

# DEFENSIVE QUANTIZATION: WHEN EFFICIENCY MEETS ROBUSTNESS

**Ji Lin**  
MIT  
jilin@mit.edu

**Chuang Gan**  
MIT-IBM Watson AI Lab  
ganchuang@csail.mit.edu

**Song Han**  
MIT  
songhan@mit.edu

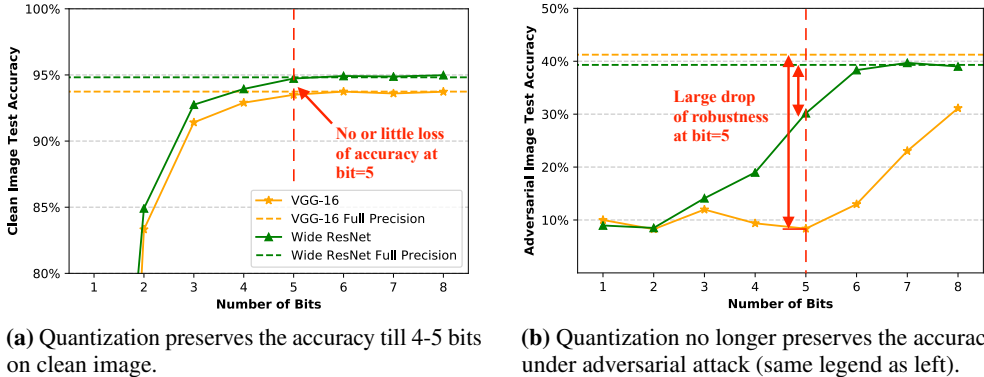
## ABSTRACT

Neural network quantization is becoming an industry standard to efficiently deploy deep learning models on hardware platforms, such as CPU, GPU, TPU, and FPGAs. However, we observe that the conventional quantization approaches are vulnerable to adversarial attacks. This paper aims to raise people’s awareness about the security of the quantized models, and we designed a novel quantization methodology to jointly optimize the efficiency and robustness of deep learning models. We first conduct an empirical study to show that vanilla quantization suffers more from adversarial attacks. We observe that the inferior robustness comes from the *error amplification effect*, where the quantization operation further enlarges the distance caused by amplified noise. Then we propose a novel Defensive Quantization (DQ) method by controlling the Lipschitz constant of the network during quantization, such that the magnitude of the adversarial noise remains non-expansive during inference. Extensive experiments on CIFAR-10 and SVHN datasets demonstrate that our new quantization method can defend neural networks against adversarial examples, and even achieves superior robustness than their full-precision counterparts, while maintaining the same hardware efficiency as vanilla quantization approaches. As a by-product, DQ can also improve the accuracy of quantized models without adversarial attack.

## 1 INTRODUCTION

Neural network quantization (Han et al., 2015; Zhu et al., 2016; Jacob et al., 2017) is a widely used technique to reduce the computation and memory costs of neural networks, facilitating efficient deployment. It has become an industry standard for deep learning hardware. However, we find that the widely used vanilla quantization approaches suffer from unexpected issues — the quantized model is more vulnerable to adversarial attacks (Figure 1). Adversarial attack is consist of subtle perturbations on the input images that causes the deep learning models to give incorrect labels (Szegedy et al., 2013; Goodfellow et al.). Such perturbations are hardly detectable by human eyes but can easily fool neural networks. Since quantized neural networks are widely deployed in many safety-critical scenarios, *e.g.*, autonomous driving (Amodei et al., 2016), the potential security risks cannot be neglected. The efficiency and latency in such applications are also important, so we need to jointly optimize them.

The fact that quantization leads to inferior adversarial robustness is counter intuitive, as small perturbations should be denoised with low-bit representations. Recent work (Xu et al., 2017) also demonstrates that quantization on input image space, *i.e.* color bit depth reduction, is quite effective to defend adversarial examples. A natural question then rises, why the quantization operator is yet effective when applied to intermediate DNN layers? We analyze that such issue is caused by the error amplification effect of adversarial perturbation (Liao et al., 2018) — although the magnitude of perturbation on the image is small, it is amplified significantly when passing through deep neural network (see Figure 3b). The deeper the layers are, the more significant such side effect is. Such amplification pushes values into a different quantization bucket, which is undesirable. We conducted empirical experiments to analyze how quantization influences the activation error between clean and adversarial samples (Figure 3a): when the magnitude of the noise is small, activation quantization is capable of reducing the errors by eliminating small perturbations; However, when the magnitude of perturbation is larger than certain threshold, quantization instead amplify the errors, which causes



**Figure 1.** Quantized neural network are more vulnerable to adversarial attack. Quantized models have no loss of accuracy on clean image ( $\geq 5$  bits), but have significant loss of accuracy under adversarial attack compared to full precision models. Setup: VGG-16 and Wide ResNet on the test set of CIFAR-10 with FGSM attack.

the quantized model to make mistakes. We argue that this is the main reason causing the inferior robustness of the quantized models.

In this paper, we propose Defensive Quantization (DQ) that not only fixes the robustness issue of quantized models, but also turns activation quantization into a defense method that further boosts adversarial robustness. We are inspired by the success of image quantization in improving robustness. Intuitively, it will be possible to defend the attacks with quantization operations if we can keep the magnitude of the perturbation small. However, due to the error amplification effect of gradient based adversarial samples, it is non-trivial to keep the noise at a small scale during inference. Recent works (Cisse et al., 2017; Qian & Wegman, 2018) have attempted to make the network *non-expansive* by controlling the Lipschitz constant of the network to be smaller than 1, which has smaller variation change in its output than its input. In such case, the input noise will not propagate through the intermediate layers and impact the output, but attenuated. Our method is built on the theory. Defensive quantization not only quantizes feature maps into low-bit representations, but also controls the Lipschitz constant of the network, such that the noise is kept within a small magnitude for all hidden layers. In such case, we keep the noise small, as in the left zone of Figure 3a, quantization can reduce the perturbation error. The produced model with our method enjoys better security and efficiency at the same time.

Experiments show that Defensive Quantization (DQ) offers three unique advantages. First, DQ provides an effective way to boost the robustness of deep learning models while maintains the efficiency. Second, DQ is a generic building block of adversarial defense, which can be combined with other adversarial defense techniques to advance state-of-the-art robustness. Third, our method makes quantization itself easier thanks to the constrained dynamic range.

