Zero-shot Adversarial Quantization

Yuang Liu , Wei Zhang*, Jun Wang*
East China Normal University, Shanghai, China
{frankliu624, zhangwei.thu2011, wongjun}@gmail.com

Abstract

Model quantization is a promising approach to compress deep neural networks and accelerate inference, making it possible to be deployed on mobile and edge devices. To retain the high performance of full-precision models, most existing quantization methods focus on fine-tuning quantized model by assuming training datasets are accessible. However, this assumption sometimes is not satisfied in real situations due to data privacy and security issues, thereby making these quantization methods not applicable. To achieve zero-short model quantization without accessing training data, a tiny number of quantization methods adopt either post-training quantization or batch normalization statistics-guided data generation for fine-tuning. However, both of them inevitably suffer from low performance, since the former is a little too empirical and lacks training support for ultra-low precision quantization, while the latter could not fully restore the peculiarities of original data and is often low efficient for diverse data generation. To address the above issues, we propose a zero-shot adversarial quantization (ZAQ) framework, facilitating effective discrepancy estimation and knowledge transfer from a full-precision model to its quantized model. This is achieved by a novel two-level discrepancy modeling to drive a generator to synthesize informative and diverse data examples to optimize the quantized model in an adversarial learning fashion. We conduct extensive experiments on three fundamental vision tasks, demonstrating the superiority of ZAQ over the strong zero-shot baselines and validating the effectiveness of its main components. Code is available at https://git.io/Jqc0y.

1. Introduction

Although deep neural networks (DNNs), especially deep convolutional networks (DCNs), have achieved remarkable performance in a broad range of computer vision tasks [20, 40, 24, 34], their ever-growing complexities — a large number of model parameters — inhibit the applications on cloud and edge devices. As a consequence, model quantization, converting high-precision parameters to low-precision ones, becomes one of the main paradigms in model compression and acceleration [10]. To mitigate the performance degradation issue due to model quantization, quantization-aware fine-tuning approaches have been extensively studied to optimize quantized models on the full training datasets [41, 16, 38]. However, in real situations, original training data is sometimes inaccessible due to privacy and security issues. For instance, electronic health records usually contain patients' private information. As such, the quantization-aware fine-tuning methods are no longer applicable.

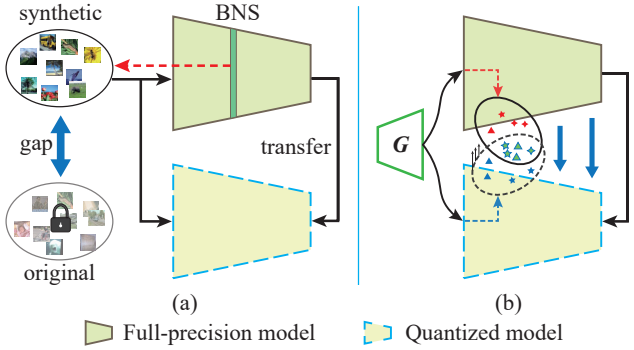


Figure 1. Overview of our framework. The methods based on sample reconstruction are shown as part (a), and part (b) is the overview of our framework. BNS is short for batch normalization statistics stored in the BN layers.

Post-training quantization methods [2, 30, 47] therefore emerge to quantize weights and activations in DNNs through correction strategies, without fine-tuning. However, there is a negligible gap between the strategies and the goals of target tasks, causing the quantized models to suffer from performance degradation. This issue is even amplified for the ultra-low precision situation. To address this, batch normalization statistics (BNS)-guided data generation is leveraged by recent methods [4, 42]. They aim at synthesizing data samples that match the real-data statistics encoded in the batch normalization layers of full-precision deep models. The synthetic data is further leveraged to fine-tune the

*Corresponding author.

quantized models by directly optimizing on target tasks supervised by its full-precision model, as shown in Figure 1(a). Although the performance of ultra-low precision models is boosted to some extent, thanks to fine-tuning, data generated by batch normalization statistics is hard to fully recover the peculiarities of training data and the generation process itself is time-consuming due to data redundancy. These issues make the results still far from satisfactory.

This paper seeks to promote the development of data-free model quantization by addressing the above-mentioned issues. We, therefore, present a novel learning framework named Zero-shot Adversarial Quantization (ZAQ) to perform model quantization without utilizing any sample from training data. Specifically, we devise a two-level discrepancy modeling strategy for ZAQ to measure the gap between a quantized model and its corresponding full-precision model. We consider not only the output discrepancy from models' top layers, just similar as existing data-free model quantization methods, but also fuses a new intermediate inter-channel discrepancy based on feature maps. A generator in ZAQ is responsible for generating informative and diverse data examples in an adversarial learning manner [15] — optimization based on a minimax game — to enable effective discrepancy estimation and knowledge transfer, as depicted Figure 1(b). In addition, activation regularization is adopted to facilitate the generator to obtain examples more sensitive to the network. To sum up, our contributions are as follows:

- We propose a zero-shot adversarial quantization framework to support effective data generation and knowledge transfer. To our best knowledge, it represents the first effort to apply adversarial learning to data-free model quantization.

- A novel two-level discrepancy modeling strategy is devised to measure the discrepancy between a quantized model and its full-precision model, thereby guiding the training of the quantized model and generator.

- We conduct extensive experiments on image classification, segmentation, and object detection tasks, showing our ZAQ framework achieves state-of-the-art results in data-free situation, works well for ultra-low precision scenarios, and is efficient compared to the approaches of BNS-guided data generation for model quantization.

