# Post-training Quantization without BN Statistics: A Data Free Approach

Akash Chavan
*Department of Computer Science and Engineering*
Oakland University
Rochester, MI, USA
akashchavan@oakland.edu

Sumit Kumar Jha
*Department of Computer and Information Science and Engineering*
University of Florida
Gainesville, FL, USA
sumit.jha@ufl.edu

Sunny Raj
*Department of Computer Science and Engineering*
Oakland University
Rochester, MI, USA
raj@oakland.edu

***Abstract*—Post-training quantization (PTQ) without access to real data is enabling efficient model optimization and deployment in scenarios where privacy or proprietary constraints restrict the use of original datasets. Traditional data free quantization methods rely on Batch Normalization (BN) statistics from the trained full-precision model to generate a calibration dataset for quantization. However, this reliance on BN statistics limits their applicability to deep neural networks (DNNs) without BN layers. In this paper, we propose a calibration dataset generation algorithm that is agnostic to BN statistics, leveraging just the backpropagation to create synthetic images for PTQ. We also demonstrate that it is not necessary to include an image for every target category in the calibration dataset to get the representative activation ranges for quantization. Extensive experiments with both large and lightweight models on large-scale image classification tasks demonstrate that our method consistently improves quantization performance across various DNN architectures, especially in low-bit settings. Notably, in 4-bit quantization, we achieve an improvement of 3.31% in top-1 accuracy for the ResNet18 model and 3.82% for the InceptionV3 model compared to the state-of-the-art (SOTA) DSG method. Importantly, we use very few synthetic images for quantization compared to other methods.**

***Index Terms*—deep neural networks, batch normalization, quantization aware training, post-training quantization.**

## I. Introduction

Deep neural networks (DNNs) have achieved remarkable success in applications such as image classification [1], object detection [2], robotics [3], and autonomous driving [4]. However, deploying these networks on resource-constrained devices remains a considerable challenge due to their substantial memory requirements and intensive computational demands [5]. Quantization, which converts the floating-point values of weights and/or activations to integers, is a favored method to address these challenges by significantly reducing model size and improving computation [6].

Quantization methods are generally categorized into Quantization-Aware Training (QAT) and Post-Training Quantization (PTQ). While QAT can achieve higher accuracy by incorporating quantization into the training process, it is computationally intensive and time-consuming [7]. PTQ applies quantization to a pre-trained model, but one of its main challenges is determining the activation ranges, which often requires a small calibration dataset [8]. When real data is unavailable due to privacy or proprietary constraints, data-free techniques are used. Existing generative approaches create calibration data by aligning its distribution with the Batch Normalization (BN) statistics of the full-precision model [9], [10]. However, this reliance on BN statistics limits their applicability to neural networks without BN layers.

In our work, we demonstrate that it is not necessary to depend on BN statistics to generate an effective calibration dataset. Additionally, we show that contrary to common practice [11], optimal performance can be achieved without including an image from each target class; a small number of images can suffice. The main contributions of this paper are:

- We propose a method to generate synthetic data using the full-precision model agnostic to BN statistics, making our approach applicable to any model architecture.
- we experimentally demonstrate that selecting only a few target classes is sufficient to create an effective calibration dataset, which in turn reduces PTQ time.
- Through extensive PTQ comparisons, we show that our method significantly outperforms existing generative quantization methods, especially in low-bit settings where we improve top-1 accuracy by over 3.31% for ResNet18 and 3.58% for InceptionV3 models compared to the SOTA DSG method.

## II. Related Work

In this section, we categorize existing research into two main methodologies: QAT and PTQ.

### A. Quantization Aware Training

QAT integrates quantization into the training phase, allowing models to adapt to quantization noise and resulting in higher accuracy, though it demands more computational resources and can be challenging to implement [7]. This approach involves low-precision computations during the forward pass while maintaining standard backpropagation [6]. Several QAT techniques have been explored, such as Binarized Neural Networks (BNNs) which constrain weights and activations to +1 or -1 [12], DoReFa-Net which uses low bit-width weights, activations, and gradients [13], and XNOR-Net which enables efficient binary CNNs [14]. Other work has

1946-0759/25/$31.00 ©2025 IEEE
DOI 10.1109/ICMLA66185.2025.00116

shown that lightweight networks often require QAT to reach baseline accuracy [15], though sometimes fine-tuning for just one epoch suffices [16]. Methods like PACT further improve accuracy by dynamically adjusting clipping values to minimize quantization error [17].