## 2 BACKGROUND AND RELATED WORK

### 2.1 MODEL QUANTIZATION

Neural network quantization (Han et al., 2015; Rastegari et al., 2016; Zhou et al., 2016; Courbariaux & Bengio, 2016; Zhu et al., 2016) are widely adopted to enable efficient inference. By quantizing the network into low-bit representation, the inference of network requires less computation and less memory, while still achieves little accuracy degradation on clean images. However, we find that the quantized models suffer from severe security issues — they are more vulnerable against adversarial attacks compared to full-precision models, even when they have the same clean image accuracy. Adversarial perturbation is applied to the input image, thus it is most related to activation quantization (Dhillon et al., 2018). We carry out the rest of the paper using ReLU6 based activation quantization (Howard et al., 2017; Sandler et al., 2018), as it is computationally efficient and is widely adopted by modern frameworks like TensorFlow (Abadi et al., 2016). As illustrated in Figure 2, a quantized convolutional network is composed of several quantized convolution block, each containing a serial of *conv* + *BN* + *ReLU6* + *linear quantize* operators. As the quantization operator has

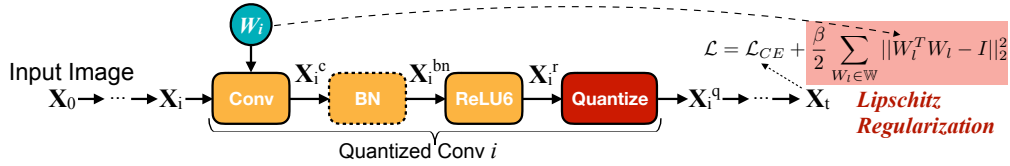


Figure 2. Defensive quantization with Lipschitz regularization.

0 gradient almost everywhere, we followed common practice to use a STE (Bengio et al., 2013) function  $y = x + \text{stop\_gradient}[\text{Quantize}(x) - x]$  for gradient computation, which also eliminates the obfuscated gradient problem (Athalye et al., 2018).

Our work bridges two domains: model quantization and adversarial defense. Previous work (Galloway et al., 2017) claims binary quantized networks can improve the robustness against some attacks. However, the improvement is not substantial and they used randomized quantization, which is not practical for real deployment (need extra random number generators in hardware). It also causes one of the obfuscated gradient situations (Athalye et al., 2018): stochastic gradients, leading to a false sense of security. Rakin et al. (2018) tries to use quantization as an effective defense method. However, they employed Tanh-based quantization, which is not hardware friendly on fixed-point units due to the large overhead accessing look-up table. Even worse, according to our re-implementation, their method actually leads to severe gradient masking problem (Papernot et al., 2016a) during adversarial training, due to the nature of Tanh function (see A.1 for detail). As a result, the actual robustness of this work under black-box attack has no improvement over full-precision model and is even worse. Therefore, there is no previous work that are conducted under real inference setting to study the quantized robustness for both black-box and white-box. Our work aim to raise people’s awareness about the security of the actual deployed models.

## 2.2 ADVERSARIAL ATTACKS & DEFENSES

Given an image  $\mathbf{X}$ , an adversarial attack method tries to find a small perturbation  $\Delta$  with constraint  $\|\Delta\| \leq \epsilon$ , such that the neural network gives different outputs for  $\mathbf{X}$  and  $\mathbf{X}^{adv} \triangleq \mathbf{X} + \Delta$ . Here  $\epsilon$  is a scalar to constrain the norm of the noise (e.g.,  $\epsilon = 8$  is commonly used when we represent colors from 0-255), so that the perturbation is hardly visible to human. For this paper we choose to study attacks defined under  $\|\cdot\|_\infty$ , where each element of the image can vary at most  $\epsilon$  to form an adversary. We introduce several attack and defense methods used in our work in the following sections.

### 2.2.1 ATTACK METHODS

**Random Perturbation (Random)** Random perturbation attack adds a uniform sampled noise within  $[-\epsilon, \epsilon]$  to the image, The method has no prior knowledge of the data and the network, thus is considered as the weakest attack method.

**Fast Gradient Sign Method (FGSM) & R+FGSM** Goodfellow et al. proposed a fast method to calculate the adversarial noise by following the direction of the loss gradient  $\nabla_{\mathbf{X}} L(\mathbf{X}, y)$ , where  $L(\mathbf{X}, y)$  is the loss function for training (e.g. cross entropy loss). The adversarial samples are computed as:

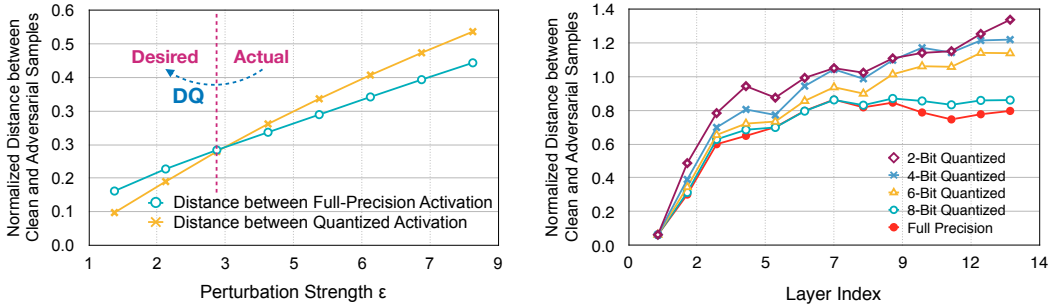
$$\mathbf{X}^{adv} = \mathbf{X} + \epsilon * \text{sign}(\nabla_{\mathbf{X}} L(\mathbf{X}, y)) \tag{1}$$

As FGSM is an one-step gradient-based method, it can suffer from sharp curvature near the data points, leading a false direction of ascent. Therefore, Tramèr et al. (2017) proposes to prepend FGSM by a random step to escape the non-smooth vicinity. The new method is called **R+FGSM**, defined as follows, for parameters  $\epsilon$  and  $\epsilon_1$  (where  $\epsilon_1 < \epsilon$ ):

$$\mathbf{X}^{adv} = \mathbf{X}' + (\epsilon - \epsilon_1) * \text{sign}(\nabla_{\mathbf{X}} L(\mathbf{X}, y)), \text{ where } \mathbf{X}' = \mathbf{X} + \epsilon_1 * \text{sign}(\mathcal{N}(\mathbf{0}^d, \mathbf{I}^d)). \tag{2}$$

In our paper, we set  $\epsilon_1 = \epsilon/2$  following Tramèr et al. (2017).

**Basic Iterative Method (BIM) & Projected Gradient Descend (PGD)** Kurakin et al. (2016) suggests a simple yet much stronger variant of FGSM by applying it multiple times iteratively with a



(a) Noise increases with perturbation strength. Quantization makes the slope deeper.