2. Related Work

Model quantization is a promising model compression methods aiming to store parameters with fewer bits so that computation can be executed on integer-arithmetic units rather than on power-hungry floating-point ones [16]. An important challenge with quantization is that it can lead to significant performance degradation, especially in ultra-low precision settings. To cope with this, PACT [7] used an activation clipping parameter to find the right quantization scale. Zhu *et al*. [49] built a flexible and unified INT8 training framework for vision tasks. Flexpoint [18], MPT [28] and DFP [9] all use 16-bit floating-point to train DNNs with accuracy comparable to full-precision model. And there are some approaches to decrease induced degradation by quantization-aware training [1, 16, 26] or reducing the dynamic range of activations by clipping outliers [47, 29, 2]. Instead of focusing on improving the quantization process itself, [27] explored an equivalent weight arrangement that make the net less sensitive to quantization. However, all above quantization methods generally require access to the entire training data which is not always available as aforementioned.

Data-free model compression has been a hot topic and draw more and more attention in recent years, which is a challenge to compress model without training data. Srinivas and Babu [39], the pioneers in data-free compression, introduced a channel pruning method without original training data. Since then, more and more kinds of data-free or zero-shot compression methods were proposed, including quantization [2, 4, 42], weight factorization [30] and knowledge distillation (KD) [25, 5, 13, 43, 23]. DFQ [30] and ACIQ [2] are both post-training quantization methods relying on weight equalization or bias correction without fine-tuning on the entire dataset. But when applied to ultra-low precision (*i.e*., lower than 6-bit) model, these kinds of quantization methods cannot prevent quantization models from performance degradation. Most of the data-free KD methods attempt to reconstruct the original data from pre-trained teacher model utilizing prior information about the underlying data distribution, such as BNS [44], Dirichlet distribution [31] and category information [5]. However, they ignore the intermediate features to guide the student network learning.

Two recent data-free quantization studies [4, 42] quantize and fine-tune models without needing original data. Their core idea is to reconstruct some samples from full-precision models to fine-tune quantized models. To be specific, ZeroQ [4] directly reconstructs samples by optimizing from random noises according to BNS of full-precision models. GDFQ [42] further adopts a generator to reconstruct samples guided by BNS and extra category label information, which limits its application to classification tasks. To sum up, there is still a large gap between the data generated based on BNS and original training data after a time-consuming generation process. Moreover, both ZeroQ and GDFQ poorly support high-level vision tasks due to the lack of considering information from intermediate layers of full-precision models.

3. The Computational Framework

Framework Overview: Figure 2 depicts the basic framework of ZAQ. It contains pretrained full-precision model P,

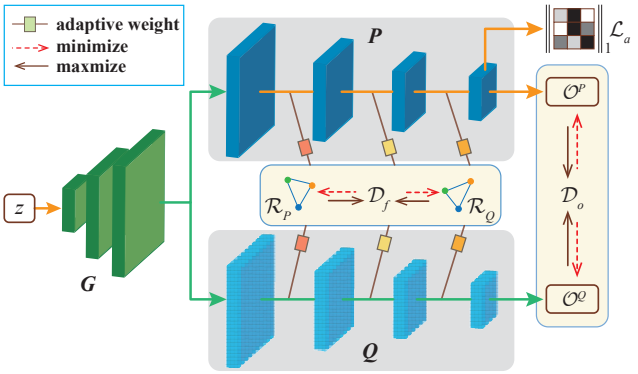


Figure 2. Framework of ZAQ.

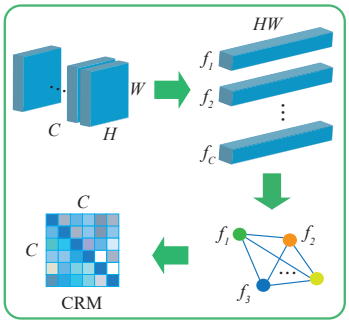


Figure 3. Illustration of obtaining channel relation map.

quantized model Q, and generator G. G is responsible for generating informative and diverse data examples, which are used by a two-level discrepancy function to compute the discrepancy between P and Q. The discrepancy function is composed of output discrepancy $\mathcal{D}_o$ and intermediate inter-channel discrepancy $\mathcal{D}_f$. Consequently, Q and G are optimized through a minimax game, where the adversarial learning of the two-level discrepancy modeling is conducted. In addition, activation regularization $\mathcal{L}_a$ encourages G to generate more informative and diverse examples.

In what follows, we first introduce the preliminary of the quantization function used in this paper. Then we detail the proposed framework.

3.1. Preliminary

A common practise in training a neural network with low-precision weights and activations is to introduce a quantization function. Considering the general case of k-bit quantization [48], we define the uniform quantization function $q(\cdot)$ as:

$$q(v) = \text{round}\big(S \cdot (v - Z)\big)\,, \tag{1}$$

where v denotes the full-precision (float32) value, S is the scaling factor, and Z is the zero point in float32. According to whether the parameter Z is zero, uniform quantization can be divided into two categories: symmetric quantization and asymmetric quantization. Here we use symmetric quantization and set $Z = 0$. Consequently, S is formulated as:

$$S = \frac{2^{k-1} - 1}{\max(|x_{\text{f}}|)}\,, \tag{2}$$

where x_{f} is any one of float32-point numbers.

The key of model quantization is to reduce the discrepancy $\mathcal{D}$ between full-precision model P and low-precision model Q through optimizing Q, which can be expressed as:

$$Q^* = \min_Q \mathcal{D}(P, Q)\,. \tag{3}$$

3.2. Two-level Discrepancy Modeling

As aforementioned, the framework ZAQ leverages a novel two-level discrepancy function to model the discrepancy between full-precision and quantized models. First, we assume there is a data example x_g generated by G, *i.e.*, $x_g = G(z)$ where z is random noise. We denote the corresponding prediction outputs for full-precision model P and quantized model Q as $P(x_g)$ and $Q(x_g)$, respectively. There are some distance metrics that can be used to measure the discrepancy, such as Kullback-Leibler (KL) divergence. KL divergence is efficient in data-driven knowledge transfer or distillation, but it is insufficient to maximize the discrepancy when training generator G. This is because some unexpected samples may be similar in prediction, making the negative KLD too small to optimize. Instead, we adopt L1 loss to measure the output discrepancy $\mathcal{D}_o$ in a more direct way:

$$\mathcal{D}_o(P, Q; G) = \mathbb{E}_{x_g}\left[\frac{1}{N}\left\|P(x_g) - Q(x_g)\right\|_1\right]\,, \tag{4}$$

where N is element number in the outputs, for instance, class number for classification and label map size for segmentation.

