Masking Adversarial Damage: Finding Adversarial Saliency
for Robust and Sparse Network

Byung-Kwan Lee[*], Junho Kim[*], Yong Man Ro[†]

Image and Video Systems Lab, School of Electrical Engineering, KAIST, South Korea

{leebk, arkimjh, ymro}@kaist.ac.kr

Abstract

Adversarial examples provoke weak reliability and potential security issues in deep neural networks. Although adversarial training has been widely studied to improve adversarial robustness, it works in an over-parameterized regime and requires high computations and large memory budgets. To bridge adversarial robustness and model compression, we propose a novel adversarial pruning method, Masking Adversarial Damage (MAD) that employs second-order information of adversarial loss. By using it, we can accurately estimate adversarial saliency for model parameters and determine which parameters can be pruned without weakening adversarial robustness. Furthermore, we reveal that model parameters of initial layer are highly sensitive to the adversarial examples and show that compressed feature representation retains semantic information for the target objects. Through extensive experiments on three public datasets, we demonstrate that MAD effectively prunes adversarially trained networks without loosing adversarial robustness and shows better performance than previous adversarial pruning methods.

1. Introduction

Deep neural networks (DNNs) have achieved impressive performances in a wide variety of computer vision tasks (*e.g.*, image classification [19,26], object detection [30,40], and semantic segmentation [5, 18]). Despite the breakthrough outcomes, DNNs are easily deceived from adversarial attacks with carefully crafted perturbations [4, 9, 32, 50]. Injecting these perturbations into benign images generates adversarial examples which hinder the decision-making process of DNNs. Although the perturbations are imperceptible to humans, they easily induce vulnerable features in DNNs [22, 23]. Due to such fragility, various deep learning applications have suffered from potential security issues that induce weak reliability of DNNs [2, 44, 55].

Accordingly, to achieve robust and reliable DNNs, a lot of adversarial research have been dedicated to presenting powerful adversarial attack and defense algorithms in the sense of cat-and-mouse games. Among various methods, adversarial training (AT) [3,9,27,32] has been widely studied to improve adversarial robustness so far, where DNNs are trained with adversarial examples. Madry *et al.* [32] have shown that adversarially trained models are robust against several white-box attacks with the knowledge of model parameters. Besides, recent studies [56,62] have further enhanced the robustness by adding various regularizations to fully utilize not only the adversarial examples but also benign examples for model generalization.

Orthogonal to the adversarial issue, most of the AT-based methods work in an over-parameterized regime to achieve the robustness, thus they induce higher computations and require larger memory budgets than benign classifiers [32]. Thereby, applying AT-based methods to resource-constrained devices is burdensome and known as a critical limitation. To bridge adversarial robustness and model compression, several studies [37, 45, 47, 52] have introduced adversarial pruning mechanisms to reduce its model capacity while preserving the adversarial robustness.

In standard training procedure, a promising pruning method is to remove the lowest weight magnitudes (LWM) [14, 15], assuming that small magnitudes affect the least changes of the model prediction. Under its assumption, Sehwang *et al.* [45] have proposed a 3-step pruning method (pre-training, pruning, and fine-tuning) for the adversarial examples. Several works [12, 59] further have enhanced LWM pruning methods by employing alternating direction method of multipliers (ADMM) [63] or Beta-Bernoulli dropout [31] to eliminate unuseful model parameters. Recently, Sehwag *et al.* [46] have argued that despite successful results of LWM pruning, such heuristic pruning methods cause performance degradation when integrated with adversarial training. Accordingly, they have formulated a way of finding importance scores [39] and removed model parameters with low importance scores for adversarial training loss. We further focus on how to reflect corre-

---

[*]Equal contribution. [†] Corresponding author.

lations of each model parameter to prune multiple parameters at once and theoretically formulate local geometry of adversarial loss to identify which combinatorial model parameters affect model prediction in adversarial settings.

In this paper, we present a novel adversarial pruning method, namely Masking Adversarial Damage (MAD) that uses second-order information (Hessian) of the adversarial loss with masking model parameters. Motivated by Optimal Brain Damage (OBD) [29] and Optimal Brain Surgeon (OBS) [17] employing Hessian of the loss function, we devise a way of estimating *adversarial saliency* for model parameters by considering their correlated conjunctions, which can more precisely represent the connectivity of model parameters for the adversarial prediction.

In recent works [49, 53], approximating Hessian of multiple parameters is regarded as computationally prohibitive. Alternatively, computing each parameter's importance and sorting it with pruning statistics is a practical approach in standard pruning. Bringing in such combinatorial problem into adversarial settings, we aim to prune adversarially less salient parameters that cannot arouse adversarial vulnerability. To that end, we first optimize masks for DNNs to be predictive to adversarial examples and approximate Hessian of the adversarial loss by utilizing optimized masks. To effectively compute Hessian, we introduce a block-wise K-FAC approximation that can consider the local geometry of the adversarial loss at multiple parameters points. Based on the change of the adversarial loss along with mask-applied parameters, we can track how sensitively specific conjunctions of model parameters respond to adversarial examples and compress DNNs without weakening the robustness.