(b) With conventional quantization, noise increases with layer index (the amplification effect).

**Figure 3.** (a) Comparison of the noise introduced by adversarial-attack, with and without quantization. For small perturbation, quantization reduces the noise; for large perturbation, quantization magnifies the noise. (b) The noise amplification effect: the noise is amplified with layer index. Setup: conventional activation quantization for VGG-16, normalized difference of full-precision and low-precision activation.

small step size  $\alpha$ . The method is called BIM, defined as:

$$\mathbf{X}_0^{adv} = \mathbf{X}, \mathbf{X}_{n+1}^{adv} = \text{clip}_{\mathbf{X}}^{\epsilon} \{ \mathbf{X}_n^{adv} + \alpha \text{sign}(\nabla_{\mathbf{X}} L(\mathbf{X}_n^{adv}, y)) \}, \quad (3)$$

where  $\text{clip}_{\mathbf{X}}^{\epsilon}$  means clipping the result image to be within the  $\epsilon$ -ball of  $\mathbf{X}$ . In (Madry et al., 2017), the BIM is preceded by a random start as in R+FGSM method. The resulting attack is called PGD, which proves to be a general first-order attack. In our experiments we used PGD for comprehensive experiments as it proves to be one of the strongest attack. Unlike Madry et al. (2017) that uses a fixed  $\epsilon$  and  $\alpha$ , we follow Kurakin et al. (2016); Song et al. (2017) to use  $\alpha = 1$  and number of iterations of  $\lfloor \min(\epsilon + 4, 1.25\epsilon) \rfloor$ , so that we can test the model’s robustness under different strength of attacks.

### 2.2.2 DEFENSE METHODS

Current defense methods either preprocess the adversarial samples to denoise the perturbation (Xu et al., 2017; Song et al., 2017; Liao et al., 2018) or making the network itself robust (Warde-Farley & Goodfellow, 2016; Papernot et al., 2016b; Madry et al., 2017; Kurakin et al., 2016; Goodfellow et al.; Tramèr et al., 2017). Here we introduced several defense methods related to our experiments.

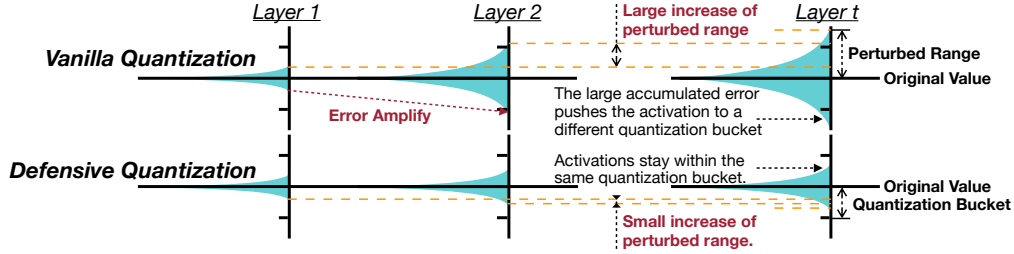
**Feature Squeezing** Xu et al. (2017) proposes to detect adversarial images by squeezing the input image. Specifically, the image is processed with color depth bit reduction (5 bits for our experiments) and smoothed by a  $2 \times 2$  median filter. If the low resolution image is classified differently as the original image, then this image is detected as adversarial.

**Adversarial Training** Adversarial training (Madry et al., 2017; Kurakin et al., 2016; Goodfellow et al.; Tramèr et al., 2017) is currently the strongest method for defense. By augmenting the training set with adversarial samples, the network learns to classify adversarial samples correctly. As adversarial FGSM can easily lead to gradient masking effect (Papernot et al., 2016a), we study adversarial R+FGSM as in (Tramèr et al., 2017). We also experimented with PGD training (Madry et al., 2017).

Experiments show that above defense methods can be combined with our DQ method to further improve the robustness. The robustness has been tested under the aforementioned attack methods.

## 3 CONVENTIONAL NN QUANTIZATION IS NOT ROBUST

Conventional neural network quantization is more vulnerable to adversarial attacks. We experimented with VGG-16 (Simonyan & Zisserman, 2014) and a Wide ResNet (Zagoruyko & Komodakis, 2016) of depth 28 and width 10 on CIFAR-10 (Krizhevsky & Hinton, 2009) dataset. We followed the training protocol as in (Zagoruyko & Komodakis, 2016). Adversarial samples are generated with a FGSM (Goodfellow et al.) attacker ( $\epsilon = 8$ ) on the entire test set. As in Figure 1, the clean image accuracy doesn’t significantly drop until the model is quantized to 4 bits (Figure 1a). However, under adversarial attack, even with 5-bit quantization, the accuracy drastically decreased by 25.3% and 9.2% respectively. Although the full precision model’s accuracy has dropped, the quantized model’s



**Figure 4.** The error amplification effect prevents activation quantization from defending adversarial attacks.

accuracy dropped much harder, showing that the conventional quantization method is not robust. Clean image accuracy used to be the sole figure of merit to evaluate a quantized model. We show that even when the quantized model has no loss of performance on clean images, it can be much more easily fooled compared to full-precision ones, thus raising security concerns.

Input image quantization, *i.e.*, color bit depth reduction is an effective defense method (Xu et al., 2017). Counter intuitively, it does not work when applied to hidden layers, and even make the robustness worse. To understand the reason, we studied the effect of quantization w.r.t. different perturbation strength. We first randomly sample 128 images  $\mathbf{X}$  from the test set of CIFAR-10, and generate corresponding adversarial samples  $\mathbf{X}^{adv}$ . The samples are then fed to the trained Wide ResNet model. To mimic different strength of activation perturbation, we vary the  $\epsilon$  from 1 to 8. We inspected the activation after the first convolutional layer  $f_1$  (Conv + BN + ReLU6), denoted as  $\mathbf{A}_1 = f_1(\mathbf{X})$  and  $\mathbf{A}_1^{adv} = f_1(\mathbf{A}^{adv})$ . To measure the influence of perturbation, we define a normalized distance between clean and perturbed activation as:

$$D(\mathbf{A}, \mathbf{A}^{adv}) = \|\mathbf{A} - \mathbf{A}^{adv}\|_2 / \|\mathbf{A}\|_2 \quad (4)$$

We compare  $D(\mathbf{A}, \mathbf{A}^{adv})$  and  $D(\text{Quantize}(\mathbf{A}), \text{Quantize}(\mathbf{A}^{adv}))$ , where Quantize indicates uniform quantization with 3 bits. The results are shown in Figure 3a. We can see that only when  $\epsilon$  is small, quantization helps to reduce the distance by removing small magnitude perturbations. The distance will be enlarged when  $\epsilon$  is larger than 3.