Inspired by the idea of harnessing intermediate feature maps to improve performance in knowledge distillation [45, 32], we further propose Channel Relation Map (CRM) to gain intermediate inter-channel discrepancy. Noting that although there are a few studies [32, 22] modeling relations between data instances, we are the first to consider similarity relations between different channels of feature maps, which are introduced later.

Specifically, we define intermediate inter-channel discrepancy as below:

$$\mathcal{D}_f(P, Q; G) = \mathbb{E}_{x_g}\left[\sum_l^L \frac{\omega^{(l)}}{{C^{(l)}}^2}\left\|\mathcal{R}_P^{(l)}(x_g) - \mathcal{R}_Q^{(l)}(x_g)\right\|_1\right]\,, \tag{5}$$

where L is the total number of layers exploited for ZAQ, $\mathcal{R}_P^{(l)}(\cdot)$ and $\mathcal{R}_Q^{(l)}(\cdot)$ represent CRM extracted from the l-th layer of P and Q, respectively. $\omega^{(l)}$ is the adaptive weight

allocated to the l-th layer, and $C^{(l)}$ is the output channel number of the l-th layer. Actually, we usually select the last layer in each group or block for residual neural networks, and the layer number is $3 \sim 4$ for VGG in our experiments.

A conventional manner to measure the discrepancy of intermediate layers relies on correlating feature maps of P and Q, just as what KD commonly does [35, 45]. However, since the numerical spans of P and Q are very different because of precision settings, the gap between feature maps in P and Q is relatively large (verified in Table 4). Therefore, we introduce CRM to address this issue. Gram matrix can represent certain relationships between feature vectors to reflect the characteristics of images, and is commonly used in style transfer [14]. But it is unreliable to measure the feature discrepancy between two networks by directly using feature vectors with different precision. Here we extend it to channel relation map to capture the relations towards different channels in the same layer of one model. It cannot only shield the influence of feature maps with different numerical spans, but also represent the high-dimensional features of samples. Figure 3 illustrates the procedures of obtaining CRM, which are the same for both P and Q. Taking the feature map $\tilde{\mathcal{F}}^{(l)} \in \mathbb{R}^{C\times H\times W}$ extracted from the l-th layer of P (or Q) for clarification, it can be flattened into $\mathcal{F}^{(l)} \in \mathbb{R}^{C\times HW}$, which is composited by C channel-wise feature vectors: $\left[\boldsymbol{f}_1^{(l)} \quad \boldsymbol{f}_2^{(l)} \quad \cdots \quad \boldsymbol{f}_C^{(l)}\right]^\top$. Then the consine similarity between channel features $\boldsymbol{f}_i^{(l)}$ and $\boldsymbol{f}_j^{(l)}$ is defined as below:

$$\mathcal{R}_{ij}^{(l)} = \frac{< \boldsymbol{f}_i^{(l)}, \boldsymbol{f}_j^{(l)} >}{\left\|\boldsymbol{f}_i^{(l)}\right\|_2 \left\|\boldsymbol{f}_j^{(l)}\right\|_2}. \tag{6}$$

Based on $\mathcal{R}_{ij}^{(l)}$ $(i, j \in \{1, 2, \ldots, C\})$, the corresponding matrices $\mathcal{R}_P^{(l)}(x_g)$ and $\mathcal{R}_Q^{(l)}(x_g)$ can be obtained.

To adaptively determine $\omega^{(l)}$ in Eq. 5, we use the discrepancy calculated for the two models and define the following computational equation:

$$\omega^{(l)} = \frac{\exp\left(\mathrm{EMA}_T\left(\mathbb{E}_{x_g\in\mathcal{B}_t}\left[\left\|\mathcal{R}_P^{(l)}(x_g) - \mathcal{R}_Q^{(l)}(x_g)\right\|_1\right]\right)\right)}{\sum_{l'}^{L}\exp\left(\mathrm{EMA}_T\left(\mathbb{E}_{x_g\in\mathcal{B}_t}\left[\left\|\mathcal{R}_P^{(l')}(x_g) - \mathcal{R}_Q^{(l')}(x_g)\right\|_1\right]\right)\right)}, \tag{7}$$

where EMA_T denotes exponential moving averaging, T is the training steps in an epoch, and L is the number of layers exploited for ZAQ. By this way, more attention will be paid to the model layer that has a larger difference in CRMs. Besides, in order to avoid breaking the balance in long-term training, $\omega^{(l)}$ needs to be re-initialized by $\frac{1}{L}$ when a new epoch starts.

3.3. Adversarial Knowledge Transfer

Our ZAQ framework trains quantized model Q and generator G in an adversarial minimax game, which contains **discrepancy estimation** and **knowledge transfer** stages. In discrepancy estimation stage, the generator G aims at maximizing the two-level discrepancy between Q and P to search for discrepancy represent space. The loss is defined as follows:

$$\mathcal{L}_{DE} = -\mathcal{D}_o(P, Q; G) - \alpha\mathcal{D}_f(P, Q; G), \tag{8}$$

where α is a hyperparameter to balance $\mathcal{D}_o$ and $\mathcal{D}_f$.