For the proposed method, we thoroughly analyze adversarial saliency and compressed feature representation, and we reveal that: i) model parameters of initial layer are highly sensitive to adversarial perturbation, ii) MAD can retain semantic information of target objects even with the high pruning ratio. Through extensive experiments on MAD in three major datasets, we corroborate that MAD can effectively compress DNNs without losing adversarial robustness and show better adversarial defense performance than the previous adversarial pruning methods.

The major contributions of this paper are as follows:

- We present a novel adversarial pruning method, Masking Adversarial Damage (MAD) which can precisely estimate adversarial saliency for model parameters using second-order information.

- By analyzing MAD, we investigate that the model parameters of initial layer are highly sensitive to adversarial examples, and compressed feature representation retains semantic information of target objects.

- Through extensive experiments, we demonstrate the

effectiveness of compression capability as well as the adversarial robustness for the proposed method.

2. Background and Related Work

In this section, we specify the notations used in our paper and summarize the related works on adversarial training and model compression.

Notations. Let x denote a benign image and y denote a target label corresponding to the input image. Let $\mathcal{D}$ indicate a dataset such that $(x, y) \sim \mathcal{D}$. A deep neural network f parameterized by model parameters w is denoted by f_w. An adversarial example is represented by $x_{\text{adv}} = x + \delta^*$, where δ^* indicates adversarial perturbation as follows:

$$\delta^* = \arg \max_{\|\delta\|_\infty \leq \gamma} \mathcal{L}(f_w(x + \delta), y), \qquad (1)$$

where $\mathcal{L}$ denotes a pre-defined loss function. In this paper, we use the cross-entropy loss for image classification. We regard δ as l_∞ perturbation within γ-ball (*i.e.*, perturbation budget) to be imperceptible to humans such that $\|\delta\|_\infty \leq \gamma$. Here, $\|\cdot\|_\infty$ describes l_∞ perturbation magnitude.

2.1. Adversarial Training

Adversarial training [9, 27, 32] becomes one of the intensive robust optimization methods in deep learning field, which allows DNNs to learn robust parameters against adversarial examples. In earlier work, Szegedy *et al.* [50] have found that the specific nature of the adversarial examples is not a random artifact of deep learning. Then, Goodfellow *et al.* [9] have introduced a single-step attack of Fast Gradient Sign Method (FGSM) that efficiently generates adversarial examples using first gradient of loss on back propagation and trained DNNs with FGSM-based adversarial examples to have an additional regularization benefit.

Madry *et al.* [32] have studied adversarial robustness in a model through the lens of robust optimization (min-max game). Then, they have modified empirical risk minimization (ERM) to incorporate adversarial perturbations and proposed an adversarial training with a multi-step attack of Projected Gradient Descent (PGD). The formulation of the min-max game for the robust optimization under l_∞ can be written as:

$$w^* = \arg \min_w \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\max_{\|\delta\|_\infty \leq \gamma} \mathcal{L}\left(f_w(x + \delta), y\right) \right]. \quad (2)$$

They draw a universal first-order attack through the inner maximization problem while outer minimization iteratively represents training DNNs with adversarial examples created at inner maximization one. Through it, they have designed a reliable method of adversarial training with PGD and empirically demonstrated that the acquired model parameters w^* provide the robustness against a first-order adversary as a natural security guarantee. It allows us to cast both attacks and defenses into a common theoretical framework.

2.2. Model Compression

Since DNNs generally have intrinsic over-parameterized properties, they induce high computation and memory inefficiency to apply on resource-limited devices [14]. Therefore, several works [21, 29, 57] have proposed to prune weight parameters based on the lowest weight magnitudes (LWM). Besides, various approaches have been introduced to improve model compression utilizing absolute weight value [15], second-order information [7, 17], and alternating direction method of multipliers (ADMM) optimization [63]. More recently, to estimate enhanced loss change when removing a single parameter, several works based on OBD [29] and OBS [17] have utilized layer-wise pruning framework (L-OBS) [7], K-FAC approximation [53], and Woodbury formulation (WoodFisher) [49] to effectively compute (inverse) Hessian matrix.

In parallel with various model compression methods for benign classifiers, achieving compression in adversarial settings also has drawn attention to mitigating expensive computational costs that upholds adversarial robustness of DNNs. Several works [13, 54] have investigated the relationship between the adversarial robustness and model compression and theoretically concluded that moderately compressed model may improve adversarial robustness.

Sehwang *et al*. [45] have demonstrated that LWM pruning can increase adversarial robustness for the adversarial examples. After that, LWM-based methods in adversarial settings become a typical pruning approach. Ye *et al*. [59] have improved LWM pruning method by employing ADMM optimization. Gui *et al*. [12] have introduced a unified framework called ATMC including pruning, factorization, and quantization using ADMM. Recently, Sehwang *et al*. [46] have argued that such heuristic LWM-based methods provoke performance degradation and proposed a generalized formulation of computing weight importance for adversarial loss using importance score optimization [39]. Madaan *et al*. [31] have defined adversarial vulnerability in feature space and integrated it into adversarial training to suppress feature-level distortions. In this paper, we propose a method for estimating adversarial saliency using second-order information, which serves as a key to prune parameters without weakening adversarial robustness.

