantization-aware training (QAT), or post-QAT. QAT methods have only recently shown results for vision transformers, so the results using Q-ViT and TerViT are only for 3/4-bit schemes and not for 8-bit (which is used by many PTQ methods).

We classify Mr. BiQ [15] as a post-QAT technique, since it reports top-1 accuracy using LSQ [9] as an initialization step to learn the best activation quantization. LSQ is a QAT technique, and potentially improves results by $1-5\%$.

Most of these methods are run on a single Nvidia A100 GPU, but some were not open-sourced at the time of this publication. In particular, TerViT [30] and LSQ [9] are estimated using LSQ's open-source ResNet-50 QAT scheme, and Mr.BiQ's runtime is reported on a Nvidia V100. There was no reported runtime for PSAQ-ViT-V2, so we estimate it to be 60 minutes based on PSAQ-ViT and our best guess at the runtime of the additional steps.

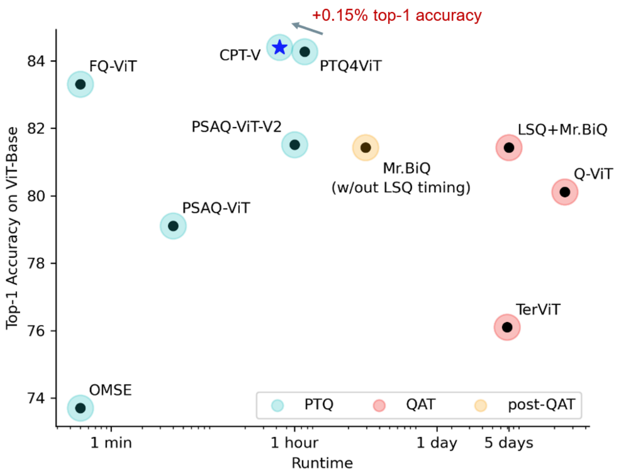


Figure 7. Runtime of various quantization methods for ViTs. We consider PTQ, QAT, and post-QAT methods in our analysis.

5.7. On Variation across Random Seeds

In Fig. 8, we show the performance of CPT-V compared to the baseline method, FQ-ViT. Across twelve random seeds, ten runs improve performance over FQ-ViT, and three result in top-1 accuracy that is superior to the full precision model. This does not typically happen for post-training methods since no training is involved. However, we find that CPT-V can traverse through the optimization space and happen upon quantization parameters that can improve performance drastically. Overall, CPT-V has an 83% (10/12) chance of improving on FQ-ViT for ViT-Base.

As with many advanced post-training methods, the random seed is a significant factor in determining the final quantization accuracy. The random seed dictates which images are chosen for the calibration dataset, as well as the way the dataset is generated. We postulate that the performance of CPT-V can be improved by choosing the proper set of images for the calibration dataset (*i.e.*, those that are representative of the test set). For fair comparison, our experiments allow for all types of calibration sets to be generated for quantization, and we attribute the poor accuracy in seeds 4 and 5 to the poor choice of calibration set.

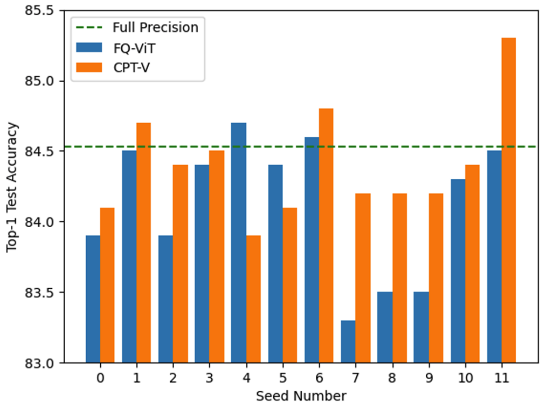


Figure 8. Comparing performance across 12 random seeds for 8W8A ViT-Base. 10/12 runs improve over the initial FQ-ViT quantization.

5.8. Generalizability

Our method requires a pre-quantized model, M_Q, and adjusts the models scales to improve accuracy. We only require that the model can be abstracted into blocks, with each block having a set of quantization scales. This makes our method readily applicable to other types of models such as CNNs, LSTMs, Graph Neural Networks, and many others. We focus exclusively on ViTs in this work, since we find them to be very successful for image classification tasks, and leave to future work the extension of this framework to other block-wise separable networks.

6. Conclusion

We propose CPT-V, a quantization method that uses a contrastive loss to jointly optimize the quantization scales of all attention blocks. Using block-wise evolutionary search, CPT-V improves the accuracy of an existing quantized model by minimizing a global fitness function. Using our method, we improve top-1 accuracy for 8-bit, 4-bit, and 3-bit quantization levels when compared to existing state-of-the-art ViT quantization techniques. Finally, CPT-V benefits from generality – we can apply it to many types of blocks with different quantization schemes. As long as there are quantization scales, our method is able to optimize using block-wise evolutionary search.

7. Acknowledgements

Partial work done during a summer internship at Arm Ltd. A special thank you to Jesse Beu for overseeing this internship project, and to Feng Liang for helpful discussion.