In knowledge transfer stage, quantized model Q is optimized to minimize the two-level discrepancy to approximate full-precision model P, denoted as:

$$\mathcal{L}_{KT} = \mathcal{D}_o(P, Q; G) + \alpha\mathcal{D}_f(P, Q; G). \tag{9}$$

As a consequence, the knowledge is transferred from P to Q progressively in the zero-shot situation.

3.4. Activation Regularization

Although the L1 loss function can relieve the model from falling into some abnormal sample points in discrepancy estimation, they exist all the time and interfere with the generator's exploration of the original input domain. These unexpected samples could make the prediction distributions of the two networks consistent but they are not in the working domain of full-precision model. We assume the infinite discrepancy space between model P and Q is $\mathbf{\Omega}$, in which the generator G explore valuable samples for transfer learning. In fact, $\mathbf{\Omega}$ consists of two subspaces $\mathbf{\Omega}_P$ and $\mathbf{\Omega}_U$, which means $\mathbf{\Omega} = \mathbf{\Omega}_P \cup \mathbf{\Omega}_U$. $\mathbf{\Omega}_P$ is the subspace that is equal to the original training data domain, or the working domain of pretrained model P. And $\mathbf{\Omega}_U$ is an infinite subspace outside the working domain of P. The goal of the generator is to synthesize samples distributed in the subspace $\mathbf{\Omega}_P$, rather in $\mathbf{\Omega}_U$.

According to several researches about interpretability of DNNs [46, 11] or sample reconstruction [25, 5], the activation layer reflects the sensitivity of the neural network to the input data, and higher activation means more correlation between synthetic samples and working domain of P. Hence, we further leverage activation regularization to constraint the generator to explore and synthesize valuable samples. We denote the i-th channel activation map extracted by the last convolution layer of network P as $h_i^P, i \in \{1, 2, \ldots, M\}$, where M is the number of activation maps. Then, the activation regularization can be formulated as

$$\mathcal{L}_a = -\frac{1}{M}\sum_i^M \left\|h_i^P\right\|_1. \tag{10}$$

With the intuition that high activation values mean a better matching between a given input example and training data, we incorporate $\mathcal{L}_a$ into Eq. 8 and minimize the following loss to guide generator training.

$$\mathcal{L}_{DE} = -\mathcal{D}_o(P, Q; G) - \alpha\mathcal{D}_f(P, Q; G) + \beta\mathcal{L}_a. \tag{11}$$

Finally, the detailed procedures of the proposed framework ZAQ is summarized in Algorithm 1.

Algorithm 1: Zero-shot Adversarial Quantization

Input: A pretrained full-precision model $P(x;\theta^p)$, quantization precision.
Output: Quantized model $Q(x;\theta^q)$

Quantize the model P as Q by Eq. 3;
for *number of epochs* **do**
  Initialize adaptive weights by $\frac{1}{L}$;
  for *number of training steps* **do**
    # Discrepancy Estimation
    $z \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $x_g \leftarrow G(z;\theta^g)$;
    Estimate $\mathcal{L}_{DE}$ by Eq. 11;
    Fix θ^q, update θ^g:

$$\theta^g \leftarrow \theta^g - \eta \frac{\partial \mathcal{L}_{DE}}{\partial \theta^g}$$

    Update adaptive weights $\omega^{(l)}$ by Eq. 7;
    # Knowledge Transfer
    $z \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$, $x_g \leftarrow G(z;\theta^g)$;
    Calculate $\mathcal{L}_{KT}$ by Eq. 9;
    Fix θ^g, update θ^q:

$$\theta^q \leftarrow \theta^q - \eta \frac{\partial \mathcal{L}_{KT}}{\partial \theta^q}$$

  end
  decay η;
end

4. Experiments

4.1. Experimental Setup

4.1.1 Datasets

We evaluate our approach on the following six datasets: CIFAR10, CIFAR100, and ImageNet for classification, Cityscapes and CamVid for segmentation, and VOC2012 for object detection.

CIFAR. CIFAR10 [19] and CIFAR100 consist of 32×32 color images with 10 and 100 classes, respectively. Both are split into a 50,000-image train set and a 10,000-image test set.

ImageNet. The 1,000-class dataset from ILSVRC 2012 [36] provides 1.2 million images for training, and 50,000 for validation.

Cityscapes. Cityscapes [8] is for urban scene understanding and contains 30 classes with only 19 classes used for evaluation. It provides 3,975 images with fine segmentation annotations, including 2,975 images for training and 500 images for testing.

CamVid. CamVid [3] is an automotive dataset, containing 367 training and 233 testing images. We perform on the commonly used 11 different classes.

VOC2012. A total of 11540 images are included in PASCAL VOC2012 [12], where each image contains a set of objects, out of 20 different classes.

4.1.2 Baselines

To evaluate the effectiveness and advantages of our proposed method, we compared it with both data-free fine-tuning methods and post-training quantization methods. The baselines are briefly described as follows.

FT. We use original training data to Fine-Tune (FT) a quantized model.

RQ. Raw Quantization (RQ) method directly testing the model after quantization without any fine-tuning.

DFQ [30]. A post-training quantization method uses a weight equalization scheme to remove outliers in both weights and activations.

ACIQ [2]. It analytically computes a clipping range, as well as a per-channel bit allocation for neural networks without any fine-tuning/training.

ZeroQ [4]. It retrains a quantized model by reconstructed data instead of original data.

GDFQ [42]. It is also a fine-tuning method by recovering fake data via a conditional generator. Yet it only supports classification tasks.