3. Methodology

3.1. Revisiting Second-order Information Pruning

Optimal Brain Damage (OBD). For well-trained DNNs, Lecun *et al*. [29] have removed a single model parameter and computed a loss change obtained from the removed single parameter. Their goal in pruning is to remove parameters that do not significantly change the loss function. Thus, they can selectively delete each model parameter corresponding to small loss change with its theoretical

evidence of capturing sensitivity in DNNs. The loss change can be written as a form of Taylor expansion:

$$\Delta\mathcal{L} \approx \underbrace{\frac{\partial\mathcal{L}}{\partial w}}_{\approx 0}\Delta w + \frac{1}{2}\Delta w^T \mathrm{H}\Delta w, \qquad (3)$$

where it satisfies $\Delta\mathcal{L} = \mathcal{L}(w + \Delta w) - \mathcal{L}(w)$, and w denotes local *Maximum A Posteriori* (MAP) parameters after training DNNs with its convergence. Note that there is an assumption that the first gradient of loss function closes to zero due to using the well-trained DNNs with their local MAP parameters around a local mode at w. Here, H indicates Hessian matrix computed by the second derivative of loss function $\mathcal{L}$ at model parameters w, such that it satisfies $\mathrm{H} = \nabla_w^2\mathcal{L}$. In calculating Hessian, Fisher information is employed to approximate it: $\nabla_w^2\mathcal{L} \simeq \mathbb{E}_\mathcal{D}[\nabla_w\mathcal{L}^2]$. (see Appendix A)

However, due to the intractability of calculating full Hessian matrix H in DNNs, they have focused on its diagonal terms to approximate the loss change $\Delta\mathcal{L}$. The formulation for estimating the loss change can be written as follows:

$$\min_{\Delta w \in \mathbb{R}^d} \frac{1}{2}\Delta w^T \mathrm{H}\Delta w, \quad \text{s.t. } e_k^T\Delta w + w_k = 0, \qquad (4)$$

where e_k denotes a canonical basis vector in k^{th} dimension. With only of the diagonal considered, the corresponding solution for the loss change is deterministic to $\Delta\mathcal{L}_{\text{OBD}} = \frac{1}{2}w_k^2\mathrm{H}_{kk}$ where a single parameter w_k is pruned. Once $\Delta\mathcal{L}_{\text{OBD}}$ is computed for all model parameters, then model parameters are cut out in order of small loss change by sorting $\Delta\mathcal{L}_{\text{OBD}}$. Note that d specifies total size of model parameters w in DNNs. In addition, Hessian is a symmetric matrix having shape of $\mathbb{R}^{d \times d}$ with positive semi-definite.

Optimal Brain Surgeon (OBS). OBD only addresses diagonal terms in Hessian matrix assuming model parameters are uncorrelated. Thereby, it is impossible to compute the loss change considering correlated model parameters in DNNs for generalization. To handle such limitation, Hassibi *et al*. [17] have proposed to compute the loss change with regards to full Hessian matrix. Its corresponding solution is also deterministic to $\Delta\mathcal{L}_{\text{OBS}} = \frac{1}{2}w_k^2/[\mathrm{H}^{-1}]_{kk}$ by Lagrangian relaxation (see Appendix B). After getting $\Delta\mathcal{L}_{\text{OBS}}$ for all model parameters, OBS equally removes model parameters in order of the small loss change.

3.2. Masking Adversarial Damage (MAD)

Although its successful achievements in standard pruning, it is difficult to directly apply it to adversarial pruning due to one major reason. The main principle of possibly computing the loss change $\Delta\mathcal{L}$ in Eq. (3) is based on the fact that $\mathcal{L}(w)$ is an important criterion to ensure highly predictive DNNs. However, once adversarial examples are

given, $\mathcal{L}(w)$ cannot be operated to be predictive anymore, where the first order of Taylor expansion $\frac{\partial \mathcal{L}}{\partial w}$ is no longer zero in adversarial settings. It is attributed to the performance degradation derived from the fragility of DNNs.

Hence, we need a highly optimized loss $\mathcal{L}(w)$ to construct Eq. (3), which signifies still predictive on the adversarial examples. Nonetheless, most of the existing adversarially trained models do not reach satisfactory robustness under adversarial attack scenario compared with their benign accuracy. To tackle such a problem, we employ adaptive masks directly multiplying to model parameters and optimize them to meet local convergence to be a highly predictive model for adversarial examples. The formulation of finding the masks can be written as follows:

$$m^* = \arg \min_{m \in \mathbb{R}^d} \max_{\|\delta\|_\infty \leq \gamma} \mathcal{L}(f_{w_m}(x + \delta), y), \qquad (5)$$

where a model f is parameterized by mask-applied parameters $w_m = w \odot m$. Here, $\odot$ indicates the Hadamard product. Note that dimension of model parameters w is same as that of masks m, thus d equally represents total size of w and m in DNNs. In addition, m belong to continuous real number set in $[0, 1]^d \in \mathbb{R}^d$.