The above experiment explains the inferior robustness of the quantized model. We argue that such issue arises from the *error amplification effect* (Liao et al., 2018), where the relative perturbed distance will be amplified when the adversarial samples are fed through the network. As illustrated in Figure 4, the perturbation applied to the *input image* has very small magnitude compared to the image itself ( $\pm 8$  versus  $0 - 255$ ), corresponding to the left zone of Figure 3a (desired), where quantization helps to denoise the perturbation. Nevertheless, the difference in activation is amplified as the inference carries on. If the perturbation after amplification is large enough, the situation corresponds to the right zone (actual) of Figure 3a, where quantization further increases the normalized distance. Such phenomenon is also observed in the quantized VGG-16. We plot the normalized distance of each convolutional layer’s input in Figure 3b. The fewer bits in the quantized model, the more severe the amplification effect.

## 4 DEFENSIVE QUANTIZATION

Given the robustness limitation of conventional quantization technique, we propose Defensive Quantization (DQ) to defend the adversarial examples for quantized models. DQ suppresses the noise amplification effect, keeping the magnitude of the noise small, so that we can arrive at the left zone (Figure 3a, desired) where quantization helps robustness instead of making it worse.

We control the neural network’s *Lipschitz constant* (Szegedy et al., 2013; Bartlett et al., 2017; Cisse et al., 2017) to suppress network’s amplification effect. Lipschitz constant describes: when input changes, how much does the output change correspondingly. For a function  $f : X \rightarrow Y$ , if it satisfies

$$D_Y(f(x_1), f(x_2)) \leq k D_X(x_1, x_2), \quad \forall x_1, x_2 \in X \quad (5)$$

for a real-valued  $k \geq 0$  and some metrics  $D_X$  and  $D_Y$ , then we call  $f$  Lipschitz continuous and  $k$  is the known as the Lipschitz constant of  $f$ . If we consider a network  $f$  with clean inputs  $x_1 = \mathbf{X}$  and corresponding adversarial inputs  $x_2 = \mathbf{X}_{adv}$ , the error amplification effect can be controlled if we

**Table 1.** The clean and adversarial accuracy of Wide ResNet on CIFAR-10 test set. We compare the accuracy of full-precision and quantized models. With our DQ method, we not only eliminate the robustness gap between full-precision and quantized models, but also improve the robustness over full-precision ones. The accuracy gain from quantization compared to the full-precision model (Quantize Gain) has gradually been improved as  $\beta$  increases. **Bold** and underline numbers are the first and second highest accuracy at each row.

| Method                | Acc.  | Full<br>Prec. | Quantize Bit |             |      |      |             | Quantize<br>Gain | Best<br>Acc. |
|-----------------------|-------|---------------|--------------|-------------|------|------|-------------|------------------|--------------|
|                       |       |               | 1            | 2           | 3    | 4    | 5           |                  |              |
| Vanilla               | clean | 94.8          | 42.0         | 84.9        | 92.8 | 93.9 | 94.7        | -9.1             | <b>39.3</b>  |
|                       | adv.  | <b>39.3</b>   | 9.0          | 8.5         | 14.1 | 19.0 | <u>30.2</u> |                  |              |
| DQ ( $\beta = 3e-4$ ) | clean | 95.3          | 95.2         | 95.3        | 95.1 | 95.1 | +1.4        | <b>43.2</b>      |              |
|                       | adv.  | <u>41.8</u>   | <b>43.2</b>  | 41.1        | 40.1 | 39.8 |             |                  | 39.3         |
| DQ ( $\beta = 6e-4$ ) | clean | 95.6          | 95.7         | 95.2        | 95.4 | 95.6 | 95.5        | +2.6             | <b>48.3</b>  |
|                       | adv.  | 45.7          | <b>48.3</b>  | <u>47.9</u> | 43.8 | 43.9 | 44.6        |                  |              |
| DQ ( $\beta = 1e-3$ ) | clean | 95.9          | 95.8         | 95.6        | 95.6 | 95.8 | 95.8        | +2.7             | <b>51.8</b>  |
|                       | adv.  | 49.1          | <u>51.3</u>  | 50.4        | 51.3 | 49.8 | <b>51.8</b> |                  |              |

have a small Lipschitz constant  $k$  (in optimal situation we can have  $k \leq 1$ ). In such case, the error introduced by adversarial perturbation will not be amplified, but reduced. Specifically, we consider a feed-forward network composed of a serial of functions:

$$f(x) = (\phi_l \circ \phi_{l-1} \circ \dots \circ \phi_1)(x) \quad (6)$$

where  $\phi_i$  can be a linear layer, convolutional layer, pooling, activation functions, *etc.* Denote the Lipschitz constant of a function  $f$  as  $Lip(f)$ , then for the above network we have

$$Lip(f) \leq \prod_{i=1}^L Lip(\phi_i) \quad (7)$$

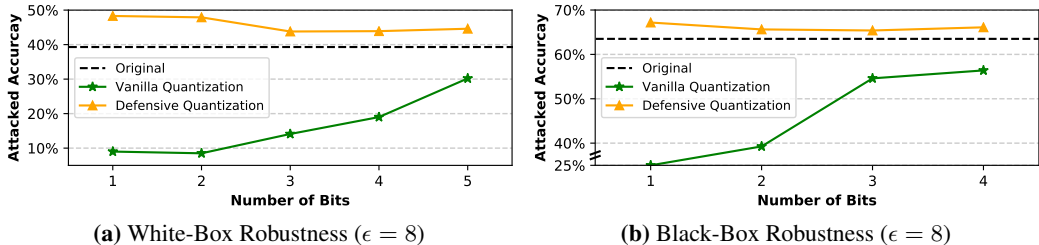
As the Lipschitz constant of the network is the product of its individual layers' Lipschitz constants,  $Lip(f)$  can grow exponentially if  $Lip(\phi_i) > 1$ . This is the common case for normal network training (Cisse et al., 2017), and thus the perturbation will be amplified for such a network. Therefore, to keep the Lipschitz constant of the whole network small, we need to keep the Lipschitz constant of each layer  $Lip(\phi_i) \leq 1$ . We call a network with  $Lip(\phi_i) \leq 1, \forall i = 1, \dots, L$  a non-expansive network.