4.1.3 Implementation Details

We implement all networks and quantization methods in Pytorch. For all datasets, we adopt the same data augmentation procedure on pretraining as [37] for making fair comparisons. We adopt SGD with momentum 0.9 and weight decay 5×10^{-4} in both pretraining and fine-tuning. All the models are pretrained for 200 epochs and the learning rates are decayed by 0.1 for every 80 epochs on datasets, except ImageNet, on which we directly use the official pretrained models. We construct a generator following DCGAN [33] with 256-dimension noise and is trained with Adam [17]. But for CIFAR, we just reduce the channels of all layers in the generator to a quarter and set the dimension of noise to 100, due to the smaller size of the samples. Moreover, the learning rates of quantized models and generators are initialized to 0.1 and 1×10^{-3}, respectively. The learning rates of SGD and Adam are decayed by different steps in different tasks. In training, we set the batch size to 256 for CIFAR, 64 for ImageNet and VOC2012, and 16 for segmentation datasets. As for the hyperparameters, we set $\alpha = 0.1$ and $\beta = 0.05$ by default. More detailed implementation and settings for different datasets are illustrated in the following parts.

Dataset	Model	size (MB)	bit	size (MB)	float32	RQ	ZeroQ	GDFQ	DFQ	ACIQ	ZAQ
CIFAR10	MobileNetV2	9.0	W6A6	1.7	92.39	78.90	89.90	91.27	85.43	91.04	**92.15**
	VGG19	149	W4A8	25.1	93.49	92.42	92.69	92.84	92.66	92.48	**93.06**
CIFAR100	ResNet20	1.1	W5A5	0.2	69.58	49.54	65.7	66.12	59.42	60.19	**67.94**
	ResNet18	43	W4A4	5.4	77.38	17.00	70.25	71.53	40.35	54.73	**72.67**
ImageNet	MobileNetV2	14	W8A8	3.5	71.88	67.09	70.88	70.17	70.58	68.92	**71.43**
	ResNet50	98	W4A4	12.3	76.13	64.90	69.30	68.69	10.32	59.34	**70.06**
	ResNet50	98	W2A2	6.1	76.13	11.25	63.12	64.96	1.48	3.25	**65.52**

Table 1. Results of image classification on three datasets.

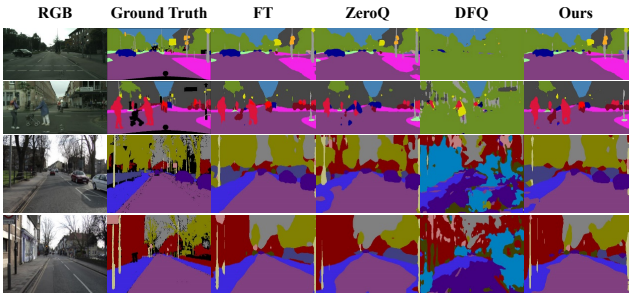


Figure 4. Visualization of segmentation results for Cityscapes (the first two rows) and CamVid (the last two rows).

Dataset	Method	W8A8	W6A6	W4A4	W2A2
Cityscapes (63.39)	FT	61.25	59.64	55.98	45.77
	RQ	58.42	55.33	29.16	0.44
	DFQ	57.34	55.29	19.06	3.13
	ZeroQ	59.52	57.97	52.73	43.18
	ZAQ	**60.18**	**58.12**	**55.12**	**44.93**
CamVid (53.34)	FT	52.76	50.75	49.13	40.06
	RQ	44.96	43.20	10.05	0.02
	DFQ	**51.02**	46.13	11.78	2.33
	ZeroQ	49.92	48.56	43.83	36.44
	ZAQ	50.89	**49.77**	**47.62**	**39.95**

Table 2. Results on Cityscapes and CamVid (mIoU).

4.2. Experimental Results

4.2.1 Performance Test for Image Classification

For image classification, we take the top-1 accuracy (abbr. Acc) as the metric. The number of fine-tuning epochs is 200 for CIFAR, while 300 for ImageNet. In each epoch, the training steps are set to 40 for CIFAR and 50 for ImageNet. The learning rates of SGD and Adam are decayed every 80 epochs for CIFAR and 100 for ImageNet. Besides, we use "W-A-" to denote the quantization bits used for weights (W) and activations (A), and "float32" as full-precision models.

Table 1 shows the classification results. First, we find DFQ and ACIQ suffer from dramatic performance degradation when taking ultra-low precision, especially for CIFAR100 and ImageNet. This verifies that due to the lack of fine-tuning, post-training quantization methods do not work well for ultra-low precision. Then we observe our framework achieves the best performance on the three classification datasets, indicating its advantages over the other quantization methods.

4.2.2 Performance Test for Image Segmentation

In this part, we mainly compare ZAQ with ZeroQ and DFQ on Cityscapes and CamVid, the images of which are all resized to 256. GDFQ requires labels as conditions to synthesize data, so it does not naturally support high-level vision tasks such as segmentation and detection. The ImageNet-pretrained MobileNetV2 and ResNet50 models are used as feature extractors within DeepLabv3 [6]. The hyperparameters $\alpha = 0.5$ and $\beta = 0.1$. We adopt mean IoU of all classes (mIoU) as the evaluation metric for segmentation. In fine-tuning, we set the size of the synthetic image as 128×128, which is enough for representing model discrepancy and transferring knowledge.

Table 2 shows the performance of quantized models fine-tuned by different methods, from which we can see that our method still exhibits superior performance, especially for ultra-low precision situations. This observation is consistent with what we find in image classification tasks.

Furthermore, we randomly select two real examples from Cityscapes and CamVid, respectively, and visualize the segmentation results of 4-bit DeeplabV3(MobileNetV2) learned by different model quantization methods. The results are shown in Figure 4, where the first two rows correspond to Cityscapes and the last two rows correspond to CamVid. Obviously, DFQ does not work properly for the examples in 4-bit quantization and thus it is hard to retain model performance. By comparing ZAQ with ZeroQ, we find it exhibits better qualitative results in complex details and small object segmentation, as shown in the second row of the figure.

