Spear: Evaluate the Adversarial Robustness of Compressed Neural Models

Chong Yu[1] , **Tao Chen**[2,*] , **Zhongxue Gan**[1,*] and **Jiayuan Fan**[1]
[1]Academy for Engineering and Technology, Fudan University
[2]School for Information Science and Technology, Fudan University
21110860050@m.fudan.edu.cn, {eetchen, ganzhongxue, jyfan}@fudan.edu.cn

Abstract

As Artificial Intelligence evolves, the neural models vulnerable to adversarial attacks may produce fatal results in critical applications. This paper mainly discusses the robustness of the compressed neural models facing adversarial attacks. A few studies discuss the interaction between model compression and adversarial attack. However, they focus on the robustness against the traditional attacks designed for the dense models, not the attacks intended explicitly for the compressed models, using sparsity and quantization techniques. Compressed models often have fewer parameters and smaller sizes that are more friendly to resource-limited devices than dense models, so they are widely deployed in various edge and mobile devices. However, introducing the sparsity and quantization into neural models further imposes higher attack risks. A specific adversarial attack method (Spear) is proposed to generate the particular adversarial attack samples for evaluating the robustness of the compressed models. The Spear attack finds minimal perturbations to create the attack samples to maximize the different behaviors between the compressed and dense reference models. We demonstrate the proposed Spear attack technique can generally be applied to various networks and tasks through quantitative and ablation experiments.

1 Introduction

As machine learning evolves, it becomes a unilateral and narrow viewpoint to evaluate the neural networks without perturbations. Prior works [Szegedy *et al.*, 2014] [Goodfellow *et al.*, 2014] have proved that the neural model will behave differently by adding negligible perturbations to specific input cases. Sometimes, the different behaviors are caused by the numerical difference, finally leading to the different tolerance to the corner cases in benign input samples. However, the situation worsens if some imperceptible perturbations are intentionally added to the benign samples. For example, if the on-purpose perturbations lead the medical auxiliary diagnosis model to the wrong focus of infection, the neural model will become a murder instrument. If the on-purpose perturbations lead the Conversational AI to generate discriminatory, intimidatory, and racist statements, it will lead to severe social problems [Sap *et al.*, 2020]. Generating such perturbations to mislead the neural models' behavior is referred to as an adversarial attack. Most prior adversarial attack studies have been successfully applied to dense neural networks, mainly for image classification tasks. Some other studies discuss the interaction between model compression and adversarial robustness. (More discussions can refer to 2.2.) However, *they mainly focus on applying the adversarial workflow to defend against the attack and improve the robustness.* ***There has been very little work to study the specific attack algorithms on the compressed networks performing various tasks.***

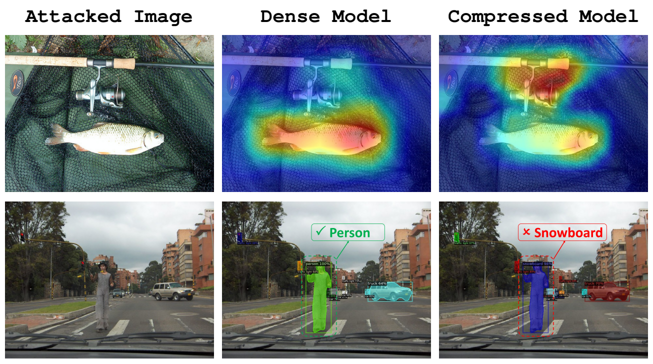


Figure 1: (***Line 1***) Class activation mapping visualization for classification. (The dense and compressed *ResNeXt-101* models have almost the same *Top-1&5* metric). (***Line 2***) Visualization for object segmentation. (The dense and compressed *Mask R-CNN with ResNet-50 backbone* models have almost the same *IoU* metric).

When we try to use a compressed neural model as the substitute for an original dense model, we usually ensure this compressed neural model has a similar model accuracy on benign samples. To further check whether a compressed neural model is a qualified substitute with perturbations, the prior practice generates the adversarial attack samples with effective perturbations according to the original model. Then, it evaluates the behavior of the compressed neural model with these adversarial attack samples. Suppose the compressed

*Tao Chen and Zhongxue Gan are corresponding authors.

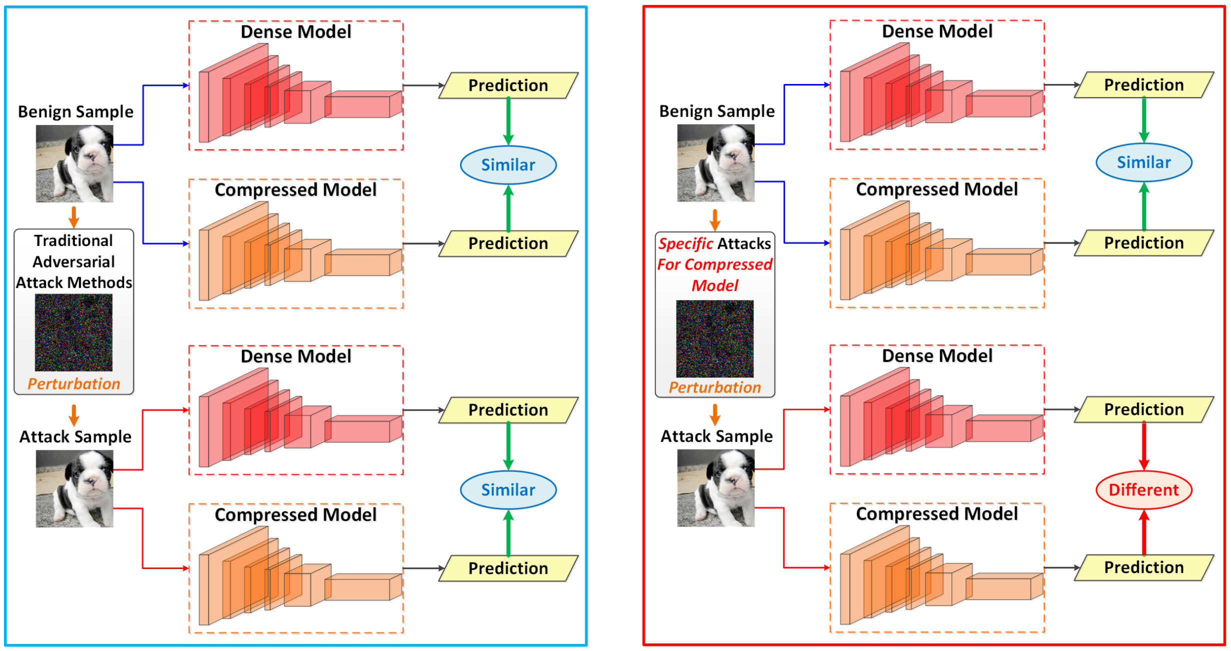


Figure 2: Comparison between the traditional adversarial robustness evaluation strategy (***Left***) and the proposed Spear evaluation strategy (***Right***). Traditional strategy evaluates the behavior of the compressed model with the adversarial attack samples generated according to the dense model. However, it has the risk if these are specific adversarial attack methods can generate the adversarial attack samples which only influence the compressed model while keeping the behavior of the dense model as usual.