Through the mask optimization, we can effectively address the aforementioned issue to compute the loss change in adversarial settings. Then, we can re-interpret the loss change for the given adversarial examples as follows:

$$\mathcal{L}_{x_{\mathrm{adv}}}(w) - \mathcal{L}_{x_{\mathrm{adv}}}(w_{m^*}) = \frac{1}{2} \Delta w^T \mathrm{H} \Delta w, \qquad (6)$$

where $\mathcal{L}_{x_{\mathrm{adv}}}(w)$ indicates an adversarial loss that have a high value due to broken model predictions for the given adversarial examples: $x_{\mathrm{adv}} = x + \delta^*$. In addition, $\mathcal{L}_{x_{\mathrm{adv}}}(w_{m^*})$ denotes a highly optimized adversarial loss against x_{adv}, which has a smaller value than previous one. Here, Hessian $\mathrm{H} \in \mathbb{R}^{d \times d}$ represents second-order information helping to build local geometry of the model parameters for capturing adversarially saliency factors.

Compared with previous works [17, 29] that remove a single parameter each, the procedures of Eq. (5) and Eq. (6) enable to prune multiple parameters at once, which have a great advantage of considering impacts of intrinsic correlated connectivity among multiple parameters for model predictions. On the other hand, to feasibly make use of the loss change, we should find out how to possibly calculate a parameter variation, denoted by Δw. In OBD and OBS, they have the tractable constraint for the variation Δw removing a single model parameter: $e_k^T \Delta w + w_k = 0$. Therefore, they can approach the loss change by a deterministic solution, as mentioned in Sec. 3.1.

However, we cannot straightforwardly get the deterministic solution for Eq. (6), since the masks are tangled with the variation Δw in DNNs. As long as such limitation exists, we should practically figure out Δw with respect to the

masks m. Thus, we provide the following constraint equation for pruning multiple parameters at once:

$$w \odot (1 - m) + \Delta w = \mathbf{0}, \qquad (7)$$

where $\mathbf{0}$ indicates zero vector in $\mathbb{R}^d$. For extreme mask case, once $k^{\mathrm{th}}(= 1, 2, \cdots, d)$ mask denoted by m_k get closes to zero value such that $w_{m_k} = w_k m_k = 0$, then Eq. (7) satisfies $w_k + \Delta w_k = 0$ which explicitly means that the kth mask-applied parameter w_{m_k} is removed to zero.

Conversely, once m_k gets close to one value such that $w_{m_k} = w_k$, then the constraint equation satisfies $\Delta w_k = 0$, which implies that w_{m_k} is not a target to remove. If a mask is above zero value within one, it provokes an effect of smoothing a parameter variation Δw, which signifies that w_{m_k} is not strictly erased yet some of them remains.

From these perspectives, masks can be formulated with parameter variation Δw, regarding correlation for all of the model parameters in DNNs. To compromise it on the variation Δw, we apply this constraint to Eq. (6). Then, we can simply present the loss change in adversarial settings. At this point, we define the overall procedures of computing the loss change and pruning as *Masking Adversarial Damage* (MAD). Here, the loss change of MAD $\Delta \mathcal{L}_{\mathrm{MAD}}$ can be formulated as:

$$\Delta \mathcal{L}_{\mathrm{MAD}} = \frac{1}{2} [w \odot (1 - m^*)]^T \mathrm{H} [w \odot (1 - m^*)], \qquad (8)$$

where it satisfies $\Delta \mathcal{L}_{\mathrm{MAD}} = \mathcal{L}_{x_{\mathrm{adv}}}(w) - \mathcal{L}_{x_{\mathrm{adv}}}(w_{m^*})$. Here, we call the loss change $\Delta \mathcal{L}_{\mathrm{MAD}}$ as *adversarial saliency* to indicate its effect to model prediction. This mathematical expansion of MAD is completely aligned with theoretical analysis of Taylor expansion (see Appendix C).

Beyond its numerical loss change by itself, it provides us internal information in DNNs about which model parameter is a much or less adversarially important factor. Once we compute adversarial saliency for all model parameters and observe that some parameters have large saliency, we infer that they easily flip model predictions for adversarial examples, even with their small changes. Thus, they can be regarded as salient factors to get a knowledge of the robustness through adversarial training. On the other hand, in the contrary case, we interpret that the model parameters with small saliency are invariant to the model prediction. Hence, we can think of they are insignificant against adversarial examples. Accordingly, depending on the value size of adversarial saliency, we can identify which model parameters can be pruned without weakening adversarial robustness.

3.3. Block-wise K-FAC Approximation

In the previous section, we have focused on how to develop the loss change $\Delta \mathcal{L}_{\mathrm{MAD}}$ in adversarial settings and how to compute variation Δw despite the absence of a deterministic solution. However, there is another important issue to compute second-order information as Hessian. This

is because Hessian is a huge matrix in terms of total parameter dimension, thus it remains challenging to efficiently reduce computational cost. To address it, there have been several methods of approximating Hessian by empirical Fisher Information [1,7,10,16,42,61], which can be written as:

$$H \simeq F = \frac{1}{N} \sum_{n=1}^{N} \{\nabla_w \mathcal{L}_n\}\{\nabla_w \mathcal{L}_n\}^T, \qquad (9)$$

where $\nabla_w \mathcal{L}_n$ indicates $\nabla_w \mathcal{L}(f_w(x_n), y_n)$ in n^{th} data sample such that N represents the total number of data samples. F denotes Fisher information matrix to possibly approximate Hessian in an empirical manner. Recently, various efficient approximation to Hessian has been proposed under *Kronecker-Factored Approximate Curvature* (K-FAC) [11,33,53] and WoodFisher approximation [49].