4.2.3 Performance Test for Object Detection

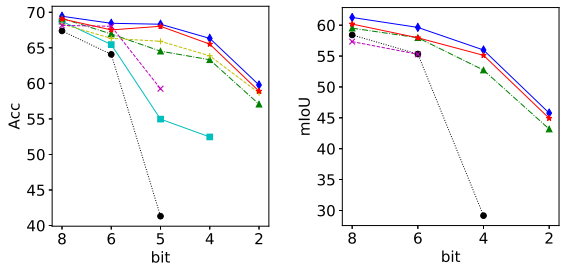
To demonstrate the application on object detection, we apply ZAQ to the model MobileNetV2 SSD [21] and evaluate it on VOC2012. Table 3 briefly demonstrates the advantages of

Method	W8A8	W4A8	W4A4	W2A2
FT	70.35	68.24	64.28	57.02
RQ	68.31	66.25	5.27	1.06
DFQ	69.16	64.57	13.15	2.65
ZeroQ	69.04	67.53	62.72	53.07
ZAQ	**70.02**	**68.12**	**64.44**	**56.96**

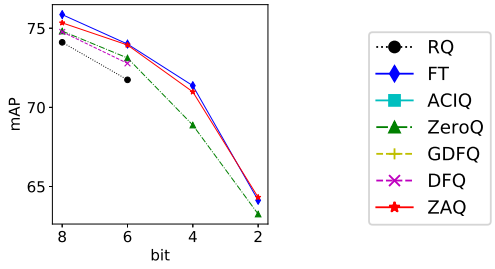
Table 3. Results of SSD(MobileNetV2) on VOC2012 (mAP).

our method compared to other quantization methods. In particular, ZAQ is comparable with FT that utilizes the original training dataset.

Finally, we end up the introduction of the performance tests for three image-based tasks with Figure 5, which provides an overview of how performance changes with different bits. The curves of different quantization methods in the figures reflect that ZAQ is consistently better and it gains greater improvements in ultra-low precision situation.



(a) ResNet18 on ImageNet (b) DeeplabV3 on Cityscapes

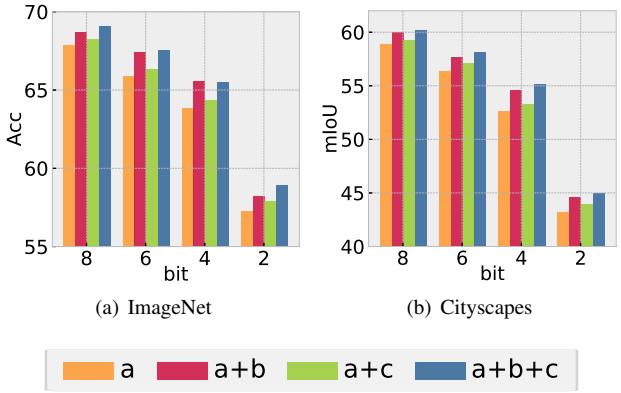


(c) SSD on VOC2012

Figure 5. Performance change versus different quantization precision.

4.2.4 Ablation Study

In this part, we conduct an ablation study to validate the contributions of the main components in ZAQ. First of all, Figure 6 presents the benefits of output discrepancy $\mathcal{D}_o$ ('a'), intermediate inter-channel discrepancy $\mathcal{D}_f$ ('b'), and activation regularization $\mathcal{L}_a$ ('c') on ImageNet (using model ResNet18) and Cityscapes (using model DeeplabV3(ResNet50)). Since output discrepancy $\mathcal{D}_o$ is directly associated with the final model output, it should not be removed anytime. As we can see, the intermediate inter-channel discrepancy could



(a) ImageNet (b) Cityscapes

Figure 6. Effectiveness of different components of the proposed ZAQ method. Noting that 'a':$\mathcal{D}_o$, 'b':$\mathcal{D}_f$, 'c':$\mathcal{L}_a$.

bring $1 \sim 2\%$ performance improvement, while the activation regularization has a smaller contribution of about 0.5% improvement. But it could prevent the generator from falling into some abnormal samples that are not sensitive to the full-precision models, which is also utilized in the previous study for KD [5].

Dataset	Model	bit	CRM	Gram	AT
CIFAR100	ResNet18	W4A4	72.67	45.32	61.80
Cityscapes	DeeplabV3	W8A8	52.17	41.36	48.65

Table 4. Ablation study of CRM.

We further demonstrate the effectiveness of CRM in model quantization by comparing it with two alternatives for learning intermediate knowledge: (1) Gram which directly uses Gram matrix in discrepancy modeling; (2) AT [45] which directly aligns normalized feature maps in knowledge transfer. Table 4 shows the performance of the above-mentioned methods, from which we can see CRM is much better than the other two methods. This verifies the necessity of considering different numerical spans in designing quantization-aware fine-tuning methods. In addition, we choose CIFAR100 to visualize the computed CRMs by ZAQ. In Figure 7, (a) and (b) are CRMs from the 2-nd exploited layer of full-precision and 4-bit ResNet18, respectively. By comparison, we can find the two CRMs are consistent with each other.

4.2.5 Efficiency Analysis

We conduct efficiency test on a single GPU (GTX 2080Ti) for ZAQ and the data generation-based quantization methods, *i.e.*, ZeroQ and GDFQ. The number of synthesized images determined for each method is conditioned on its performance convergence state following [4] and [42]. Due to the poor diversity of synthetic images, ZeroQ and GDFQ need to synthesize more samples in training. Besides, the images in Cityscapes have high resolution, making ZeroQ cost

too much time in synthesizing procedure. So we conduct the comparative experiment on Cityscapes with the same number of samples approximate to the original dataset in each epoch. Table 5 shows the results, where our method reduces GPU time by 41.8% compared to GDFQ on CIFAR100, while 57.5% compared to ZeroQ on Cityscapes. The conclusion is intuitive since ZeroQ needs 500 to 1500 iterations to generate per image and GDFQ is prone to generate redundant images.