neural model's behavior is similar to the original model; we regard the compressed neural model as having similar adversarial robustness on par with its dense counterpart. This traditional evaluation workflow is shown in Figure 2 **Left**.

However, such prior practice ignores whether some specific adversarial attack algorithms can generate the adversarial attack samples that only influence the compressed model while keeping the behavior of the original model as usual, as shown in Figure 2 **Right**. Here are two examples. In **Line 1** of Figure 1, the dense model classifies [Zhou *et al.*, 2016] the attacked input image into the ***tench*** category. In contrast, the compressed model classifies the same image into the ***reel*** category. The adversarial attack algorithm does not influence the classification behavior of the dense model. However, it confuses the classification behavior of the compressed model by changing its decision from one to another evident category in the image. In **Line 2** of Figure 1, the dense *Mask R-CNN* model [He *et al.*, 2017] successfully segments the traffic lights, the car in the background, and the ***person*** walking across the pedestrian crossing from the attacked input image. The compressed model segments all these object regions in the ground truth image; however, it regards the person as the ***snowboard*** under such a specific adversarial attack. It may lead to a fatal disaster if the autopilot system uses the wrong segmentation result, which controls the autonomous vehicle.

The compressed models have reduced weight parameters with sparse patterns or quantized data types. For intentional attackers, ***these characteristics can be utilized to discover the potential flaw of the compressed models and eventually be targeted by the specific adversarial attack algorithm.*** So, in this paper, we mainly focus on exploring the specific adversarial attack for the compressed neural models and further evaluate the adversarial robustness. On the other hand, the adversarial attack algorithm study for the compressed model is ***much more complicated*** than designing the attack for the dense model. Various model compression methods like pruning, distillation, and quantization generate the compressed models with different patterns, like different sparse granularities and data formats. ***Our work designs a unified adversarial robustness evaluation framework that is effective, efficient, and general to most model compression patterns, which is not fully addressed in the previous works.***

2 Related Work

2.1 Adversarial Attack

Adversarial attack [Kianpour and Wen, 2019], a technique that attempts to fool machine learning models with deceptive data, is a growing threat in the AI industry and research community [Cheng *et al.*, 2021]. The ultimate goal is to cause the specific malfunction in a machine learning model. An adversarial attack might entail presenting a model with misleading data during its training process or directly introducing maliciously designed data to deceive an already trained model.

The adversarial attack methods can be divided into the white-box attack and the black-box attack. In white-box attack settings, the adversaries have access to the parameters and structure of the target model, and the training dataset. Instead, the adversaries have no access to or little knowledge of the target model in the black-box attack scenarios. Still, they can train a local substitute model [Papernot *et al.*, 2017] by querying the target model or applying the model inversion method. [Szegedy *et al.*, 2013] found the deep neural models learn input-output mappings that are somewhat discontinuous to a significant extent, causing the model to misclassify an image by applying a certain hardly perceptible perturbation, namely adversarial samples. They proposed the box-constrained Broyden-Fletcher-Goldfarb-Shanno (*BFGS*) algorithm to search for these adversarial samples. Goodfellow et al. [Goodfellow *et al.*, 2014] proposed the Fast Gradient Sign Method (*FGSM*), which generates the adversarial examples with one gradient sign step. They involved these adversarial examples during the back-propagation, which is the rudiment of adversarial training. [Papernot *et al.*, 2016] crafted adversarial samples based on computing the Jacobian matrix as the forward derivatives. [Moosavi-Dezfooli *et al.*, 2016] proposed a *DeepFool* algorithm to compute adversarial examples based on an iterative linearization of the classifier to generate minimal perturbations that are sufficient to change classification labels. In this paper, we design the white-box adversarial attack with the access to original model. The comparison with other white-box attacks can refer to the experiment results shown in section 4.2 and 4.3.

2.2 Interaction between Compression and Attack

There are a few studies to discuss the interaction between model compression and adversarial attack. [Ye *et al.*, 2019] proposed a framework of concurrent adversarial training and weight pruning that tries to preserve the adversarial robustness best when enabling model pruning. [Song *et al.*, 2020] observed that adversarially-trained neural networks are vulnerable to quantization loss. So they proposed a boundary-based retraining method to mitigate adversarial and quantization losses together and adopted a nonlinear mapping method to defend against white-box gradient-based adversarial attacks. [Shumailov *et al.*, 2019] reported an empirical study to investigate whether the adversarial samples are transferable between uncompressed and compressed models. Pruning and quantization may lead to different transferability. Adversarially trained model compression (ATMC) [Gui *et al.*, 2019] is a unified optimization framework to trade-off model size, accuracy, and robustness. These studies focused more on the adversarial robustness when facing the traditional adversarial attacks designed for dense models. ***They do not discuss whether some adversarial attacks are designed explicitly for the compressed models and the corresponding robustness against these attacks.*** Moreover, most of the experiments are evaluated at relatively small datasets, e.g., MNIST [LeCun *et al.*, 1998], CIFAR-10 [Krizhevsky *et al.*, 2009], and CIFAR-100 [Krizhevsky *et al.*, 2009]. They did not provide the results on large-scale datasets, like ImageNet [Deng *et al.*, 2009], COCO [Lin *et al.*, 2014], etc.

3 Spear: Attack for Compressed Model

Sparsity and quantization are two standard and applicable model compression methods. We will design an adversarial attack against these two compression processes. We do not include the discussion of network architecture search (NAS) here because NAS will lead the model structures to various shapes, so the attacks against the NAS-compressed models may be case by case, with no general rules to be summarized.