We describe a regularization term to keep the Lipschitz constant small. Let us first consider linear layers with weight  $\mathbf{W} \in \mathbb{R}^{c_{out} \times c_{in}}$  under  $\|\cdot\|_2$  norm. The Lipschitz constant is by definition the spectral norm of  $W$ :  $\rho(\mathbf{W})$ , *i.e.*, the maximum singular value of  $\mathbf{W}$ . Computing the singular values of each weight matrix is not computationally feasible during training. Luckily, if we can keep the weight matrix row orthogonal, the singular values are by nature equal to 1, which meets our non-expansive requirements. Therefore we transform the problem of keeping  $\rho(\mathbf{W}) \leq 1$  into keeping  $\mathbf{W}^T \mathbf{W} \approx \mathbf{I}$ , where  $\mathbf{I}$  is the identity matrix. Naturally, we introduce a regularization term  $\|\mathbf{W}^T \mathbf{W} - \mathbf{I}\|$ , where  $\mathbf{I}$  is the identity matrix. Following (Cisse et al., 2017), for convolutional layers with weight  $\mathbf{W} \in \mathbb{R}^{c_{out} \times c_{in} \times k \times k}$ , we can view it as a two-dimension matrix of shape  $\mathbf{W} \in \mathbb{R}^{c_{out} \times (c_{in} k k)}$  and apply the same regularization. The final optimization objective is:

$$\mathcal{L} = \mathcal{L}_{CE} + \frac{\beta}{2} \sum_{\mathbf{W}_l \in \mathbb{W}} \|\mathbf{W}_l^T \mathbf{W}_l - \mathbf{I}\|^2 \quad (8)$$

where  $\mathcal{L}_{CE}$  is the original cross entropy loss and  $\mathbb{W}$  denotes all the weight matrices of the neural network.  $\beta$  is the weighting to adjust the relative importance. The above discussion is based on simple feed forward networks. For ResNets in our experiments, we also follow Cisse et al. (2017) to modify the aggregation layer as a convex combination of their inputs, where the 2 coefficients are updated using specific projection algorithm (see (Cisse et al., 2017) for details).

Our Defensive Quantization is illustrated in Figure 2. The key part is the regularization term, which suppress the noise amplification effect by regularizing the Lipschitz constant. As a result, the



**Figure 5.** The white-box and black-box robustness are consistent: vanilla quantization leads to significant robustness drop, while DQ can bridge the gap and improve the robustness, especially with lower bits (bit=1). Setup: white-box and black-box robustness of Wide ResNet with vanilla quantization and defensive quantization.

perturbation at each layer is kept within a certain range, the adversarial noise won’t propagate. Our method not only fixes the drop of robustness induced by quantization, but also takes quantization as a defense method to further increase the robustness. Therefore it’s named Defensive Quantization.

## 5 EXPERIMENTS

Our experiments demonstrate the following advantages of Defensive Quantization. First, DQ can retain the robustness of a model when quantized with low-bit. Second, DQ is a general and effective defense method under various scenarios, thus can be combined with other defensive techniques to further advance state-of-the-art robustness. Third, as a by-product, DQ can also improve the accuracy of training quantized models on clean images without attacks, since it limits the dynamic range.

### 5.1 FIXING ROBUSTNESS DROP

**Setup:** We conduct experiments with Wide ResNet (Zagoruyko & Komodakis, 2016) of  $28 \times 10$  on the CIFAR-10 dataset (Krizhevsky & Hinton, 2009) using ReLU6 based activation quantization, with number of bits ranging from 1 to 5. All the models are trained following (Zagoruyko & Komodakis, 2016) with momentum SGD for 200 epochs. The adversarial samples are generated using FGSM attacker with  $\epsilon = 8$ .

**Result:** The results are presented in Table 1. For vanilla models, though the adversarial robustness increases with the number of bits, *i.e.*, the models closer to full-precision one has better robustness, the best quantized model still has inferior robustness by  $-9.1\%$ . While with our Defensive Quantization, the quantized models have better robustness than full-precision counterparts. The robustness is better when the number of bits are small, since it can de-noise larger adversarial perturbations. We also find that the robustness is generally increasing as  $\beta$  gets larger, since the regularization of Lipschitz constant itself keeps the noise smaller at later layers. At the same time, the quantized models consistently achieve better robustness. The robustness of quantized model also increases with  $\beta$ . We conduct a detailed analysis of the effect of  $\beta$  in Section B. The conclusion is: (1) conventional quantized models are less robust. (2) Lipschitz regularization makes the model robust. (3) Lipschitz regularization + quantization makes model even more robust.

As shown in (Athalye et al., 2018), many of the defense methods actually lead to obfuscated gradient, providing a false sense of security. Therefore it is important to check the model’s robustness under black-box attack. We separately trained a substitute VGG-16 model on the same dataset to generate adversarial samples, as it was proved to have the best transferability (Su et al., 2018). The results are presented in Figure 5. Trends of white-box and black-box attack are consistent. Vanilla quantization leads to inferior black-box robustness, while with our method can further improve the models’ robustness. As the robustness gain is consistent for both white-box and black-box setting, our method does not suffer from gradient masking.

### 5.2 DEFEND WITH DEFENSIVE QUANTIZATION

In this section, we show that we can combine Defensive Quantization with other defense techniques to achieve state-of-the-art robustness.



**Table 2.** SVHN experiments tested with  $\epsilon = 2/8/16$ . **(B)** indicates black-box attack.

| Training Technique             | Clean       | Random          | FGSM            | R+FGSM          | PGD             | FGSM(B)         | PGD(B)          |
|--------------------------------|-------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Normal                         | <b>96.8</b> | <b>97/97/96</b> | 74/42/26        | 86/55/35        | 77/05/00        | 90/62/40        | 92/58/30        |
| Normal + <b>DQ</b>             | 96.7        | 96/96/96        | 77/45/31        | 87/59/40        | 79/10/00        | 90/63/42        | 92/60/31        |
| Feature Squeezing              | 96.3        | 96/96/95        | 69/34/20        | 84/48/28        | 72/03/00        | 89/61/38        | 91/56/27        |
| Feature Squeezing + <b>DQ</b>  | 96.2        | 96/96/96        | 75/42/28        | 86/56/36        | 77/06/00        | 90/63/40        | 92/59/30        |
| Adversarial R+FGSM             | 96.6        | <b>97/96/96</b> | 84/53/38        | 91/70/57        | 87/30/06        | 93/74/54        | 94/83/72        |
| Adversarial R+FGSM + <b>DQ</b> | 96.6        | <b>97/96/95</b> | <b>88/59/40</b> | <b>93/77/61</b> | <b>91/47/17</b> | <b>94/80/61</b> | <b>95/89/84</b> |

**Setup:** We conducted experiments on the Street View House Number dataset (SVHN) (Netzer et al., 2011) and CIFAR-10 dataset (Krizhevsky & Hinton, 2009). Since adversarial training is time consuming, we only use the official training set for experiments. CIFAR-10 is another widely used dataset containing 50,000 training samples and 10,000 testing samples of size  $32 \times 32$ . For both datasets, we divide the pixel values by 255 as a pre-processing step.