However, since classical loss change (Eq. (3)) for standard pruning has deterministic solutions for OBD and OBS, the several works of approximating Hessian does not have to fully compute the loss change. Instead, they figure out the loss change of $\Delta\mathcal{L}_{\text{OBD}} = \frac{1}{2}w_k^2 H_{kk}$ and $\Delta\mathcal{L}_{\text{OBS}} = \frac{1}{2}w_k^2/[H^{-1}]_{kk}$, where diagonal elements of Hessian or inverse Hessian are only used. On the other hand, in order to acquire precise $\Delta\mathcal{L}_{\text{MAD}}$ (*i.e.*, adversarial saliency), we should need to even consider off-diagonal terms in Hessian, but using full Hessian gives a striking computational burden. Hence, we introduce a block-wise K-FAC approximation to regard some blocks of the off-diagonal terms. The following Lemma 1 describes K-FAC approximation, which efficiently approximates Hessian in DNNs with convolution layer. K-FAC for the fully-connected layer is described in Appendix D, where the differences are dimension scales of spatial size and its corresponding gradient.

Lemma 1 *Kronecker-Factored Approximate Curvature.*
At l^{th} convolution layer in DNNs, let us define activation map $a \in \mathbb{R}^{c_{in} \times s_a}$, weight matrix $\mathcal{W} \in \mathbb{R}^{c_{out} \times c_{in}k^2}$, and layer output $z \in \mathbb{R}^{c_{out} \times s_z}$ such that it satisfies $z = \mathcal{W} * a$, $\nabla_{\mathcal{W}}\mathcal{L} = \sum_i \left[\nabla_{z_i}\mathcal{L} \times a_i^T\right]$, and $F = \mathbb{E}[\{\nabla_{\mathcal{W}}\mathcal{L}\}\{\nabla_{\mathcal{W}}\mathcal{L}\}^T]$, where i denotes spatial index. Note that c_{in} denotes channel number of lth layer, c_{out} denotes channel number of $(l+1)^{th}$ layer, s_a denotes spatial size of activation map a, s_z denotes spatial size of layer output z, and k denotes kernel size of weight.

$$F \approx \sum_i \mathbb{E}[\{\nabla_{z_i}\mathcal{L}\}\{\nabla_{z_i}\mathcal{L}\}^T] \otimes \sum_i \mathbb{E}[a_i a_i^T] = \mathcal{Z} \otimes \mathcal{A},$$

where $\mathcal{Z} \in \mathbb{R}^{c_{out} \times c_{out}}$ and $\mathcal{A} \in \mathbb{R}^{c_{in}k^2 \times c_{in}k^2}$ stand for correlation of layer output z and activation map a, respectively.

Decomposing Fisher information matrix F into each Kronecker Factor for the correlations $\mathcal{Z}$ and $\mathcal{A}$ with relatively low dimension not only reduces storage cost but also enables efficient computation. Then, we first expand the adversarial saliency $\Delta\mathcal{L}_{\text{MAD}}$ using K-FAC, which can be written in l^{th} convolution layer as:

$$\begin{aligned} \Delta\mathcal{L}_{\text{MAD}} &= \frac{1}{2}\Delta w^T \mathcal{Z} \otimes \mathcal{A}\Delta w \\ &= \frac{1}{2}\text{Tr}\left[\Delta w \Delta w^T \mathcal{Z} \otimes \mathcal{A}\right], \end{aligned} \qquad (10)$$

where the variation is denoted by $\Delta w = -w \odot (1-m)$ for MAD constraint, of which vector shape is $\mathbb{R}^{d=c_{out}c_{in}k^2}$ in the convolution layer.

Here, we develop a block-wise K-FAC that allows for efficiently computing the adversarial saliency $\Delta\mathcal{L}_{\text{MAD}}$. Since we cannot use full Hessian for all model parameters due to computational limitation and physical memory budget, we instead deal with only of the diagonal terms in the correlation $\mathcal{Z}$ of layer output z, where we consider off-diagonal terms in $\mathcal{Z}$ as zero such that $\mathcal{Z}_{ij} = 0$ $(i \neq j)$. Then, Fisher information matrix can be described with the block-wise K-FAC as follows:

$$F \approx \mathcal{Z} \otimes \mathcal{A} = \begin{bmatrix} \mathcal{Z}_{11}\mathcal{A} & 0 & \cdots & 0 \\ 0 & \mathcal{Z}_{22}\mathcal{A} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & \mathcal{Z}_{rr}\mathcal{A} \end{bmatrix}, \quad (11)$$

where $r = c_{out}$. By using this Fisher matrix F, we can efficiently compute the adversarial saliency from Eq. (10), which can be formulated as follows (see Appendix E):

$$\Delta\mathcal{L}_{\text{MAD}} = \sum_{i=1}^{r} \frac{\mathcal{Z}_{ii}}{2}\text{Tr}\left[\Delta\mathcal{W}_i \Delta\mathcal{W}_i^T \mathcal{A}\right], \qquad (12)$$