3.1 General Attack for Neural Model

A general supervised neural model can be expressed as:

$$\boldsymbol{y} = \boldsymbol{F}(\theta, \boldsymbol{x}) \tag{1}$$

where $\boldsymbol{x}$ is the input, θ is the weight parameters of the model, and $\boldsymbol{y}$ is the corresponding output of the neural function $\boldsymbol{F}$. Given the supervised label info $\boldsymbol{y}_l$, the model training and optimization process can be expressed as the following minimization problem:

$$\min_{\theta} \boldsymbol{L}(\boldsymbol{F}(\theta, \boldsymbol{x}), \boldsymbol{y}_l) \tag{2}$$

where $\boldsymbol{L}(\cdot)$ is the loss function, for instance, the cross-entropy loss for a classification network, the Dice loss for a segmentation task, the negative log-likelihood for a language processing network.

For the adversarial attack settings, the perturbation δ is added to the input $\boldsymbol{x}$ to constitute the attack samples $\boldsymbol{x}_{adv}$, which subjects to the L_* perturbation constraint:

$$\|\boldsymbol{x}_{adv} - \boldsymbol{x}\|_* = \|\delta\|_* \leq \epsilon, \quad * \in \{0, 1, 2, \infty\} \tag{3}$$

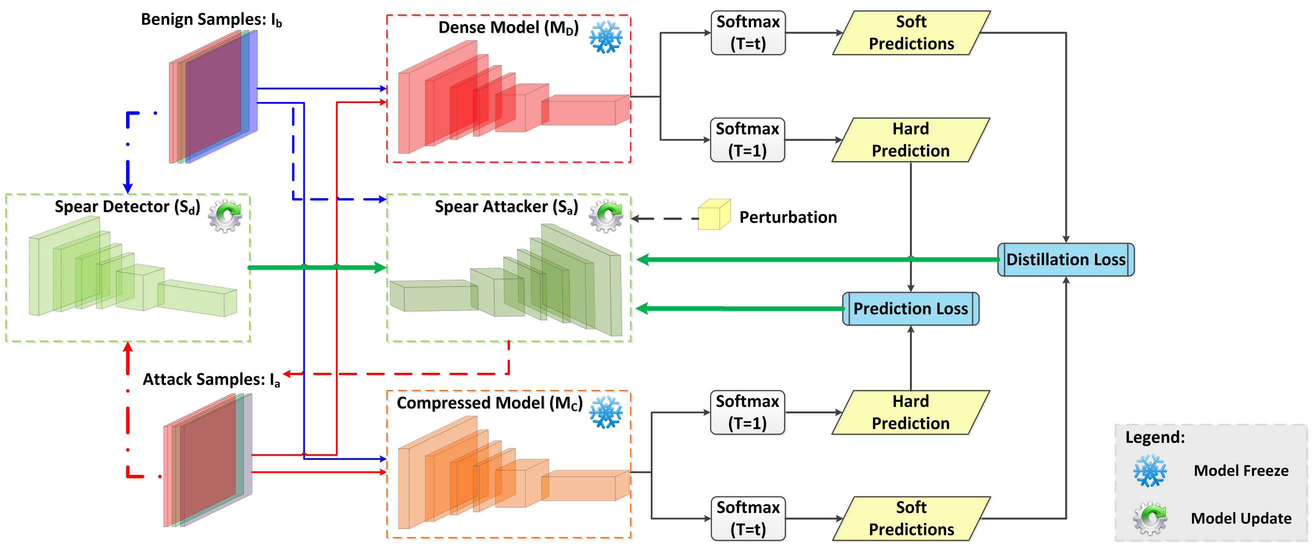
where ϵ refers to the attack strength. *The objective for the adversarial attack is to maximally deteriorate and change the 'normal' behavior of the model within the perturbation constraint.* If the whole dataset is defined as $\boldsymbol{D}$, with N categories, and each category is labeled with $\boldsymbol{y}_{l:k}$ and denoted as $\boldsymbol{D}(\boldsymbol{y}_{l:k})$, then the adversarial target can be denoted with the following joint optimization problem.

$$\begin{cases} \min \sum_{k=1}^{N} \sum_{\boldsymbol{x}_k \in \boldsymbol{D}(\boldsymbol{y}_{l:k})} \boldsymbol{L}(\boldsymbol{F}(\theta, \boldsymbol{x}_k), \boldsymbol{y}_{l:k}) \\ \max \sum_{k=1}^{N} \sum_{\boldsymbol{x}_k \in \boldsymbol{D}(\boldsymbol{y}_{l:k})} \boldsymbol{L}(\boldsymbol{F}(\theta, \boldsymbol{x}_k + \delta_k), \boldsymbol{y}_{l:k}) \\ \min \sum_{k=1}^{N} \|\delta_k\|_*, \quad \|\delta_k\|_* \leq \epsilon, \quad * \in \{0, 1, 2, \infty\} \end{cases} \tag{4}$$

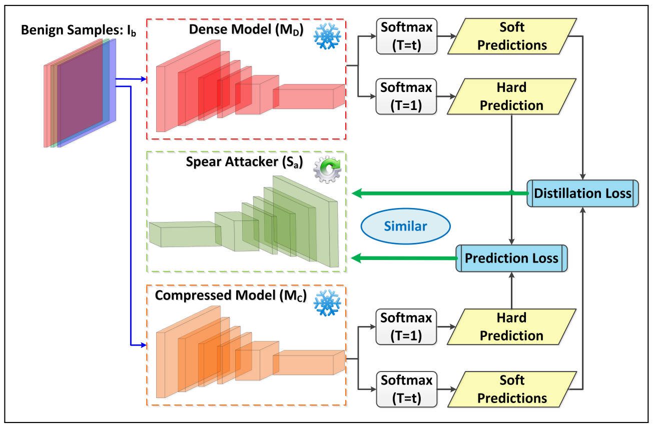
3.2 Specific Attack for Sparsity

For any sparse model, we can always regard it as performing an element-wise multiplication ($\odot$) between a mask function $\boldsymbol{f}_M$ and the original weight parameters θ_d from the corresponding dense model. The different sparse algorithms just change the concrete form and sparse granularity of the mask function. ***In our assumption, we want to know whether there is a specific adversarial attack type, which deteriorates the sparse model's performance while keeping the performance of its dense form at the meanwhile.*** So the joint optimization problem changes to the following form:

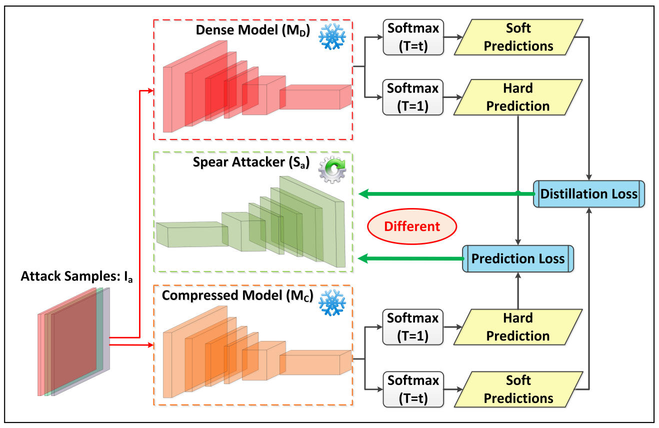
$$\begin{cases} \min \sum_{k=1}^{N} \sum_{\boldsymbol{x}_k \in \boldsymbol{D}(\boldsymbol{y}_{l:k})} \boldsymbol{L}(\boldsymbol{F}(\theta_d, \boldsymbol{x}_k), \boldsymbol{y}_{l:k}) \\ \max \sum_{k=1}^{N} \sum_{\boldsymbol{x}_k \in \boldsymbol{D}(\boldsymbol{y}_{l:k})} \boldsymbol{L}(\boldsymbol{F}(\theta_d \odot \boldsymbol{f}_M, \boldsymbol{x}_k + \delta_k), \boldsymbol{y}_{l:k}) \\ \min \sum_{k=1}^{N} \|\delta_k\|_*, \quad \|\delta_k\|_* \leq \epsilon, \quad * \in \{0, 1, 2, \infty\} \end{cases} \tag{5}$$



(a) Whole workflow chart of entire ***Spear*** attack algorithm. The weight parameters of dense model and compressed model are fixed during the training process of the ***Spear*** attacker and detector models. When we finish the ***Spear*** training process, we can use the ***Spear*** attacker to generate various attack samples which only effective to the compressed model, while with no influence on the behavior to the dense model.



(b) Benign samples enter ***Spear*** algorithm.



(c) Attack samples enter ***Spear*** algorithm.

Figure 3: Detailed workflow charts of ***Spear*** attack algorithm. (a) shows the whole workflow chart, while two different situations when benign or attack samples as input are shown in (b) and (c), respectively. (b) When the benign samples enter ***Spear*** algorithm, the predictions from the dense and compressed models are expected to be almost the ***same***. (c) When the attack samples enter ***Spear*** algorithm, the predictions from the dense and compressed models are expected to be totally ***different***.

3.3 Specific Attack for Quantization

Quantization uses alternative formats than the standard 32-bit single-precision floating-point (**FP32**) data type for parameters in a dense model. Because the standard **FP32** data format can emulate almost all lower-precision formats, including floating-point, fixed-point, and integer. So for any quantized model, we can always regard it as performing an element-wise multiplication ($\odot$) between a quantization function $\boldsymbol{f}_Q$ and the original weight parameters θ_d from the corresponding dense model. The different quantization methods just change the concrete form and quantized data format of the quantization function. ***In our assumption, we want to know whether there is a specific adversarial attack type to deteriorate the quantized model's performance while keeping the performance of its dense form at the meanwhile.*** So the joint optimization problem changes to the following form:

$$\begin{cases} \min \sum_{k=1}^{N} \sum_{\boldsymbol{x}_k \in \boldsymbol{D}(\boldsymbol{y}_{l:k})} \boldsymbol{L}(\boldsymbol{F}(\theta_d, \boldsymbol{x}_k), \boldsymbol{y}_{l:k}) \\ \max \sum_{k=1}^{N} \sum_{\boldsymbol{x}_k \in \boldsymbol{D}(\boldsymbol{y}_{l:k})} \boldsymbol{L}(\boldsymbol{F}(\theta_d \odot \boldsymbol{f}_Q, \boldsymbol{x}_k + \delta_k), \boldsymbol{y}_{l:k}) \\ \min \sum_{k=1}^{N} \|\delta_k\|_*, \quad \|\delta_k\|_* \le \epsilon, \quad * \in \{0, 1, 2, \infty\} \end{cases} \tag{6}$$

3.4 Spear: Specific Attack for Compressed Model

Based on the analysis of the adversarial attack for the sparse and quantized model in expressions (5) and (6), we can use a **unified** format of (5) and (6) to express the adversarial optimization target of these two orthogonal model compression methods. So we also propose a **unified** adversarial attack algorithm named ***Spear*** that can generate specific attack examples to maximally and efficiently deteriorate the compressed models. The workflow of ***Spear*** attack is shown in Figure 3a.

In the proposed algorithm, the dense baseline model is referred as $\boldsymbol{M}_D$, and the corresponding compressed model is $\boldsymbol{M}_C$ ($\boldsymbol{M}_C = \boldsymbol{M}_D \odot \boldsymbol{f}_M$ for the sparse model, and $\boldsymbol{M}_C = \boldsymbol{M}_D \odot \boldsymbol{f}_Q$ for the quantized model). The ***Spear*** consists of two sub-neural-models: ***Spear*** attacker ($\boldsymbol{S}_a$) and ***Spear*** detector ($\boldsymbol{S}_d$). ***Spear*** attacker model takes the benign samples as input ($\boldsymbol{I}_b$), and adds the minimal perturbation δ within attack strength constraint (ϵ) to generate the attack samples ($\boldsymbol{I}_a$).

$$\boldsymbol{I}_a = \boldsymbol{S}_a(\boldsymbol{I}_b, \delta), \quad \|\delta\|_* \leq \epsilon, \quad * \in \{0, 1, 2, \infty\} \tag{7}$$

When the dense and compressed models take the samples as input, they will provide the predicted results. Inspired by the **K**nowledge **D**istillation (**KD**) [Hinton *et al.*, 2015], we consider the hard prediction to learn from the one-hot representation of the ground-truth label; meanwhile, we also consider the soft predictions to learn from the soft labels to represent all probabilities over the whole classes in the dataset. We use the cross-entropy to calculate the prediction loss ($\boldsymbol{L}_p$) between the hard prediction from dense (for benign and attack samples: $\boldsymbol{P}_h(\boldsymbol{M}_D,\boldsymbol{I}_b), \boldsymbol{P}_h(\boldsymbol{M}_D,\boldsymbol{I}_a)$) and compressed model (for benign and attack samples: $\boldsymbol{P}_h(\boldsymbol{M}_C,\boldsymbol{I}_b)$, $\boldsymbol{P}_h(\boldsymbol{M}_C,\boldsymbol{I}_a)$). **K**ullback-**L**eibler (**KL**) divergence is used to compute the distillation loss ($\boldsymbol{L}_d$) between soft predictions from the dense (for benign and attack samples: $\boldsymbol{P}_s(\boldsymbol{M}_D,\boldsymbol{I}_b)$, $\boldsymbol{P}_s(\boldsymbol{M}_D,\boldsymbol{I}_a)$) and compressed model (for benign and attack samples: $\boldsymbol{P}_s(\boldsymbol{M}_C,\boldsymbol{I}_b), \boldsymbol{P}_s(\boldsymbol{M}_C,\boldsymbol{I}_a)$).