Following (Athalye et al., 2018; Cisse et al., 2017; Madry et al., 2017), we used Wide ResNet (Zagoruyko & Komodakis, 2016) models in our experiments as it is considered as the standard model on the dataset. We used depth 28 and widening factor 10 for CIFAR-10, and depth 16 and widening factor 4 for SVHN. We followed the training protocol in (Zagoruyko & Komodakis, 2016) that uses a SGD optimizer with momentum=0.9. For CIFAR-10, the model is trained for 200 epochs with initial learning rate 0.1, decayed by a factor of 0.2 at 60, 120 and 160 epochs. For SVHN dataset, the model is trained for 160 epochs, with initial learning rate 0.01, decayed by 0.1 at 80 and 120 epochs. For DQ, we used bit=1 and  $\beta = 2e-3$  as it offers the best robustness (see Section B).

We combine DQ with other defense methods to further boost the robustness. For Feature Squeezing (Xu et al., 2017), we used 5 bit for image color reduction, followed by a  $2 \times 2$  median filter. As adversarial FGSM training leads to gradient masking issue (Tramèr et al., 2017) (see A.2 for our experiment), we used the variant adversarial R+FGSM training. To avoid over-fitting into certain  $\epsilon$ , we randomly sample  $\epsilon'$  using  $\epsilon' \sim \mathcal{N}(0, \delta)$ ,  $\epsilon = \text{clip}^{[0, 2\delta]}(\text{abs}(\epsilon'))$ . Specifically we used  $\delta = 8$  to cover  $\epsilon$  from 0-16. During test time, the  $\epsilon$  is set to a fixed value (2/8/16). We also conducted adversarial PGD training. Following (Kurakin et al., 2016; Song et al., 2017), during training we sample random  $\epsilon$  as in R+FGSM setting, and generate adversarial samples using step size 1 and number of iterations  $\lfloor \min(\epsilon + 4, 1.25\epsilon) \rfloor$ .

**Result:** The results are presented in Table 2 and Table 3, where **(B)** indicates black-box attack with a separately trained VGG-16 model. The bold number indicates the best result in its column. We observe that for all normal training, feature squeezing and adversarial training settings, our DQ method can further improve the model’s robustness. Among all the defenses, adversarial training provides the best performance against various attacks, especially adversarial R+FGSM training. While white box PGD attack is generally the strongest attack in our experiments. Our DQ method also consistently improves the black-box robustness and there is no sign of gradient masking. Thus DQ proves to be an effective defense for various white-box and black-box attacks.

### 5.3 IMPROVE THE TRAINING OF QUANTIZED MODELS

As a by-product of our method, Defensive Quantization can even improve the accuracy of quantized models on clean images without attack, making it a beneficial drop-in substitute for normal quantization procedures. Due to conventional quantization method’s amplification effect, the distribution of activation can step over the truncation boundary (0-6 for ReLU6, 0-1 for ReLU1), which makes the optimization difficult. DQ explicitly add a regularization to shrink the dynamic range of activation, so that it is fitted within the truncation range. To demonstrate our hypothesis, we experimented with ResNet and CIFAR-10. We quantized the activation with 4-bit (because NVIDIA recently introduced INT4 in Turing architecture) using ReLU6 and ReLU1 respectively. Vanilla quantization and DQ training are conducted for comparison. As shown in Table 4, with vanilla quantization, ReLU1 model has around 1% worse accuracy than ReLU6 model, although they are mathematically equal if we



**Table 3.** CIFAR-10 Experiments tested with  $\epsilon = 2/8/16$ . (B) indicates black-box attack.

| Training Technique             | Clean       | Random          | FGSM            | R+FGSM          | PGD             | FGSM(B)         | PGD(B)          |
|--------------------------------|-------------|-----------------|-----------------|-----------------|-----------------|-----------------|-----------------|
| Normal                         | 94.8        | 95/91/77        | 59/39/29        | 72/37/17        | 56/01/00        | 88/63/36        | 90/64/42        |
| Normal + <b>DQ</b>             | <b>95.9</b> | <b>96/94/84</b> | 68/53/42        | 77/50/30        | 62/04/00        | 84/59/40        | 87/50/27        |
| Feature Squeezing              | 94.1        | 94/92/81        | 61/35/27        | 76/40/21        | 64/02/00        | 87/62/30        | 89/70/42        |
| Feature Squeezing + <b>DQ</b>  | 94.9        | 95/93/82        | 66/48/33        | 77/51/29        | 66/06/01        | 87/62/29        | 90/70/43        |
| Adversarial R+FGSM             | 91.6        | 92/91/91        | 81/52/38        | 87/69/48        | 84/43/11        | 92/89/85        | 92/91/89        |
| Adversarial R+FGSM + <b>DQ</b> | 94.0        | 94/93/93        | <b>85/63/51</b> | <b>90/74/58</b> | <b>87/50/22</b> | <b>93/91/87</b> | <b>94/92/92</b> |
| Adversarial PGD                | 86.6        | 86/86/86        | 74/46/31        | 79/63/46        | 76/44/20        | 84/83/81        | 84/83/83        |
| Adversarial PGD + <b>DQ</b>    | 87.5        | 87/87/87        | 79/53/36        | 83/69/52        | 81/50/22        | 87/86/82        | 87/86/86        |

**Table 4.** DQ method improves the training of normal quantized models by limiting the dynamic range of activation. With conventional quantization, ReLU1 suffers from inferior performance than ReLU6. While with DQ, the gap is fixed, ReLU1 and ReLU6 quantized models achieve similar accuracy.

|                        | ReLU1  | ReLU6  | Difference |
|------------------------|--------|--------|------------|
| Vanilla Quantization   | 93.49% | 94.41% | -0.92%     |
| Defensive Quantization | 95.14% | 95.09% | 0.05%      |

multiply the previous BN scaling by 1/6 and next convolution weight by 6. It demonstrates that improper truncation function and range will lead to training difficulty. While with DQ training, both model has improved accuracy compared to vanilla quantization, and the gap between ReLU1 model and ReLU6 model is filled, making quantization easier regardless of truncation range.