where we reshape variation vector Δw to variation matrix $\Delta\mathcal{W} \in \mathbb{R}^{c_{in}k^2 \times c_{out}}$ such that $\Delta\mathcal{W} = -\mathcal{W} \odot (1-\mathcal{M})$, where mask matrix is also reshaped to $\mathcal{M} \in \mathbb{R}^{c_{in}k^2 \times c_{out}}$. Note that $\Delta\mathcal{W}_i$ denotes i^{th} column vector in this variation matrix. In Eq. (10), it has $\mathcal{O}(c_{out}^4 c_{in}^4 k^8)$ computation complexity due to full Hessian, but we can reduce it to $\mathcal{O}(c_{out}^2 c_{in}^4 k^8)$ by using the block-wise K-FAC. We get a computational benefit per convolution layer by $c_{out}^2(= r^2)$ as the matrix size of the correlation $\mathcal{Z}$ for layer output z.

3.4. Pruning Low Adversarial Saliency

Based on Eq. (12), we finally get adversarial saliency for the given adversarial examples. In practice, there arises a natural phenomenon that the optimized mask m^* overfits too much to specific samples of adversarial examples, thus it cannot be generalized to other adversarial examples. Therefore, after we optimize masks and compute the adversarial saliency for each data sample, we average them to generalize in all of the data samples. Then, by controlling pruning ratio $p\%$, we can readily prune model parameters by eliminating them in order of low adversarial saliency. Here, we describe Algorithm 1 to explain MAD in detail.

Algorithm 1 Masking Adversarial Damage

Require: pruning ratio $p\%$
1: **for** $(x, y) \sim \mathcal{D}$ **do** ▷ Finding Adversarial Saliency
2:   $x_{\mathrm{adv}} \leftarrow \mathrm{attack}(x, y)$ ▷ PGD Attack
3:   $m^* \leftarrow \arg\min\limits_{m \in \mathbb{R}^d} \mathcal{L}(f_{w_m}(x_{\mathrm{adv}}), y)$ ▷ Finding Masks
4:   **for** l^{th} layer index **do**
5:     $\Delta\mathcal{W}^{(l)} \leftarrow -\mathcal{W}^{(l)} \odot (1 - \mathcal{M}^{*(l)})$ ▷ Constraint
6:     **for** i^{th} channel index **do** ▷ Block-wise K-FAC
7:       $\mathcal{B}_{ii}^{(l)} \leftarrow \frac{\mathcal{Z}_{ii}^{(l)}}{2}\{\Delta\mathcal{W}_i^{(l)}\}\{\Delta\mathcal{W}_i^{(l)}\}^T \mathcal{A}^{(l)}$
8:     **end for**
9:     $\Delta\mathcal{L}_{\mathrm{MAD}}^{(l)} \leftarrow \Delta\mathcal{L}_{\mathrm{MAD}}^{(l)} + \frac{1}{N}\mathrm{diag}(\mathcal{B}^{(l)})$ ▷ Average
10:   **end for**
11: **end for**
12: Sorting $\Delta\mathcal{L}_{\mathrm{MAD}}$ and Pruning $p\%$ parameters
13: **for** epoch **do** ▷ Adversarial Training $(\alpha : lr)$
14:   **for** $(x, y) \sim \mathcal{D}$ **do**
15:     $x_{\mathrm{adv}} \leftarrow \mathrm{attack}(x, y)$ ▷ PGD Attack
16:     $w \leftarrow w - \alpha \frac{\partial}{\partial w}\mathcal{L}(f_w(x_{\mathrm{adv}}), y)$ ▷ Update
17:   **end for**
18: **end for**

---

4. Experiments

4.1. Implementation Details

Datasets and Networks. We conduct exhaustive experiments on three datasets and three networks. For dataset, we use CIFAR-10 [25] and SVHN [34] which are well known as standard low dimensional datasets which equally have 32×32 dimensional images with 10 classes. In addition, to demonstrate generalization of MAD in a larger dataset, we adopt Tiny-ImageNet [28] with 64×64 pixels, which is a small subset of ImageNet separated in 200 different classes. For the three datasets, we train the following three networks: VGG-16 [48], ResNet-18 [20], and WideResNet-28-10 [60].

Mask Optimization. First, we adversarially train the networks based on a standard adversarial training (AT) [32]. After we complete to adversarially train the networks, we perform mask optimization in Eq. (5) to find out adversarial saliency for model parameters. To do this, we make use of Adam [24] with a learning rate of 0.1 and momentum of 0.9 for each data sample. In addition, we use 20 iterations to optimize masks, which allows sufficient convergence to predict well for adversarial examples. Note that since we also observe that in a standard network, mask optimization does not work well for the given adversarial examples [23], thus we cover the adversarially trained network only.