### *B. Post-Training Quantization*

PTQ reduces the memory and computational requirements of a fully trained model using a small calibration dataset. Determining activation ranges is a key part of this process. Some methods address this by analytically finding activation clipping ranges [8], while others focus on optimizing the quantized values. Techniques like Outlier Channel Splitting (OCS) handle outliers in weights and activations [18], BRECQ optimizes weight quantization in a block-wise manner [19], and AdaRound uses an adaptive rounding technique [20]. AdaQuant grants more freedom by independently optimizing each layer's weights using the calibration set [21].

More recent data-free PTQ methods generate synthetic data using BN statistics from the trained full-precision model [10], [22]. However, it has been shown that data generated this way can suffer from homogenization, which DSG addresses by relaxing distribution alignment and enhancing samples layer-wise [23]. While methods like IntraQ [24] and LRQ [25] further increase performance, they still rely on BN statistics and require fine-tuning.

## III. Motivation

BN is often used during neural network training to stabilize and speed up convergence by normalizing layer activations. However, for PTQ, generating synthetic data requires creating inputs that cover a wide range of activations within the network. This is more effectively achieved through optimization techniques like backpropagation, which refine synthetic inputs using the model's gradients without relying on BN statistics. Prior work, such as [23], has shown that images generated solely from BN statistics lack the diversity of real-world data. To address this, we use diverse loss functions during synthetic data generation.

Traditionally, generating datasets for quantization involves selecting at least one image from each target class to cover the activation space [11]. However, this approach can be computationally expensive for large datasets with many classes. Inspired by [26], who demonstrate that weight and activation distributions follow a bell curve (with rare large values), we explore whether a smaller subset of classes can still provide an effective activation range for quantization. Our hypothesis, discussed in subsection V-D1, is that focusing on fewer classes might simplify the calibration process while maintaining quantization accuracy.

## IV. Method

In this section, we introduce a framework for generating synthetic images aimed at improving PTQ of image classification models. Our approach involves designing a comprehensive loss function that guides the optimization of synthetic images.

**Algorithm 1** Generate Calibration Dataset

**Require:** Pre-trained model $F$, total number of classes $N$, number of target classes $M$
**Require:** Learning rate $\alpha$, total iterations $T$, loss weights $\lambda_{\text{tv}}$, $\lambda_{l_2}$, threshold $\epsilon$
**Ensure:** Set of synthetic images $X$
Randomly select $M$ unique target classes from $N$
Initialize an empty set $X$
**for** each target class $c$ in $M$ **do**
  Initialize synthetic image $x \sim \mathcal{N}(0, 1)$
  Set target label $y \leftarrow c$
  **for** $t = 1$ to $T$ **do**
    Calculate loss $\mathcal{L}$ using (4)
    Update synthetic image using (5)
    **if** $F(x) = y$ and $\mathcal{L} < \epsilon$ **then**
      **break**
    **end if**
  **end for**
  Append $x$ to $X$
**end for**
**return** $X$

The primary goal is to produce synthetic images that not only lead to accurate classification by the pre-trained model but also possess properties to get representative activation statistics essential for effective quantization. To achieve these objectives, we integrate three key components into our loss function: the classification loss, total variation loss, and L2 regularization loss. Below, we detail each component and its contribution to the synthetic data generation process.

### *A. Classification Loss*

The classification loss measures the discrepancy between the synthetic data's predicted labels and the target labels. This loss ensures that the synthetic images generated are informative for the classification task at hand.

$$\mathcal{L}_{\text{classification}} = \mathcal{CE}\left(F(x), y\right) \tag{1}$$

### *B. Total Variation Loss*

The total variation (TV) acts as a regularization term that promotes spatial smoothness in the synthetic images. By penalizing rapid intensity changes between neighboring pixels, the TV loss reduces noise and artifacts in the generated data.

$$\mathcal{L}_{\text{tv}} = \sum_{i,j} \left[ \left(x_{i+1,j} - x_{i,j}\right)^2 + \left(x_{i,j+1} - x_{i,j}\right)^2 \right] \tag{2}$$