Dataset	Method	images	GPU time
CIFAR100	ZeroQ	10000	5.5 h
	GDFQ	12800	7.6 h
	Ours	5120	3.2 h
Cityscpaes	ZeroQ	1280	12.7 h
	Ours	1280	5.4 h

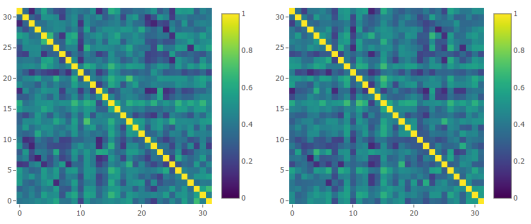
Table 5. Time cost comparison.

4.3. Case Study of Generated Images

This part conducts case studies on the generated data by different model quantization methods. We take CIFAR and CamVid for illustration. For CIFAR, the randomly selected images are shown in Figure 8. The first row of the images corresponds to CIFAR10 and the second row corresponds to CIFAR100. The first column shows the original images. The images in the middle three columns are gotten from MobileNetV2 (quantized to 8 bits) and the last column is from ResNet20 (quantized to 4 bits). By investigating the image patterns generated by GDFQ and ZeroQ, we find there is a big gap between them and those of the original images.

Although the image samples by ZAQ seem to be not recognizable by humans or be similar to the original data, their goal is to represent the discrepancy between two models with different precision. The comparison between the synthetic images of ZQA and those of GDFQ and ZeroQ indicates that ZQA could generate more diverse images, while GDFQ and ZeroQ suffer from more repeated patterns in their generated images. This empirically shows the efficiency of knowledge transfer in ZAQ.

Furthermore, we visualize the semantic image samples generated by ZAQ and ZeroQ in Figure 9. The observation



(a) full-precision model (b) quantization model

Figure 7. Visualization of CRMs on CIFAR100.

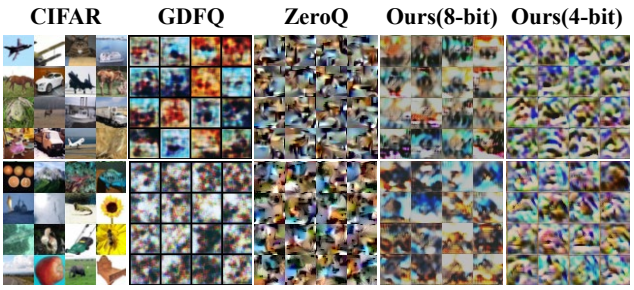


Figure 8. Generated samples about CIFAR.

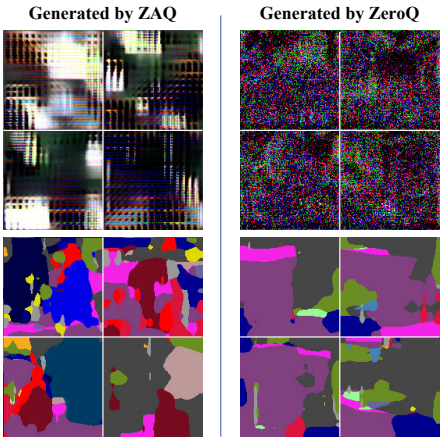


Figure 9. Sematic images about CamVid generated by ZAQ and ZeroQ based on DeeplabV3(MobileNetV2).

is accordant with what we have observed in image classification datasets. That is, ZAQ tends to generate semantic images with more diversity, while ZeroQ reconstructs semantic images with some duplicated local patterns.

5. Conclusion and Future Work

In this paper, we have proposed ZAQ, a novel zero-shot adversarial quantization framework without needing to access any original training data. Its main innovations lie in applying adversarial learning to data-free model quantization through alternating two-level discrepancy estimation and knowledge transfer. Our framework is welcomed for its ability of modeling prediction discrepancy, as well as intermediate inter-channel discrepancy between full-precision and quantized models. Extensive experiments on various deep neural models for three common vision tasks demonstrate the superiority of ZAQ, especially for ultra-low precision situations. In the future work, we consider applying the proposed method to other domains such as BERT quantization [38], and extending ZAQ to automatic mixed precision quantization.

Acknowledgement: This work was supported in part by National Natural Science Foundation of China under Grant (No. 62072182).

References

[1] Ron Banner, Itay Hubara, Elad Hoffer, and Daniel Soudry. Scalable methods for 8-bit training of neural networks. In *NeurIPS*, pages 5145–5153, 2018.

[2] Ron Banner, Yury Nahshan, Elad Hoffer, and Daniel Soudry. Aciq: analytical clipping for integer quantization of neural networks. In *ICLR*, 2018.

[3] Gabriel J Brostow, Jamie Shotton, Julien Fauqueur, and Roberto Cipolla. Segmentation and recognition using structure from motion point clouds. In *ECCV*, pages 44–57. Springer, 2008.

[4] Yaohui Cai, Zhewei Yao, Zhen Dong, Amir Gholami, Michael W Mahoney, and Kurt Keutzer. Zeroq: A novel zero shot quantization framework. In *CVPR*, pages 13169–13178, 2020.

[5] Hanting Chen, Yunhe Wang, Chang Xu, Zhaohui Yang, Chuanjian Liu, Boxin Shi, Chunjing Xu, Chao Xu, and Qi Tian. Data-free learning of student networks. In *ICCV*, pages 3514–3522, 2019.

[6] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[7] Jungwook Choi, Zhuo Wang, Swagath Venkataramani, Pierce I-Jen Chuang, Vijayalakshmi Srinivasan, and Kailash Gopalakrishnan. Pact: Parameterized clipping activation for quantized neural networks. *arXiv preprint arXiv:1805.06085*, 2018.