## 6 CONCLUSION

In this work, we aim to raise people’s awareness about the security of the quantized neural networks, which is widely deployed in GPU/TPU/FPGAs, and pave a possible direction to bridge two important areas in deep learning: efficiency and robustness. We connect these two domains by designing a novel Defensive Quantization (DQ) module to defend adversarial attacks while maintain the efficiency. Experimental results on two datasets validate that the new quantization method can make the deep learning models be safely deployed on mobile devices.

### ACKNOWLEDGMENTS

We acknowledge Google and Amazon providing us the cloud computing resources.

### REFERENCES

- Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: a system for large-scale machine learning. In *OSDI*, volume 16, pp. 265–283, 2016.
- Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in ai safety. *arXiv preprint arXiv:1606.06565*, 2016.
- Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *arXiv preprint arXiv:1802.00420*, 2018.
- Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Advances in Neural Information Processing Systems*, pp. 6240–6249, 2017.

- Yoshua Bengio, Nicholas Léonard, and Aaron Courville. Estimating or propagating gradients through stochastic neurons for conditional computation. *arXiv preprint arXiv:1308.3432*, 2013.
- Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. *arXiv preprint arXiv:1704.08847*, 2017.
- Matthieu Courbariaux and Yoshua Bengio. Binarynet: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*, 2016.
- Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy Bernstein, Jean Kossaifi, Aran Khanna, and Anima Anandkumar. Stochastic activation pruning for robust adversarial defense. *arXiv preprint arXiv:1803.01442*, 2018.
- Angus Galloway, Graham W Taylor, and Medhat Moussa. Attacking binarized neural networks. *arXiv preprint arXiv:1711.00449*, 2017.
- Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples (2014). *arXiv preprint arXiv:1412.6572*.
- Song Han, Huizi Mao, and William J Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv preprint arXiv:1510.00149*, 2015.
- Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.
- 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. *arXiv preprint arXiv:1712.05877*, 2017.
- Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.
- Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.
- Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Jun Zhu, and Xiaolin Hu. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1778–1787, 2018.
- Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.
- Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS workshop on deep learning and unsupervised feature learning*, volume 2011, pp. 5, 2011.
- NVIDIA. Nvidia turing architecture whitepaper. URL <https://www.nvidia.com/content/dam/en-zz/Solutions/design-visualization/technologies/turing-architecture/NVIDIA-Turing-Architecture-Whitepaper.pdf>.
- Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against deep learning systems using adversarial examples. *arXiv preprint*, 2016a.
- Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597. IEEE, 2016b.
- Haifeng Qian and Mark N Wegman. L2-nonexpansive neural networks. *arXiv preprint arXiv:1802.07896*, 2018.
- Adnan Siraj Rakin, Jinfeng Yi, Boqing Gong, and Deliang Fan. Defend deep neural networks against adversarial examples via fixed anddynamic quantized activation functions. *arXiv preprint arXiv:1807.06714*, 2018.

- Mohammad Rastegari, Vicente Ordonez, Joseph Redmon, and Ali Farhadi. Xnor-net: Imagenet classification using binary convolutional neural networks. In *European Conference on Computer Vision*, pp. 525–542. Springer, 2016.
- Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *arXiv preprint arXiv:1801.04381*, 2018.
- Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- Yang Song, Taesup Kim, Sebastian Nowozin, Stefano Ermon, and Nate Kushman. Pixeldefend: Leveraging generative models to understand and defend against adversarial examples. *arXiv preprint arXiv:1710.10766*, 2017.
- Dong Su, Huan Zhang, Hongge Chen, Jinfeng Yi, Pin-Yu Chen, and Yupeng Gao. Is robustness the cost of accuracy?—a comprehensive study on the robustness of 18 deep image classification models. *arXiv preprint arXiv:1808.01688*, 2018.
- Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.
- Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.
- David Warde-Farley and Ian Goodfellow. 11 adversarial perturbations of deep neural networks. *Perturbations, Optimization, and Statistics*, pp. 311, 2016.
- Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.
- Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.
- 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.
- Chenzhuo Zhu, Song Han, Huizi Mao, and William J Dally. Trained ternary quantization. *arXiv preprint arXiv:1612.01064*, 2016.

## A GRADIENT MASKING OF OTHER METHODS

### A.1 TANH-BASED ACTIVATION QUANTIZATION LEADS TO FAKE SECURITY

Rakin et al. (2018) tried to use activation quantization as a defense method against adversarial samples. According to their experiments, simply Tanh based activation quantization can greatly improve the model’s robustness under PGD adversarial training setting. We reproduced their experiments by training a ResNet-18 model with PGD adversarial training on CIFAR-10 following Madry et al. (2017). To make a comparison, we also trained a full precision model and a ReLU6 quantized model following same setting. All the quantized models use bit=2. We tested the trained model using PGD attack by Madry et al. (2017) under both white-box and black-box setting. We use a VGG-16 model separately trained with PGD adversarial training as the black-box attacker. The results are provided in Table 5.

**Table 5.** Tanh-based quantization improves white-box robustness, while suffers from inferior black-box robustness compared to full-precision model. The white-box accuracy is even higher than black-box, suggesting gradient masking. Setup: white-box and black-box robustness of PGD adversarially trained ResNet-18 on CIFAR-10. Quantization uses 2 bits.

|                 | Clean Acc. | White-Box Acc. | Black-Box Acc. |
|-----------------|------------|----------------|----------------|
| Full-Precision  | 83.47%     | 50.51%         | 65.58%         |
| Tanh Quantized  | 81.60%     | 71.03%         | 61.57%         |
| ReLU6 Quantized | 78.97%     | 48.90%         | 59.44%         |

Our white-box result is consistent with (Rakin et al., 2018), where Tanh based quantization with PGD training gives much higher white-box accuracy compared to the full precision model. However, we can see that the black-box robustness decreases. Worse still, the black-box attack successful rate is even lower than white-box, which is abnormal since black-box attack is generally weaker than white-box. This phenomenon indicates severe gradient masking problem (Papernot et al., 2016a; Athalye et al., 2018), which gives a false sense of security. As a comparison, we also trained a ReLU6 quantized model. With ReLU6 quantization, there is no sign of gradient masking, nor improvement in robustness, indicating that gradient masking problem majorly comes from Tanh activation function. Actually, ReLU6 quantized model has slightly worse robustness.

Therefore we conclude that simple quantization cannot help to improve robustness. Instead, it leads to inferior robustness.