### *C. $L_2$ Regularization Loss*

The L2 regularization loss penalizes the overall magnitude of the pixel values in the synthetic images. This loss prevents the generation of images with excessively high pixel intensities, which could adversely affect the stability and performance of the quantized model.

$$\mathcal{L}_{l_2} = \sum_{i,j} x_{i,j}^2 \tag{3}$$

### D. Total Loss

The total loss combines all the above loss terms, weighted by their respective hyperparameters.

$$\mathcal{L} = \mathcal{L}_{\text{classification}} + \lambda_{tv} * \mathcal{L}_{\text{tv}} + \lambda_{l_2} * \mathcal{L}_{l_2} \tag{4}$$

### E. Data Generation

Algorithm 1 generates synthetic images by randomly selecting $M$ unique target classes from $N$ total classes. For each target class $c$, a synthetic image $x$ is initialized with values drawn from $\mathcal{N}(0,1)$, and the target label is set to $y \leftarrow c$. Inspired by the iterative optimization methods, the algorithm iteratively refines $x$ for $T$ iterations as shown in (5), perturbing $x$ in the opposite direction of the gradient to minimize the loss. If the model's prediction for $x$ matches $y$ and the loss falls below a threshold $\epsilon$, the loop breaks early. This process repeats for each of the $M$ target classes, accumulating the generated images in $X$, which is then returned as the output.

$$x = x - \alpha \nabla_x \mathcal{L}(F(x), y) \tag{5}$$

## V. Experiments

In this section, we perform an in-depth assessment of the performance of our approach on image classification tasks using a series of comprehensive experiments.

### A. Implementation Details

We implemented our approach in PyTorch [27] for its strong automatic differentiation features. Experiments ran on an NVIDIA A100 GPU using pre-trained models from PyTorchCV.[1] We generated calibration dataset with Algorithm 1 and applied the independent calibration process from [10]. All layers underwent quantization with per-layer activation clipping. We used Stochastic Gradient Descent (SGD) with a momentum of 0.999 as the optimizer. Hyperparameters were tuned empirically: the number of target labels $M$ was optimized per model and bit setting (up to 35), iterations $T$ ranged from 100 to 300, learning rate $\alpha$ ranged from 0.1 to 0.3, and the threshold $\epsilon$ was set to 0.001.

### B. Evaluation

To demonstrate the effectiveness of our approach, we evaluate it on various network architectures with different bit settings. Our experiments include VGG16bn [28], ResNet18/20/50, SqueezeNext, InceptionV3, ShuffleNet, AlexNet, and MobileNetV2/V3. We assess these models using various bit-width configurations, such as W4A4 (4-bit weights and 4-bit activations), W6A6, and W8A8. We use validation datasets from ImageNet [29] and CIFAR10 [30] to evaluate our approach, measuring the effectiveness by assessing the top-1 accuracy of the quantized models.

[1]PyTorchCV: https://pypi.org/project/pytorchcv/

TABLE I: SqueezeNext, InceptionV3, ShuffleNet, and AlexNet on ImageNet

| Model | Method | W-bit | A-bit | Top-1 |
|---|---|---|---|---|
| SqueezeNext | Baseline | 32 | 32 | 69.38% |
| | Real Data | 6 | 6 | 66.51% |
| | ZeroQ | 6 | 6 | 39.83% |
| | DSG | 6 | 6 | 66.23% |
| | Ours | 6 | 6 | 67.55% |
| | Real Data | 8 | 8 | 69.23% |
| | ZeroQ | 8 | 8 | 68.01% |
| | DSG | 8 | 8 | 69.27% |
| | Ours | 8 | 8 | 69.31% |
| InceptionV3 | Baseline | 32 | 32 | 78.80% |
| | Real Data | 4 | 4 | 73.50% |
| | ZeroQ | 4 | 4 | 12.00% |
| | DSG | 4 | 4 | 57.17% |
| | Ours | 4 | 4 | 60.99% |
| | Real Data | 6 | 6 | 78.59% |
| | ZeroQ | 6 | 6 | 75.14% |
| | DSG | 6 | 6 | 78.12% |
| | Ours | 6 | 6 | 78.41% |
| | Real Data | 8 | 8 | 78.79% |
| | ZeroQ | 8 | 8 | 78.70% |
| | DSG | 8 | 8 | 78.81% |
| | Ours | 8 | 8 | 78.84% |
| ShuffleNet | Baseline | 32 | 32 | 65.07% |
| | Real Data | 6 | 6 | 56.25% |
| | ZeroQ | 6 | 6 | 39.92% |
| | DSG | 6 | 6 | 60.71% |
| | Ours | 6 | 6 | 62.17% |
| | Real Data | 8 | 8 | 64.52% |
| | ZeroQ | 8 | 8 | 64.46% |
| | DSG | 8 | 8 | 64.87% |
| | Ours | 8 | 8 | 64.94% |
| AlexNet | Baseline | 32 | 32 | 59.04% |
| | Ours | 4 | 4 | 45.97% |
| | Ours | 6 | 6 | 57.22% |
| | Ours | 8 | 8 | 57.47% |