[8] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *CVPR*, 2016.

[9] Dipankar Das, Naveen Mellempudi, Dheevatsa Mudigere, Dhiraj Kalamkar, Sasikanth Avancha, Kunal Banerjee, Srinivas Sridharan, Karthik Vaidyanathan, Bharat Kaul, Evangelos Georganas, et al. Mixed precision training of convolutional neural networks using integer operations. In *ICLR*, 2018.

[10] Lei Deng, Guoqi Li, Song Han, Luping Shi, and Yuan Xie. Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proceedings of the IEEE*, 108(4):485–532, 2020.

[11] Yinpeng Dong, Hang Su, Jun Zhu, and Fan Bao. Towards interpretable deep neural networks by leveraging adversarial examples. *arXiv preprint arXiv:1708.05493*, 2017.

[12] Mark Everingham and John Winn. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Analysis, Statistical Modelling and Computational Learning, Tech. Rep*, 8, 2011.

[13] Gongfan Fang, Jie Song, Chengchao Shen, Xinchao Wang, Da Chen, and Mingli Song. Data-free adversarial distillation. In *CVPR*, 2020.

[14] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NeurIPS*, pages 2672–2680, 2014.

[16] 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 *CVPR*, pages 2704–2713, 2018.

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] Urs Köster, Tristan Webb, Xin Wang, Marcel Nassar, Arjun K Bansal, William Constable, Oguz Elibol, Scott Gray, Stewart Hall, Luke Hornof, et al. Flexpoint: An adaptive numerical format for efficient training of deep neural networks. In *NeurIPS*, pages 1742–1752, 2017.

[19] Alex Krizhevsky. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012.

[20] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pages 1097–1105, 2012.

[21] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016.

[22] Yufan Liu, Jiajiong Cao, Bing Li, Chunfeng Yuan, Weiming Hu, Yangxi Li, and Yunqiang Duan. Knowledge distillation via instance relationship graph. In *CVPR*, pages 7096–7104, 2019.

[23] Yuang Liu, Wei Zhang, and Jun Wang. Adaptive multi-teacher multi-level knowledge distillation. *Neurocomputing*, 415:106–113, 2020.

[24] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, page 3431–3440, 2015.

[25] Raphael Gontijo Lopes, Stefano Fenu, and Thad Starner. Data-free knowledge distillation for deep neural networks. In *NeurIPS*, 2017.

[26] Jeffrey L McKinstry, Steven K Esser, Rathinakumar Appuswamy, Deepika Bablani, John V Arthur, Izzet B Yildiz, and Dharmendra S Modha. Discovering low-precision networks close to full-precision networks for efficient embedded inference. *arXiv preprint arXiv:1809.04191*, 2018.

[27] Eldad Meller, Alexander Finkelstein, Uri Almog, and Mark Grobman. Same, same but different: Recovering neural network quantization error through weight factorization. In *ICML*, pages 4486–4495, 2019.

[28] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. In *ICLR*, 2018.

[29] Szymon Migacz. 8-bit inference with tensorrt. In *GPU technology conference*, volume 2, page 5, 2017.

[30] Markus Nagel, Mart van Baalen, Tijmen Blankevoort, and Max Welling. Data-free quantization through weight equalization and bias correction. In *ICCV*, pages 1325–1334, 2019.

[31] Gaurav Kumar Nayak, Konda Reddy Mopuri, Vaisakh Shaj, Venkatesh Babu Radhakrishnan, and Anirban Chakraborty. Zero-shot knowledge distillation in deep networks. In *ICML*, pages 4743–4751, 2019.

[32] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *CVPR*, pages 3967–3976, 2019.

[33] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[34] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE TPAMI*, 39(6):1137–1149, 2017.

[35] Adriana Romero, Nicolas Ballas, Samira Ebrahimi Kahou, Antoine Chassang, Carlo Gatta, and Yoshua Bengio. Fitnets: Hints for thin deep nets. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015.

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

[37] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018.

[38] 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 *AAAI*, pages 8815–8821, 2020.

[39] Suraj Srinivas and R Venkatesh Babu. Data-free parameter pruning for deep neural networks. *arXiv preprint arXiv:1507.06149*, 2015.

[40] Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *CVPR*, pages 1653–1660, 2014.

[41] Jiaxiang Wu, Cong Leng, Yuhang Wang, Qinghao Hu, and Jian Cheng. Quantized convolutional neural networks for mobile devices. In *CVPR*, pages 4820–4828, 2016.

[42] Shoukai Xu, Haokun Li, Bohan Zhuang, Jing Liu, Jiezhang Cao, Chuangrun Liang, and Mingkui Tan. Generative low-bitwidth data free quantization. In *ECCV*, 2020.

[43] Jingwen Ye, Yixin Ji, Xinchao Wang, Xin Gao, and Mingli Song. Data-free knowledge amalgamation via group-stack dual-gan. In *CVPR*, pages 12516–12525, June 2020.

[44] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *CVPR*, pages 8715–8724, 2020.

[45] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In *ICLR*, 2017.

[46] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. Springer, 2014.

[47] Ritchie Zhao, Yuwei Hu, Jordan Dotzel, Chris De Sa, and Zhiru Zhang. Improving neural network quantization without retraining using outlier channel splitting. In *ICML*, pages 7543–7552, 2019.

[48] Shuchang Zhou, Yuxin Wu, Zekun Ni, Xinyu Zhou, He Wen, and Yuheng Zou. Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients. *arXiv preprint arXiv:1606.06160*, 2016.

[49] Feng Zhu, Ruihao Gong, Fengwei Yu, Xianglong Liu, Yanfei Wang, Zhelong Li, Xiuqi Yang, and Junjie Yan. Towards unified int8 training for convolutional neural network. In *CVPR*, pages 1969–1979, 2020.