According to the joint optimization formula, when taking the benign samples as input, we hope the predictions from dense and compressed models are almost the same. The detailed workflow of this situation is shown in Figure 3b.

$$\begin{cases} \min & \boldsymbol{L}_p\left(\boldsymbol{P}_h(\boldsymbol{M}_D,\boldsymbol{I}_b), \boldsymbol{P}_h(\boldsymbol{M}_C,\boldsymbol{I}_b); T = 1\right) \\ \min & \boldsymbol{L}_d\left(\boldsymbol{P}_s(\boldsymbol{M}_D,\boldsymbol{I}_b), \boldsymbol{P}_s(\boldsymbol{M}_C,\boldsymbol{I}_b); T = t\right) \end{cases} \tag{8}$$

When the attack samples are taken as input, we expect the dense and compressed models' predictions to differ. The detailed workflow of this situation is shown in Figure 3c.

$$\begin{cases} \max & \boldsymbol{L}_p\left(\boldsymbol{P}_h(\boldsymbol{M}_D,\boldsymbol{I}_a), \boldsymbol{P}_h(\boldsymbol{M}_C,\boldsymbol{I}_a); T = 1\right) \\ \max & \boldsymbol{L}_d\left(\boldsymbol{P}_s(\boldsymbol{M}_D,\boldsymbol{I}_a), \boldsymbol{P}_s(\boldsymbol{M}_C,\boldsymbol{I}_a); T = t\right) \end{cases} \tag{9}$$

Spear detector takes the benign samples, and the generated attack samples as input. *In the settings of adversarial attacks, the perturbation is intended to be optimized to a minimum.* So ***Spear*** detector takes the responsibility to judge the difference between the benign and attack samples in a perception manner, and provide the adversarial loss ($\boldsymbol{L}_a$) to optimize the ***Spear*** attacker and ***Spear*** detector in the next iteration. ***During the optimization process, if the adversarial loss converges to a minimal value, that means the perturbation added to the benign samples is also minimal.***

$$\begin{aligned} L_{Overall} =& \alpha_1 * \boldsymbol{L}_p(\boldsymbol{I}_b) + \alpha_2 * \boldsymbol{L}_p(\boldsymbol{I}_a)^{-1} + \beta_1 * \boldsymbol{L}_d(\boldsymbol{I}_b) \\ & + \beta_2 * \boldsymbol{L}_d(\boldsymbol{I}_a)^{-1} + \gamma * \boldsymbol{L}_a\left(\boldsymbol{S}_a, \boldsymbol{S}_d\right) \end{aligned} \tag{10}$$

Where $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma$ are loss adjustment factors. We take the reciprocals of the prediction and distillation loss items for the attack samples, change them to the minimization form, and integrate them into the overall loss formula.

4 Experiments and Results

4.1 Experiment Settings

For the experiments in this section, we choose PyTorch [Paszke *et al.*, 2017] with version 1.8.0 as the framework to implement all algorithms. All of the state-of-the-art methods and ***Spear*** attack training and fine-tuning experimental results are obtained with V100 [NVIDIA, 2017] and A100 [NVIDIA, 2020a] GPU clusters. The acceleration performance results are obtained with A100 GPU clusters to fully utilize its Tensor Core [NVIDIA, 2020b] support for mixed-precision calculation, fine-grained sparsity[1], and quantization[2] [3].

4.2 Effectiveness Experiments for Classification

To evaluate the effectiveness of the ***Spear*** attack on the image classification task, ResNet-50 [He *et al.*, 2016][4], and Swin Transformer [Liu *et al.*, 2022][5] are chosen as the experiment target models. The loss adjustment parameters among the prediction loss (α_1, α_2), the distillation loss (β_1, β_2) and the adversarial loss (γ) apply 1, 1.5, 2, 4, 1, respectively.

In the Figure 4, ***_Sparse** represents the sparse model trained with knowledge distillation and the weight pattern is aligned with Sparse Tensor Cores[1] [Yu, 2021]. **INT8_*** represents both the weight and activation tensors of the model are quantized to INT8 with Quantization Aware Training (***QAT***) method [Krishnamoorthi, 2018]. **INT4_*** represents the weight tensors are quantized to INT4 and the activation tensors are quantized to INT8 with ***QAT*** method. **FP8_*** represents both the weight and activation tensors of the model are quantized to FP8 in the train-from-scratch process [Micikevicius *et al.*, 2022].

For the reference adversarial attack methods, we choose the ***FGSM*** [Goodfellow *et al.*, 2014], ***BIM*** [Kurakin *et al.*, 2018], ***PGD*** [Madry *et al.*, 2017], ***Jitter*** [Schwinn *et al.*, 2023], and Carlini-Wagner (***CW***) [Carlini and Wagner, 2017] as the reference algorithms. In this experiment, ***FGSM***, ***BIM***, ***PGD*** and ***Jitter*** choose L_∞, while ***CW*** and ***PGDL2*** choose L_2 to measure the perturbation range and the attack strength.