### C. Comparison with SOTA Methods

To evaluate the benefits of our proposed PTQ scheme, we compare our method with other data-free PTQ approaches, such as DSG [23], ZeroQ [10], DFQ [31], ACIQ [26], MSE [32], KL [33], and OCS [18], on CIFAR10 and ImageNet datasets. Notably, DSG and ZeroQ are representative generative data-free PTQ methods that reconstruct synthetic data and calibrate the quantized network. We assess these methods under various bit-width configurations, with the results presented in table IV for the CIFAR10 dataset and table I, II, and III for the ImageNet dataset. For MobilNetV2/V3 on ImageNet dataset we compare our method with SelectQ [11] which uses training data for quantization.

On the CIFAR10 dataset, we evaluate our method with ResNet20 and VGG16bn, as shown in table IV. Our method consistently outperforms other methods across all bit-widths. Specifically, for ResNet20, our method improved accuracy by approximately 1.8% over DSG in the 4-bit setting. For the ImageNet dataset, we conducted experiments on MobileNetV2, MobileNetV3, ResNet18/50, SqueezeNext, InceptionV3, ShuffleNet, and AlexNet models. Our method consistently outperforms other quantization methods across different bit-widths. Notably, for MobileNetV3, our method achieved a significant improvement of 22.65% over SelectQ in the 4-bit

TABLE II: ResNet18/50 on ImageNet

| Model | Method | W-bit | A-bit | Top-1 |
|---|---|---|---|---|
| ResNet18 | Baseline | 32 | 32 | 71.47% |
| | Real Data | 4 | 4 | 65.22% |
| | DFQ | 4 | 4 | 0.10% |
| | ACIQ | 4 | 4 | 7.19% |
| | MSE | 4 | 4 | 15.08% |
| | KL | 4 | 4 | 16.27% |
| | ZeroQ | 4 | 4 | 26.04% |
| | DSG | 4 | 4 | 39.90% |
| | Ours | 4 | 4 | 43.21% |
| | Real Data | 6 | 6 | 71.18% |
| | ACIQ | 6 | 6 | 61.15% |
| | KL | 6 | 6 | 61.34% |
| | MSE | 6 | 6 | 66.96% |
| | DFQ | 6 | 6 | 67.30% |
| | ZeroQ | 6 | 6 | 69.74% |
| | DSG | 6 | 6 | 70.46% |
| | Ours | 6 | 6 | 70.59% |
| | Real Data | 8 | 8 | 71.48% |
| | ACIQ | 8 | 8 | 68.78% |
| | DFQ | 8 | 8 | 69.70% |
| | KL | 8 | 8 | 70.69% |
| | MSE | 8 | 8 | 71.01% |
| | ZeroQ | 8 | 8 | 71.43% |
| | DSG | 8 | 8 | 71.49% |
| | Ours | 8 | 8 | 71.47% |
| ResNet50 | Baseline | 32 | 32 | 77.72% |
| | Real Data | 4 | 4 | 68.13% |
| | ACIQ | 4 | 4 | 61.15% |
| | ZeroQ | 4 | 4 | 8.20% |
| | DFQ | 4 | 4 | 10.32% |
| | DSG | 4 | 4 | 56.12% |
| | Ours | 4 | 4 | 58.31% |
| | Real Data | 6 | 6 | 76.84% |
| | ZeroQ | 6 | 6 | 75.56% |
| | DSG | 6 | 6 | 76.90% |
| | Ours | 6 | 6 | 76.73% |
| | Real Data | 8 | 8 | 77.70% |
| | ZeroQ | 8 | 8 | 77.67% |
| | DSG | 8 | 8 | 77.72% |
| | Ours | 8 | 8 | 77.69% |