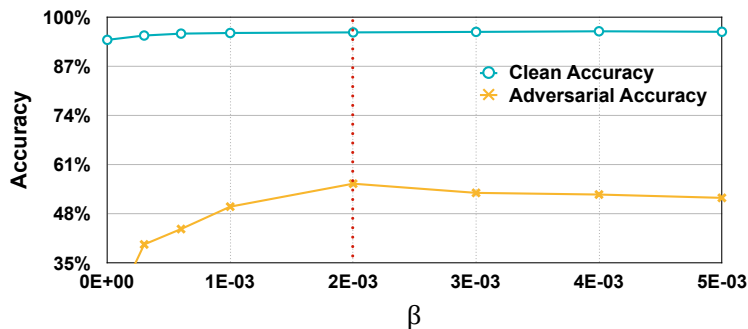
### A.2 ADVERSARIAL FGSM TRAINING CAUSES GRADIENT MASKING

Here we demonstrate that FGSM adversarial training leads to significant gradient masking problem, while R+FGSM fixes such issues. We trained a Wide ResNet using adversarial FGSM and adversarial R+FGSM respectively. Then the model is tested using FGSM with  $\epsilon = 8$  under white-box and black-box setting (VGG-16 as substitute). The results are shown in Table

**Table 6.** White-box and black-box robustness of FGSM/R+FGSM adversarially trained Wide ResNet on CIFAR-10.

|                    | Clean Acc. | White-Box Acc. | Black-Box Acc. |
|--------------------|------------|----------------|----------------|
| Adversarial FGSM   | 94.55%     | 94.18%         | 51.65%         |
| Adversarial R+FGSM | 91.61%     | 51.65%         | 87.39%         |

We can clearly see the same gradient masking effect. For adversarial FGSM training, it gives much higher white box robustness than R+FGSM adversarial training, while the black-box robustness is much worse. To avoid gradient masking, we thus use R+FGSM adversarial training instead.









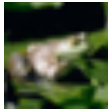
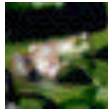
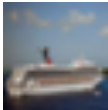

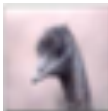





B HYPER-PARAMETERS:  $\beta$  STUDY

**Figure 6.** Clean and adversarial accuracy of 4-bit quantized Wide ResNet w.r.t. different  $\beta$ .

As observed in Section 5.1, the adversarial robustness of the model generally increases as  $\beta$  gets larger. Here we aim to find what is the optimal  $\beta$  for our experiment. We took 4-bit quantized Wide ResNet for example. As shown in Figure 6, the adversarial accuracy first gets larger as  $\beta$  increases, and slowly goes down afterwards, reaching peak performance at  $\beta = 0.002$ . The clean accuracy is more stable. At the first stage, as regularization gets stronger, the effect of noise amplify is suppressed more. While for the second stage, the training suffers from a too strong regularization. Therefore, we used a  $\beta = 0.002$  for our experiments unless specified.

## C VISUALIZE SAMPLES AND PREDICTIONS

We visualize some of the clean and adversarial samples from CIFAR-10 test set, and the corresponding prediction for full precision model (**FP**), vanilla 4-bit quantized model (**VQ**) and our 4-bit Defensive Quantization model (**DQ**), as 4-bit quantization is now supported by Tensor Cores with NVIDIA's Turing Architecture (NVIDIA). Predicted class and probability is provided. Compared to FP and DQ, VQ has worse robustness by misclassifying more adversarial samples (sample 1, 2, 4, 6). Our DQ model enjoys better robustness than FP model in two aspects: 1. our DQ model succeeds to defend some of the attacks when FP failed (sample 5, 7); 2. our DQ model has a better confidence for true label compared to FP model when both succeeds to defend (sample 1, 4, 6). Even when all models fail to defend, our DQ model has the lowest confidence for the misclassified class (sample 3, 8).

|              | Clean   | Adversarial  |
|--------------|---|--|
| 1. GT: bird  |  <p>FP: bird (1.00) ✓<br/>VQ: bird (1.00) ✓<br/>DQ: bird (1.00) ✓</p>      |  <p>FP: bird (0.37) ✓<br/>VQ: cat (0.98) ✗<br/>DQ: bird (0.94) ✓</p>       |
| 2. GT: truck |  <p>FP: truck (1.00) ✓<br/>VQ: truck (1.00) ✓<br/>DQ: truck (1.00) ✓</p>   |  <p>FP: truck (1.00) ✓<br/>VQ: deer (0.50) ✗<br/>DQ: truck (1.00) ✓</p>    |
| 3. GT: plane |  <p>FP: plane (1.00) ✓<br/>VQ: plane (1.00) ✓<br/>DQ: plane (1.00) ✓</p>  |  <p>FP: deer (1.00) ✗<br/>VQ: cat (0.99) ✗<br/>DQ: deer (0.51) ✗</p>      |
| 4. GT: horse |  <p>FP: horse (1.00) ✓<br/>VQ: horse (1.00) ✓<br/>DQ: horse (1.00) ✓</p> |  <p>FP: horse (0.97) ✓<br/>VQ: deer (0.97) ✗<br/>DQ: horse (0.99) ✓</p>  |
| 5. GT: frog  |  <p>FP: frog (1.00) ✓<br/>VQ: frog (1.00) ✓<br/>DQ: frog (1.00) ✓</p>    |  <p>FP: car (0.95) ✗<br/>VQ: deer (0.45) ✗<br/>DQ: frog (0.73) ✓</p>     |
| 6. GT: ship  |  <p>FP: ship (1.00) ✓<br/>VQ: ship (1.00) ✓<br/>DQ: ship (1.00) ✓</p>    |  <p>FP: ship (0.96) ✓<br/>VQ: frog (0.57) ✗<br/>DQ: ship (1.00) ✓</p>    |
| 7. GT: bird  |  <p>FP: bird (1.00) ✓<br/>VQ: bird (1.00) ✓<br/>DQ: bird (1.00) ✓</p>    |  <p>FP: horse (0.32) ✗<br/>VQ: cat (0.87) ✗<br/>DQ: bird (0.99) ✓</p>    |
| 8. GT: ship  |  <p>FP: ship (0.99) ✓<br/>VQ: ship (1.00) ✓<br/>DQ: ship (1.00) ✓</p>    |  <p>FP: car (0.81) ✗<br/>VQ: frog (0.82) ✗<br/>DQ: frog (0.50) ✗</p>     |
| 9. GT: truck |  <p>FP: truck (1.00) ✓<br/>VQ: truck (1.00) ✓<br/>DQ: truck (1.00) ✓</p> |  <p>FP: truck (1.00) ✓<br/>VQ: truck (0.53) ✓<br/>DQ: truck (1.00) ✓</p> |

**Figure 7.** Visualization of adversarial samples and the corresponding predictions (label and probability). **FP**: full-precision model; **VQ**: vanilla 4-bit quantized model; **DQ**: our 4-bit Defensive Quantization model