References

[1] Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jing Jin, Xin Jiang, Qun Liu, Michael Lyu, and Irwin King. Binarybert: Pushing the limit of bert quantization. *arXiv preprint arXiv:2012.15701*, 2020.

[2] Ron Banner, Yury Nahshan, and Daniel Soudry. Post training 4-bit quantization of convolutional networks for rapid-deployment. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] Jingyong Cai, Masashi Takemoto, and Hironori Nakajo. A deep look into logarithmic quantization of model parameters in neural networks. In *Proceedings of the 10th International Conference on Advances in Information Technology*, pages 1–8, 2018.

[4] Yun-Hao Cao, Peiqin Sun, Yechang Huang, Jianxin Wu, and Shuchang Zhou. Synergistic self-supervised and quantization learning. In *European Conference on Computer Vision*, pages 587–604. Springer, 2022.

[5] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021.

[6] Yoojin Choi, Jihwan Choi, Mostafa El-Khamy, and Jungwon Lee. Data-free network quantization with adversarial knowledge distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 710–711, 2020.

[7] Yoni Choukroun, Eli Kravchik, Fan Yang, and Pavel Kisilev. Low-bit quantization of neural networks for efficient inference. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, pages 3009–3018. IEEE, 2019.

[8] Yifu Ding, Haotong Qin, Qinghua Yan, Zhenhua Chai, Junjie Liu, Xiaolin Wei, and Xianglong Liu. Towards accurate post-training quantization for vision transformer. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 5380–5388, 2022.

[9] Steven K Esser, Jeffrey L McKinstry, Deepika Bablani, Rathinakumar Appuswamy, and Dharmendra S Modha. Learned step size quantization. *arXiv preprint arXiv:1902.08153*, 2019.

[10] Yonggan Fu, Qixuan Yu, Meng Li, Xu Ouyang, Vikas Chandra, and Yingyan Lin. Contrastive quant: quantization makes stronger contrastive learning. In *Proceedings of the 59th ACM/IEEE Design Automation Conference*, pages 205–210, 2022.

[11] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[12] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[13] Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. Accurate post training quantization with small calibration sets. In *International Conference on Machine Learning*, pages 4466–4475. PMLR, 2021.

[14] Benoit Jacob, Skirmantas Kligys, Bo Chen, Menglong Zhu, Matthew Tang, Andrew Howard, Hartwig Adam, and Dmitry Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713, 2018.

[15] Yongkweon Jeon, Chungman Lee, Eulrang Cho, and Yeonju Ro. Mr. biq: Post-training non-uniform quantization based on minimizing the reconstruction error. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12329–12338, 2022.

[16] 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. *arXiv preprint arXiv:2102.05426*, 2021.

[17] Zhikai Li, Mengjuan Chen, Junrui Xiao, and Qingyi Gu. Psaq-vit v2: Towards accurate and general data-free quantization for vision transformers. *arXiv preprint arXiv:2209.05687*, 2022.

[18] Zhikai Li and Qingyi Gu. I-vit: Integer-only quantization for efficient vision transformer inference. *arXiv preprint arXiv:2207.01405*, 2022.

[19] Zhengang Li, Mengshu Sun, Alec Lu, Haoyu Ma, Geng Yuan, Yanyue Xie, Hao Tang, Yanyu Li, Miriam Leeser, Zhangyang Wang, et al. Auto-vit-acc: An fpga-aware automatic acceleration framework for vision transformer with mixed-scheme quantization. *arXiv preprint arXiv:2208.05163*, 2022.

[20] Zhexin Li, Tong Yang, Peisong Wang, and Jian Cheng. Q-vit: Fully differentiable quantization for vision transformer. *arXiv preprint arXiv:2201.07703*, 2022.

[21] Yang Lin, Tianyu Zhang, Peiqin Sun, Zheng Li, and Shuchang Zhou. Fq-vit: Post-training quantization for fully quantized vision transformer. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 1173–1179, 2022.

[22] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[23] Zhenhua Liu, Yunhe Wang, Kai Han, Wei Zhang, Siwei Ma, and Wen Gao. Post-training quantization for vision transformer. *Advances in Neural Information Processing Systems*, 34:28092–28103, 2021.

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

[25] 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.

[26] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[27] Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Q-bert: Hessian based ultra low precision quantization of bert. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8815–8821, 2020.

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

[29] Di Wu, Qi Tang, Yongle Zhao, Ming Zhang, Ying Fu, and Debing Zhang. Easyquant: Post-training quantization via scale optimization. *arXiv preprint arXiv:2006.16669*, 2020.

[30] Sheng Xu, Yanjing Li, Teli Ma, Bohan Zeng, Baochang Zhang, Peng Gao, and Jinhu Lu. Tervit: An efficient ternary vision transformer. *arXiv preprint arXiv:2201.08050*, 2022.

[31] Zhihang Yuan, Chenhao Xue, Yiqi Chen, Qiang Wu, and Guangyu Sun. Ptq4vit: Post-training quantization for vision transformers with twin uniform quantization. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XII*, pages 191–207. Springer Nature Switzerland Cham, 2022.