setting, reaching 23.01%. In the case of ResNet18, our method achieved the highest accuracy in the 4-bit setting, with a 3.31% improvement over DSG, reaching 43.21%. For ResNet50, our method improved accuracy by 2.19% over DSG in the 4-bit setting, achieving 58.31%. InceptionV3 showed a significant improvement with our method, achieving 60.99% in the 4-bit setting, which is 3.82% higher than DSG, while slightly improving accuracies in 6 and 8-bit settings. In table I we also demonstrate the quantization results of AlexNet [1] model using our method which is not shown by other methods due to the unavailability of BN layer in AlexNet.

Furthermore similar to methods such as ZeroQ and DSG we require a small number of synthetic images to achieve effective PTQ. Empirically, we have determined the effective calibration dataset size for each model and bit setting. For example, we use 25 images for ResNet18 in a 4-bit setting and 35 images for ResNet50 in a 4-bit setting. Importantly, we never exceed 35 images for effective quantization across all models and bit settings. This substantial reduction in calibration dataset size does not compromise quantization performance, making our method highly efficient for various classification models.

Overall, our quantization method demonstrated superior performance across various models and bit-widths with fewer images, particularly in the 4-bit setting. It consistently achieved high accuracy, making it suitable for resource-constrained environments without significant loss in performance.

TABLE III: MobileNetV2/V3 on ImageNet

| Model | Method | W-bit | A-bit | Top-1 |
|---|---|---|---|---|
| MobileNetV2 | Baseline | 32 | 32 | 72.97% |
| | SelectQ | 4 | 4 | 10.88% |
| | Ours | 4 | 4 | 12.31% |
| | SelectQ | 6 | 6 | 70.25% |
| | Ours | 6 | 6 | 70.11% |
| | SelectQ | 8 | 8 | 72.84% |
| | Ours | 8 | 8 | 72.83% |
| MobileNetV3 | Baseline | 32 | 32 | 75.34% |
| | SelectQ | 4 | 4 | 0.36% |
| | Ours | 4 | 4 | 23.01% |
| | SelectQ | 6 | 6 | 60.04% |
| | Ours | 6 | 6 | 72.85% |
| | SelectQ | 8 | 8 | 75.04% |
| | Ours | 8 | 8 | 75.06% |

### *D. Ablation Study*

In this section, we perform ablation studies to examine the impact of various hyperparameters, quantizing all layers of ResNet18 to 4 bits and evaluating top-1 accuracy on ImageNet.

TABLE IV: ResNet20 and VGG16bn on CIFAR-10

| Model | Method | W-bit | A-bit | Top-1 |
|---|---|---|---|---|
| ResNet20 | Baseline | 32 | 32 | 94.08% |
| | Real Data | 4 | 4 | 87.38% |
| | ZeroQ | 4 | 4 | 85.39% |
| | DSG | 4 | 4 | 87.79% |
| | Ours | 4 | 4 | 89.59% |
| | Real Data | 6 | 6 | 93.80% |
| | ZeroQ | 6 | 6 | 93.33% |
| | DSG | 6 | 6 | 93.55% |
| | Ours | 6 | 6 | 93.63% |
| | Real Data | 8 | 8 | 93.95% |
| | ZeroQ | 8 | 8 | 93.94% |
| | DSG | 8 | 8 | 93.97% |
| | Ours | 8 | 8 | 94.00% |
| VGG16bn | Baseline | 32 | 32 | 93.86% |
| | Real Data | 4 | 4 | 92.50% |
| | ZeroQ | 4 | 4 | 91.79% |
| | DSG | 4 | 4 | 92.89% |
| | Ours | 4 | 4 | 93.17% |
| | Real Data | 6 | 6 | 93.48% |
| | ZeroQ | 6 | 6 | 93.45% |
| | DSG | 6 | 6 | 93.68% |
| | Ours | 6 | 6 | 93.85% |
| | Real Data | 8 | 8 | 93.59% |
| | ZeroQ | 8 | 8 | 93.53% |
| | DSG | 8 | 8 | 93.61% |
| | Ours | 8 | 8 | 93.79% |

*1) Impact of Calibration Dataset Size on Quantization:* We conducted two experiments to evaluate the necessity of having a large calibration dataset in model quantization.