From the comparison results shown in Figure 4, we can find robust accuracy of the compressed models attacked by the ***Spear*** attack is similar or lower than applying the other reference adversarial attack algorithms. The obvious difference is the ***Spear*** attack has the trivial influence on the dense model than the other algorithm. So it proves that the ***Spear*** attack can more effectively generate the attack samples for

[1]https://github.com/NVIDIA/apex/tree/master/apex/contrib/sparsity

[2]https://github.com/NVIDIA/TensorRT/tree/master/tools/pytorch-quantization

[3]https://github.com/NVIDIA/TransformerEngine

[4]https://github.com/pytorch/vision

[5]https://github.com/microsoft/Swin-Transformer

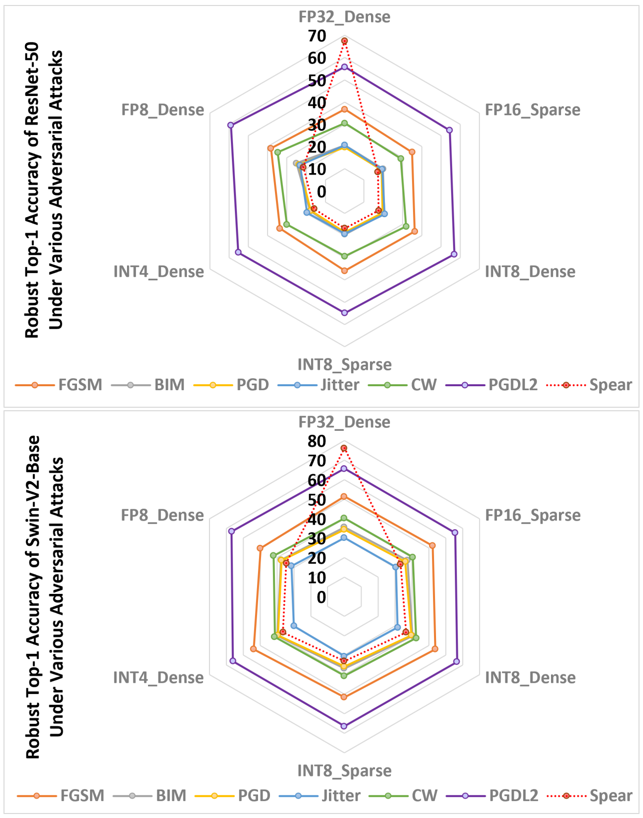


Figure 4: Comparison of ***Spear*** and the state-of-the-art adversarial attacks effectiveness on image classification models: (***Upper***) ResNet-50, (***Lower***) Swin-V2 Base, i.e., robust Top-1 accuracy of various compressed models under different adversarial attacks.

these compressed classification models and lead them to the wrong categories without causing much trouble to the original dense model.

4.3 Effectiveness Experiments for Detection Task

To evaluate the effectiveness of the ***Spear*** attack on the detection task, RetinaNet [Lin *et al.*, 2017], and Mask R-CNN [He *et al.*, 2017][6] are chosen as the experiment target models.

The sparse models are compressed with the fine-grained gradual pruning [Zhu and Gupta, 2018] (****-FINE***) and Sparse Tensor Cores pattern [Yu, 2021] (****-STC***) methods. The quantized models are obtained with ***QAT*** method. ***RN50*** and ***RN101*** in the brackets represent the ResNet-50 and ResNet-101 models served as the backbone of the detection networks. ***AP*** and ***AR*** represent the average precision and average recall metrics. We evaluate the robust ***AP*** and ***AR*** on the test set of **COCO** [Lin *et al.*, 2014] dataset. The loss adjustment parameters among the prediction loss (α_1, α_2), the distillation loss (β_1, β_2) and the adversarial loss (γ) apply 1, 1.5, 3, 6, 1.5, respectively. The results are shown in Table 1. (The variance is within ± 0.15 for ***AP***, and ± 0.21 for ***AR*** with different random seeds.)

Network	Compression Method & Ratio	Data Type	Attack Type	Perturbation Epsilon (ϵ) 0.001		0.1	
				Box AP	Box AR	Box AP	Box AR
RetinaNet (RN50)	0%-Dense	FP32	FGSM	38.663	54.520	34.586	48.497
			DeepFool	38.317	54.447	33.501	47.841
			PGD	38.252	54.192	32.201	46.323
	90%-FINE	FP32	FGSM	37.783	53.281	33.725	47.372
			DeepFool	37.454	53.206	32.798	46.595
			PGD	37.268	52.990	31.347	45.346
			Spear	**35.712**	**50.326**	**30.412**	**43.515**
	50%-STC	FP16	FGSM	38.094	53.724	34.181	47.961
			DeepFool	37.909	53.675	33.669	46.731
			PGD	37.857	53.511	33.101	46.307
			Spear	**35.152**	**49.480**	**31.864**	**45.886**
	QAT	INT8	FGSM	38.730	54.659	34.666	48.711
			DeepFool	38.371	54.516	34.345	48.377
			PGD	38.337	54.356	33.667	48.088
			Spear	**36.210**	**50.849**	**32.455**	**47.175**
Mask R-CNN (RN101)	0%-Dense	FP32	FGSM	41.954	55.761	41.431	55.018
			DeepFool	41.931	55.724	38.901	52.625
			PGD	41.920	55.713	35.146	46.615
	90%-FINE	FP32	FGSM	41.331	55.110	40.884	54.431
			DeepFool	41.341	55.026	38.327	51.902
			PGD	41.311	54.942	34.689	46.121
			Spear	**38.648**	**51.588**	**32.276**	**43.375**
	50%-STC	FP16	FGSM	41.998	55.817	41.487	55.077
			DeepFool	41.981	55.800	38.942	52.700
			PGD	41.976	55.779	35.184	46.677
			Spear	**38.214**	**50.718**	**31.980**	**42.438**
	QAT	INT8	FGSM	42.408	56.358	42.010	55.798
			DeepFool	42.251	56.209	39.308	53.397
			PGD	42.045	56.065	35.568	47.268
			Spear	**39.537**	**52.523**	**33.061**	**43.928**

Table 1: ***Spear*** attack effectiveness on detection task

4.4 Visualization Experiments

This section provides the visualization results to reveal how the "attention" focused regions change during the ***Spear*** attack process. We apply the Class Activation Mapping (**CAM**) tool [Zhou *et al.*, 2016] to the dense *ResNeXt-101* model without attack, the quantized *ResNeXt-101* model during and after the attack process. **CAM** can highlight the importance of the image region to the final prediction. The visualization results are shown in Figure 5.

For **CAM**, the red color highlight the "attention" focused areas of each model. For the example in ***Line 1***, the dense *ResNeXt-101* model without attack classifies the ground truth image into the ***tench*** category. During the ***Spear*** attack process, the "attention" focus start to change from the ***tench*** to the ***reel*** region. And finally the compressed model after the ***Spear*** attack process classifies the same ground truth image into the ***reel*** category.