Pruning by MAD. We prune model parameters in order of low adversarial saliency. In addition, we adversarially train the networks under a sparse network across a pruning ratio $p\%$. For the setting of adversarial training, we follow

the conventional settings for l_∞ perturbation budget $8/255$. In addition, we generate adversarial examples by PGD attack [32] with random restarts, where we set attack steps to 10 and set attack step size to 0.0069 in training. Especially, since adversarially training Tiny-ImageNet is a computational burden, we employ fast adversarial training [58] with FGSM attack [9] where the step size is set to $1.25 \times 8/255$. For training hyper-parameter, we set a learning rate of 0.1 with SGD [43] in 60 epochs, and we early stop [41] to reduce overfitting harming adversarial robustness. In addition, we take a step scheduler to lower the learning rate by 0.1 times on each 30 and 45 epoch, and we set the weight decay parameter to 2×10^{-4}.

Adversarial Attacks. To fairly validate adversarial robustness, we adopt three standard attacks: FGSM [9], PGD [32], CW$_\infty$ [4], and two more recent advanced attacks with a budget-aware step size-free: AP (Auto-PGD) and parameter-free: AA (Auto-Attack) introduced by Francesco *et al.* [6]. In inference, PGD and AP have 30 steps with random starts where PGD has 0.0023 step size and AP has momentum coefficient $\rho = 0.75$. In addition, we use CW$_\infty$ attack on l_∞ perturbation budget $8/255$ by employing PGD gradient clamping with CW objective [4] on $\kappa = 0$.

4.2. Validating MAD

Adversarial Robustness. To validate the effectiveness of MAD, we compare adversarial robustness with advanced recent defense methods: TRADES [62], MART [56] and strong baselines for adversarial pruning: LWM [45], ADMM [59], and HYDRA [46]. To fairly compare the performance, the defense methods and the baselines are aligned with our experiment setup, so that we build them equally based on adversarially trained model (AT) on PGD attack [32]. Tab. 1 demonstrates two perspectives: i) MAD mostly outperforms the robustness of the baselines. ii) Despite using 10% of model parameters, MAD shows less performance degradation of the robustness than the baselines compared by the best performance with full parameters. For all of the conducted attacks, MAD has $4.2(\%)$ and $1.9(\%)$ degradation for CIFAR-10 and SVHN each on average of three networks. For the baselines, HYDRA has $7.2(\%)$ and $6.1(\%)$ degradation, ADMM has 8.7 and 8.3, and LWM has 9.4 and 8.4 for CIFAR-10 and SVHN. Besides, for benign accuracy, MAD has $0.4(\%)$ and $0.4(\%)$ degradation, HYDRA 3.4 and 4.1, ADMM has 1.4 and 0.3, and LWM has 4.8 and 3.9. Moreover, we also compare the robustness of MAD in Tiny-ImageNet with that of the baselines. As illustrated in Tab. 2, we corroborate that MAD also works well in a larger dataset, while preserving the robustness.

Ablation Study. Despite its theoretical evidence of capturing adversarial saliency and its experimental improvements to the robustness, it is necessary to investigate that removing model parameters in reverse order induces much

		CIFAR-10							SVHN						
	Method	AT	TRADES	MART	LWM	ADMM	HYDRA	MAD	AT	TRADES	MART	LWM	ADMM	HYDRA	MAD
Model	Sparsity	0%			90%				0%			90%			
	Params	14.7M			1.5M				14.7M			1.5M			
VGG	Clean	**81.3**	80.2	80.2	76.0	79.0	76.4	**81.4**	**93.2**	91.8	91.7	88.4	92.4	87.9	**92.8**
	FGSM	55.2	54.3	**57.9**	50.3	52.6	50.4	**57.0**	67.1	66.5	**68.4**	62.4	65.2	61.2	**68.2**
	PGD	50.2	50.5	**54.2**	46.8	47.4	47.4	**51.8**	56.0	57.3	**60.4**	53.3	54.9	54.6	**58.4**
	CW$_\infty$	48.6	48.1	**48.8**	43.0	45.0	44.8	**47.1**	52.3	**52.6**	52.2	49.7	49.5	49.6	**52.0**
	AP	48.7	49.0	**52.3**	42.7	44.8	46.8	**50.0**	52.4	54.1	**57.0**	49.1	48.7	51.8	**54.1**
	AA	45.9	46.1	**47.0**	42.4	42.9	43.5	**45.1**	48.5	**49.9**	49.0	48.2	46.9	47.5	**50.8**
	Params	11.2M			1.1M				11.2M			1.1M			
ResNet	Clean	**84.2**	82.4	83.3	79.4	**83.4**	80.9	82.7	**93.7**	93.1	92.6	90.4	**93.8**	90.5	93.3
	FGSM	58.1	57.9	**60.3**	55.0	56.7	56.4	**58.4**	75.3	81.3	**81.7**	68.9	72.0	70.6	**74.4**
	PGD	53.3	54.4	**56.4**	51.5	50.9	52.5	**53.0**	59.6	63.0	**65.0**	58.8	57.7	59.2	**60.6**
	CW$_\infty$	52.4	52.2	**52.8**	48.9	49.2	49.6	**56.1**	56.7	**59.3**	58.6	52.7	52.6	55.1	**69.8**
	AP	51.8	53.2	**55.1**	48.2	48.8	51.8	51.6	54.4	55.4	**58.3**	51.8	51.2	56.2	**58.9**
	AA	49.4	50.6	**51.2**	47.8	46.9	47.6	**48.2**	**50.5**	49.4	49.4	51.6	48.7	51.0	**53.2**
	Params	36.5M			3.7M				36.5M			3.7M			
WRN	Clean	**87.6**	86.9	87.2	85.6	87.4	87.4	**88.0**	**94.0**	92.9	92.5	91.1	**94.0**	91.0	93.8
	FGSM	61.7	61.6	**63.6**	60.3	59.8	61.6	**61.9**	73.6	74.0	**76.9**	68.9	70.6	69.0	**74.9**
	PGD	55.2	56.3	**57.6**	53.3	52.3	54.0	**55.2**	59.9	60.9	**64.5**	59.4	58.6	60.3	**61.2**
	CW$_\infty$	55.1	**56.3**	55.7	54.0	52.6	**54.4**	54.4	57.5	57.7	**57.8**	54.8	54.7	56.4	**57.5**
	AP	53.9	55.1	**55.5**	51.8	52.1	52.5	**53.2**	56.0	57.7	**60.4**	53.5	54.3	58.1	**59.6**
	AA	52.7	**53.9**	53.0	48.2	49.4	50.8	**51.5**	52.7	**54.1**	53.3	51.2	52.0	54.4	**55.1**