In the first experiment, we tested unique dataset sizes ranging from 10 to 1000 images and measured the top-1 accuracy of the quantized model using our synthetic dataset generation method, as reported in Fig. 3. Contrary to the common belief that larger calibration datasets lead to better quantized model performance, we found that the best results were achieved with

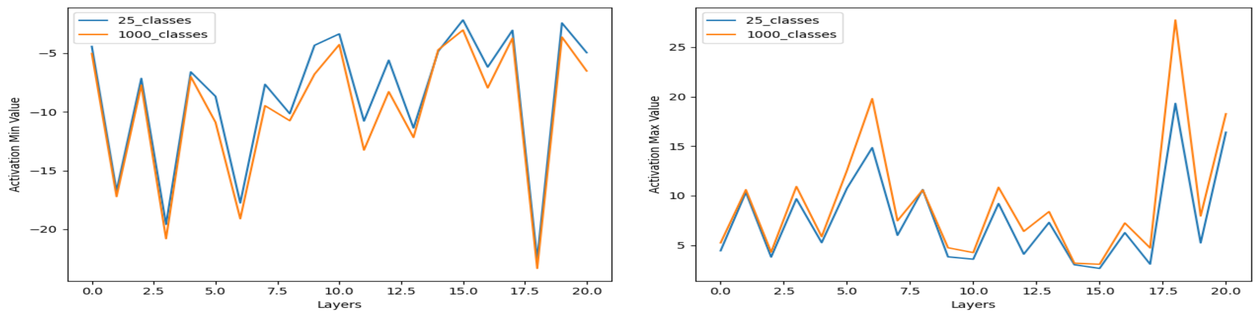
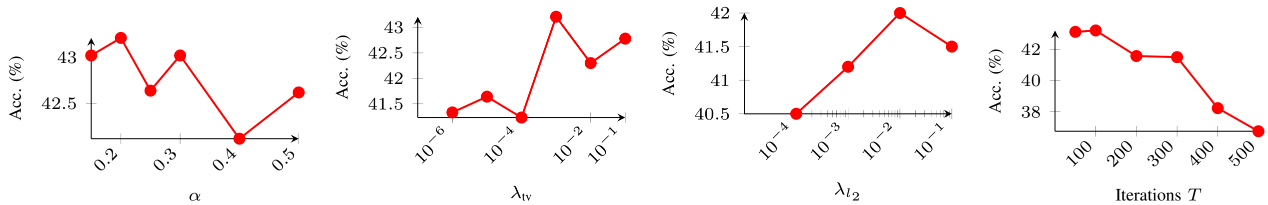



Fig. 1: Minimum (left) and maximum (right) activation values for ResNet18.




(a) Learning Rate $\alpha$ (b) Weight $\lambda_{\text{tv}}$ (c) Weight $\lambda_{l_2}$ (d) Number of Iterations $T$

Fig. 2: Effect of hyperparameters on the top-1 accuracy of the 4-bit ResNet18 model on ImageNet.

a smaller dataset size. For ResNet18, the highest accuracy of 43.21% was obtained with just 25 images. Interestingly, increasing the dataset size beyond this point resulted in a decline in accuracy; for example, accuracy dropped to 41.83% with 1000 images. This suggests that using a smaller dataset can be more effective for PTQ, and that an optimal dataset size exists beyond which additional images may introduce noise or redundancy, negatively impacting performance.

In the second experiment, we plotted the minimum and maximum activation ranges for each convolution layer in ResNet18 using datasets created with 25 and 1000 images, respectively. The results, shown in Fig. 1, indicate that the activation ranges for 25 images closely match those obtained with 1000 images. This observation demonstrates that even with significantly fewer images, the activation ranges remain stable and representative of the model's behavior. Consequently, it is not necessary to have at least one image from each class to create a representative dataset for PTQ.