For the example in ***Line 2***, the dense *ResNeXt-101* model without attack classifies the ground truth image into the ***tape player*** category. During the ***Spear*** attack process, the "attention" focus start to change. The compressed model during the attack process classifies the image into the ***cash machine*** category. And finally the compressed model after the ***Spear*** attack process classifies the same ground truth image into the ***weighing machine*** category.

The **CAM** visualization results explain the adversarial attack can guide the "attention" of the model to the different areas on the ground truth images, which finally misleading to the wrong classification.

4.5 Ablation Experiments and Insights

In this experiment, We want to check the contribution of each component in ***Spear*** to the final adversarial attack effect on

[6]https://github.com/facebookresearch/detectron2

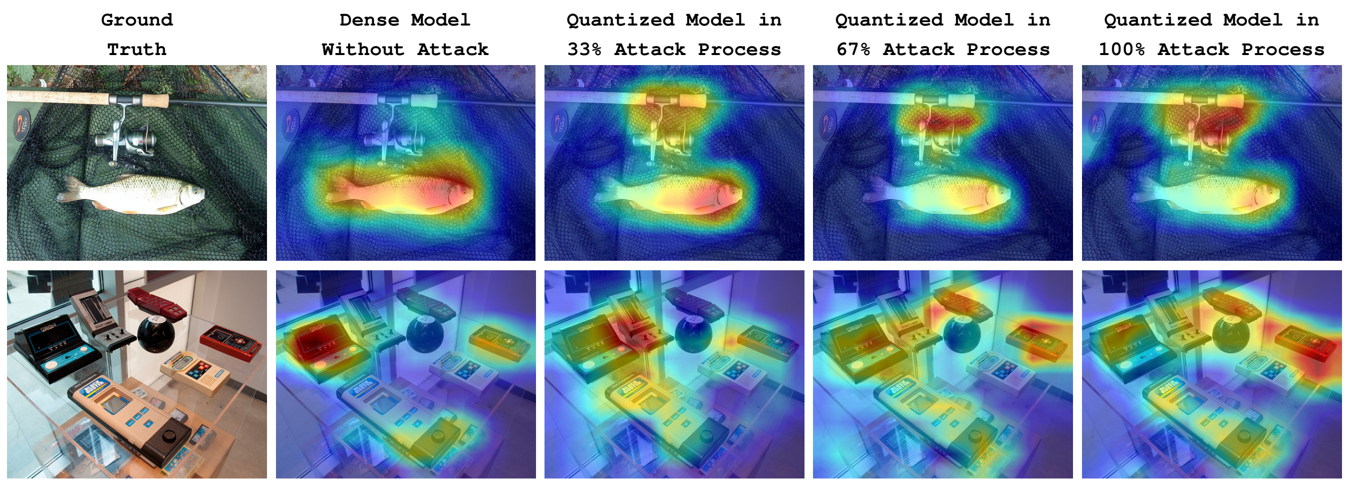

Ground Truth
Dense Model Without Attack
Quantized Model in 33% Attack Process
Quantized Model in 67% Attack Process
Quantized Model in 100% Attack Process

Figure 5: Class activation mapping visualization for classification task under the ***Spear*** evaluation. The dense *ResNeXt-101* model without attack and the compressed *ResNeXt-101* model after the ***Spear*** attack have almost the same *Top-1 and Top-5* metric. However they still have very different behaviors and "attention" focused areas to specific cases caused by the ***Spear*** attack.

the compressed model. Then we can have a deep insight into why ***Spear*** can outperform state-of-the-art methods and whether we can further improve it. We include five key components which may have an apparent potential contribution.

- **A:** Prediction loss for benign samples: $\boldsymbol{L}_p(\boldsymbol{I}_b)$.
- **B:** Prediction loss for attack samples: $\boldsymbol{L}_p(\boldsymbol{I}_a)$.
- **C:** Distillation loss for benign samples: $\boldsymbol{L}_d(\boldsymbol{I}_b)$.
- **D:** Distillation loss for attack samples: $\boldsymbol{L}_d(\boldsymbol{I}_a)$.
- **E:** Adversarial loss: $\boldsymbol{L}_a\,(\boldsymbol{S}_a, \boldsymbol{S}_d)$.

In this ablation experiment, five independent ***Spear*** attack models are trained on **ImageNet** training dataset. Each model lacks one of the aforementioned key component. For fairness, the loss adjustment parameters (α_1, α_2, β_1, β_2, γ) all apply 1. The results are shown in Table 2.

Network	Compression Method & Ratio	Data Type	Components	Perturbation Epsilon (ϵ) 0.001	0.01	0.1
				Top-1 %	Top-1 %	Top-1 %
ResNet-50	90%-FINE	FP32	**Complete**	**69.535**	**50.636**	**19.790**
			Without A	69.821	52.324	22.459
			Without B	70.628	56.772	30.828
			Without C	70.001	52.965	23.037
			Without D	71.218	57.390	31.227
			Without E	70.415	54.428	26.563
	PTQ	INT8	**Complete**	**68.346**	**51.729**	**20.319**
			Without A	70.227	52.986	22.072
			Without B	72.134	57.430	25.283
			Without C	70.457	53.389	23.413
			Without D	73.479	58.287	26.279
			Without E	72.892	57.726	25.852

Table 2: Ablation study of key components for ***Spear*** attack.

From the quantitative results shown in Table. 2, we can get the following conclusions.

- Loss items for attack samples have more contributions than those for benign samples. (Robust accuracy without **B** or **D** is higher than the counterpart without **A** or **C**).
- Distillation loss items have more contributions than prediction loss items. (Robust accuracy without **C** or **D** is higher than the counterpart without **A** or **B**).
- Adversarial loss items have more contributions than loss items for benign samples, but roughly less than loss items for attack samples. (Robust accuracy without **E** is higher than the counterpart without **A** or **C**, but is roughly lower or slightly higher than the counterpart without **B** or **D**).
- The quantized model is more sensitive to the adversarial loss items than the sparse model. (Robust accuracy without **E** for the quantized model has a more noticeable increase than the sparse model).