Overall, our findings demonstrate that a smaller, more manageable dataset can be effectively used for calibration, simplifying the process and reducing the need for extensive calibration datasets.

*2) Influence of Random Target Class Selection:* In exploring the impact of random target class selection on our synthetic data generation process, we found that the top-1 accuracy of the quantized ResNet18 model on ImageNet varied significantly depending on the specific target classes selected. To achieve similar performance levels across different random selections, we needed to tune hyperparameters $\alpha$, $\lambda_{\text{tv}}$, $\lambda_{l_2}$, and $T$. This necessity arises because different classes may present varying levels of complexity or feature distinct characteristics that influence the effectiveness of the synthetic data in capturing the necessary activation statistics for quantization.

*3) Analyzing Hyperparameters:* We investigate the impact of the hyperparameters: learning rate $\alpha$, loss weights $\lambda_{\text{tv}}$, $\lambda_{l_2}$, and the number of iterations $T$. In our experiments, we fix three hyperparameters at their baseline values while tuning the fourth to assess its individual effect on the synthetic data generation and the performance of the quantized model. The results, illustrating how each hyperparameter influences the top-1 accuracy on ImageNet, are presented in Fig 2. From the figure, we can see that for ResNet18, the optimal values of the parameters are $\alpha = 0.2$, $\lambda_{\text{tv}} = 0.001$, $\lambda_{l_2} = 0.0001$, and $T = 100$. We did not observe any impact from different values of threshold $\epsilon$. The same accuracy is obtained for values of $\epsilon$ in the range [0.00001, 0.01].

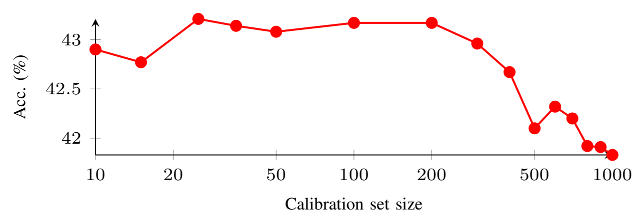



Fig. 3: ResNet18 quantized model performance with different calibration set sizes.

## VI. Conclusions

In this paper, we introduced a post-training data-free quantization method that generates synthetic data using the trained full-precision model independent of BN statistics, making our approach versatile and applicable to any model architecture. Our experimental results demonstrate that it is not necessary to include images from each target category; selecting only a few target classes is sufficient to create an effective calibration dataset. Our method consistently outperforms existing generative data-free quantization methods, as demonstrated across architectures like ResNet18/50, SqueezeNext, InceptionV3, ShuffleNet, and MobileNetV2/V3. Notably, our approach shows significant improvements in 4-bit precision settings, increasing the top-1 accuracy on the ResNet18 model by over 3.31% compared to the SOTA DSG method. These findings underscore the effectiveness and generalizability of our approach, highlighting its potential to achieve high accuracy with lower bit-widths and fewer calibration images, making it a promising solution for efficient model deployment in resource-constrained environments.

## VII. Acknowledgment

We acknowledge support from NSF Award #2245756 and DOE Award #DE-SC0024598. This research was partially supported in part by DARPA under the TIAMAT project HR00112490420 (FIRST-FM) and the Cooperative Agreement HR00112420004 (AI-CRAFT), as well as by start-up institutional funds from the University of Florida. Any findings or conclusions expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of Defense or the United States Government.

## References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

[3] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, *et al.*, "Human-level control through deep reinforcement learning," *nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[4] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[5] A. G. Howard, "Mo-bilenets: Efficient convolutional neural networks for mo-bile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[6] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. 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*, pp. 2704–2713, 2018.

[7] Y. Nahshan, B. Chmiel, C. Baskin, E. Zheltonozhskii, R. Banner, A. M. Bronstein, and A. Mendelson, "Loss aware post-training quantization," *Machine Learning*, vol. 110, no. 11, pp. 3245–3262, 2021.

[8] R. Banner, Y. Nahshan, and D. Soudry, "Post training 4-bit quantization of convolutional networks for rapid-deployment," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[9] S. Xu, H. Li, B. Zhuang, J. Liu, J. Cao, C. Liang, and M. Tan, "Generative low-bitwidth data free quantization," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pp. 1–17, Springer, 2020.