From the ablation experiments, we can see each component contributes to the final success of the ***Spear*** attack performance. However, the importance of each component is not the same. The ***Spear*** target is searching for the specific attack samples, which can maximally deteriorate the compressed model while keeping the 'normal' behavior of the dense counterpart. *So it is understandable that the loss items from the attack samples have more contribution.* Inspired by **KD**, the soft predictions could provide more knowledge about the whole dataset distribution than the hard predictions for most cases. *It helps to explain why the distillation loss items have more contributions than prediction loss items.* The induction of the adversarial loss item aims to minimize the perturbation from the neural model perspective and within the perturbation constraint simultaneously. *And it is proved to be very helpful for improving the attack efficiency and performance from the experimental results.*

5 Conclusions

We proposed the ***Spear*** attack method, which can effectively generate the attack samples to evaluate the adversarial robustness of the compressed models in this paper. It is a unified adversarial attack framework that is effective, efficient, and general to most of the model compression pattern. We evaluate the effectiveness of the proposed ***Spear*** attack technique on various tasks on large-scale datasets.

Ethical Statement

We want to emphasis that **the ultimate goal for our study of this specific adversarial attack for the compressed models is to effectively find the vulnerabilities and finally fix them to improve the robustness and safety of the AI models and systems**. We have some initial results to show that we can improve the robustness of the compressed models by applying the adversarial training workflow with the attack samples generated by the ***Spear*** method. But we will leave it for in-depth study to defend the adversarial attacks effectively.

We also encourage the community to understand and mitigate the risks arising from the ***Spear*** attack. As the ***Spear*** attack has the power to generate the misleading attack samples effectively, we should notice the risk that it can be used to misrepresent objective truth or to attack the neural models intentionally. However, we expect such misuse will become ineffectual as detection and defense techniques improve; these techniques may also benefit from our findings.

Acknowledgements

This work is supported by National Natural Science Foundation of China (No. 62071127, and 62101137), National Key Research and Development Program of China (No. 2022ZD0160100), Shanghai Natural Science Foundation (No. 23ZR1402900), Shanghai Municipal Science and Technology Major Project (No.2021SHZDZX0103).

The computations in this research were performed using the CFFF platform of Fudan University.

References

[Carlini and Wagner, 2017] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[Cheng *et al.*, 2021] Yupeng Cheng, Qing Guo, Felix Juefei-Xu, Wei Feng, Shang-wei Lin, Weisi Lin, and Yang Liu. Pasadena: Perceptually aware and stealthy adversarial denoise attack. *IEEE Transactions on Multimedia*, 2021.

[Deng *et al.*, 2009] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009.

[Goodfellow *et al.*, 2014] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[Gui *et al.*, 2019] Shupeng Gui, Haotao N Wang, Haichuan Yang, Chen Yu, Zhangyang Wang, and Ji Liu. Model compression with adversarial robustness: A unified optimization framework. *Advances in Neural Information Processing Systems*, 32:1285–1296, 2019.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[He *et al.*, 2017] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2961–2969, 2017.

[Hinton *et al.*, 2015] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[Kianpour and Wen, 2019] Mazaher Kianpour and Shao-Fang Wen. Timing attacks on machine learning: State of the art. In *Proceedings of SAI Intelligent Systems Conference*, pages 111–125. Springer, 2019.

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

[Krizhevsky *et al.*, 2009] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[Kurakin *et al.*, 2018] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018.

[LeCun *et al.*, 1998] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[Lin *et al.*, 2014] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[Lin *et al.*, 2017] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.

[Liu *et al.*, 2022] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12009–12019, 2022.

[Madry *et al.*, 2017] 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.

[Micikevicius *et al.*, 2022] Paulius Micikevicius, Dusan Stosic, Neil Burgess, Marius Cornea, Pradeep Dubey, Richard Grisenthwaite, Sangwon Ha, Alexander Heinecke, Patrick Judd, John Kamalu, et al. Fp8 formats for deep learning. *arXiv preprint arXiv:2209.05433*, 2022.

[Moosavi-Dezfooli *et al.*, 2016] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2574–2582, 2016.

[NVIDIA, 2017] NVIDIA. NVIDIA Tesla V100 GPU Architecture. https://images.nvidia.com/content/volta-architecture/pdf/volta-architecture-whitepaper.pdf, 2017. Accessed: 2024-05-31.

[NVIDIA, 2020a] NVIDIA. NVIDIA A100 Tensor Core GPU Architecture. https://images.nvidia.com/aem-dam/en-zz/Solutions/data-center/nvidia-ampere-architecture-whitepaper.pdf, 2020. Accessed: 2024-05-31.

[NVIDIA, 2020b] NVIDIA. NVIDIA Tensor Core. https://www.nvidia.com/en-us/data-center/tensor-cores, 2020. Accessed: 2024-05-31.

[Papernot *et al.*, 2016] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[Papernot *et al.*, 2017] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, pages 506–519, 2017.

[Paszke *et al.*, 2017] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

[Sap *et al.*, 2020] Maarten Sap, Saadia Gabriel, Lianhui Qin, Dan Jurafsky, Noah A Smith, and Yejin Choi. Social bias frames: Reasoning about social and power implications of language. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pages 5477–5490, 2020.

[Schwinn *et al.*, 2023] Leo Schwinn, René Raab, An Nguyen, Dario Zanca, and Bjoern Eskofier. Exploring misclassifications of robust neural networks to enhance adversarial attacks. *Applied Intelligence*, pages 1–17, 2023.

[Shumailov *et al.*, 2019] Ilia Shumailov, Yiren Zhao, Robert Mullins, and Ross Anderson. To compress or not to compress: Understanding the interactions between adversarial attacks and neural network compression. *Proceedings of Machine Learning and Systems*, 1:230–240, 2019.

[Song *et al.*, 2020] Chang Song, Elias Fallon, and Hai Li. Improving adversarial robustness in weight-quantized neural networks. *arXiv preprint arXiv:2012.14965*, 2020.

[Szegedy *et al.*, 2013] 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.

[Szegedy *et al.*, 2014] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2014.

[Ye *et al.*, 2019] Shaokai Ye, Kaidi Xu, Sijia Liu, Hao Cheng, Jan-Henrik Lambrechts, Huan Zhang, Aojun Zhou, Kaisheng Ma, Yanzhi Wang, and Xue Lin. Adversarial robustness vs. model compression, or both? In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 111–120, 2019.

[Yu, 2021] Chong Yu. Minimally invasive surgery for sparse neural networks in contrastive manner. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3589–3598, 2021.

[Zhou *et al.*, 2016] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.

[Zhu and Gupta, 2018] Michael Zhu and Suyog Gupta. To prune, or not to prune: exploring the efficacy of pruning for model compression. In *International Conference on Learning Representations*, 2018.