[10] Y. Cai, Z. Yao, Z. Dong, A. Gholami, M. W. Mahoney, and K. Keutzer, "Zeroq: A novel zero shot quantization framework," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 13169–13178, 2020.

[11] Z. Zhang, Y. Gao, J. Fan, Z. Zhao, Y. Yang, and S. Yan, "Selectq: Calibration data selection for post-training quantization," *Authorea Preprints*, 2023.

[12] M. Courbariaux, I. Hubara, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1," *arXiv preprint arXiv:1602.02830*, 2016.

[13] S. Zhou, Y. Wu, Z. Ni, X. Zhou, H. Wen, and Y. Zou, "Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients," *arXiv preprint arXiv:1606.06160*, 2016.

[14] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "Xnor-net: Imagenet classification using binary convolutional neural networks," in *European conference on computer vision*, pp. 525–542, Springer, 2016.

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

[16] J. L. McKinstry, S. K. Esser, R. Appuswamy, D. Bablani, J. V. Arthur, I. B. Yildiz, and D. S. Modha, "Discovering low-precision networks close to full-precision networks for efficient embedded inference," *arXiv preprint arXiv:1809.04191*, 2018.

[17] J. Choi, Z. Wang, S. Venkataramani, P. I.-J. Chuang, V. Srinivasan, and K. Gopalakrishnan, "Pact: Parameterized clipping activation for quantized neural networks," *arXiv preprint arXiv:1805.06085*, 2018.

[18] R. Zhao, Y. Hu, J. Dotzel, C. De Sa, and Z. Zhang, "Improving neural network quantization without retraining using outlier channel splitting," in *International conference on machine learning*, pp. 7543–7552, PMLR, 2019.

[19] Y. Li, R. Gong, X. Tan, Y. Yang, P. Hu, Q. Zhang, F. Yu, W. Wang, and S. Gu, "Brecq: Pushing the limit of post-training quantization by block reconstruction," *arXiv preprint arXiv:2102.05426*, 2021.

[20] M. Nagel, R. A. Amjad, M. Van Baalen, C. Louizos, and T. Blankevoort, "Up or down? adaptive rounding for post-training quantization," in *International Conference on Machine Learning*, pp. 7197–7206, PMLR, 2020.

[21] I. Hubara, Y. Nahshan, Y. Hanani, R. Banner, and D. Soudry, "Accurate post training quantization with small calibration sets," in *International Conference on Machine Learning*, pp. 4466–4475, PMLR, 2021.

[22] M. Haroush, I. Hubara, E. Hoffer, and D. Soudry, "The knowledge within: Methods for data-free model compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8494–8502, 2020.

[23] H. Qin, Y. Ding, X. Zhang, J. Wang, X. Liu, and J. Lu, "Diverse sample generation: Pushing the limit of generative data-free quantization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 10, pp. 11689–11706, 2023.

[24] Y. Zhong, M. Lin, G. Nan, J. Liu, B. Zhang, Y. Tian, and R. Ji, "Intraq: Learning synthetic images with intra-class heterogeneity for zero-shot network quantization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12339–12348, 2022.

[25] Y. Luo, Y. Gao, Z. Zhang, J. Fan, H. Zhang, and M. Xu, "Long-range zero-shot generative deep network quantization," *Neural Networks*, vol. 166, pp. 683–691, 2023.

[26] R. Banner, Y. Nahshan, E. Hoffer, and D. Soudry, "Aciq: Analytical clipping for integer quantization of neural networks," 2018.

[27] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, Ieee, 2009.

[30] A. Krizhevsky, G. Hinton, *et al.*, "Learning multiple layers of features from tiny images," 2009.

[31] M. Nagel, M. v. Baalen, T. Blankevoort, and M. Welling, "Data-free quantization through weight equalization and bias correction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1325–1334, 2019.

[32] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems," *arXiv preprint arXiv:1512.01274*, 2015.

[33] W. Sung, S. Shin, and K. Hwang, "Resiliency of deep neural networks under quantization," *arXiv preprint arXiv:1511.06488*, 2015.