Table 1. Comparing adversarial robustness by pruning ratio 90% (Sparsity) on low dimensional datasets: CIFAR-10 and SVHN trained with VGG-16, ResNet-18, and WideResNet-28-10. The bold expression denotes the best performance within the same sparsity. In addition, *Params* indicates the number of non-zero model parameters.

Model	Method	Clean	FGSM	PGD	CW$_\infty$	AP	AA
VGG	LWM	44.9	20.1	17.5	12.4	12.4	11.6
	HYDRA	46.1	20.6	17.8	12.8	14.8	11.8
	MAD	**47.0**	**21.5**	**18.9**	**14.4**	**15.0**	**12.7**
ResNet	LWM	47.4	22.7	20.6	15.4	15.3	13.8
	HYDRA	49.7	23.9	21.0	15.6	18.0	14.2
	MAD	**50.7**	**25.4**	**21.9**	**17.3**	**18.5**	**15.6**
WRN	LWM	52.5	24.7	20.6	16.7	16.7	12.2
	HYDRA	53.5	24.8	21.0	16.4	17.3	14.5
	MAD	**55.9**	**27.6**	**23.4**	**19.8**	**19.8**	**17.2**

Table 2. Comparing adversarial robustness by pruning ratio 90% (Sparsity) on a larger dataset: Tiny-ImageNet. All descriptions in this table are same as those of Tab. 1.

Sparsity	Method	Clean	FGSM	PGD	CW$_\infty$	AP	AA
90%	Random	70.5	46.7	43.5	41.2	42.7	39.7
	$\Delta\mathcal{L}_{MAD} \uparrow$	69.6	46.1	43.1	40.1	41.7	38.7
	$\Delta\mathcal{L}_{MAD} \downarrow$	**81.4**	**57.0**	**51.8**	**47.1**	**50.0**	**45.1**
99%	Random	46.3	35.6	31.3	30.1	31.6	28.4
	$\Delta\mathcal{L}_{MAD} \uparrow$	53.2	36.3	33.9	31.2	33.2	30.1
	$\Delta\mathcal{L}_{MAD} \downarrow$	**61.5**	**43.0**	**41.0**	**60.0**	**39.8**	**35.0**

Table 3. Ablation study results for the comparison of adversarial robustness in three different cases on pruning ratio 90% and 99%. The validation is experimented on CIFAR-10 for VGG-16.

performance degradation. If it is correct, then we can exhibit adversarial saliency is operated well empirically in adversarial settings. Tab. 3 shows the comparison of the robustness with three different cases. First one is randomly eliminating them (Random), and second is removing them in order of high adversarial saliency ($\Delta\mathcal{L}_{MAD} \uparrow$). Lastly, it is the case of MAD ($\Delta\mathcal{L}_{MAD} \downarrow$). The result in Tab. 3 confirms that pruning model parameters in order of low adversarial saliency actually improves the adversarial robustness, and its reverse order pruning significantly loses the robustness. Furthermore, we wonder how MAD performs when model parameters are extremely pruned to 99%. Thereby, comparing MAD with the baselines, we validate the robustness across pruning ratio from 0% (full parameters) to 99% (extremely few parameters). According to Fig. 2, we observe that MAD shows better robustness than the baselines as the pruning ratio increases.

In Appendix F, we additionally experiment on Block-wise K-FAC to validate where the adversarial improvement of MAD derives from. Also, we discuss the violation of standard baselines (OBD/OBS) in adversarial settings.

4.3. Analyzing Adversarial Saliency

After we obtain an adversarially pruned model using MAD, what is our next question is *"How is adversarial saliency distributed according to certain model parameters?"* To understand and track such sensitivity for adversarial examples, we plot adversarial saliency $\Delta\mathcal{L}_{MAD}$ from the model parameters of initial layer to the last layer in Fig. 1. As in the figure, we observe the most salient parameters peak in the initial layers, sometimes referred as *stem*. In standard training, it is well known that stem plays an important role to capture local features such as colours and edges, which will be integrated to recognize global objects [38].

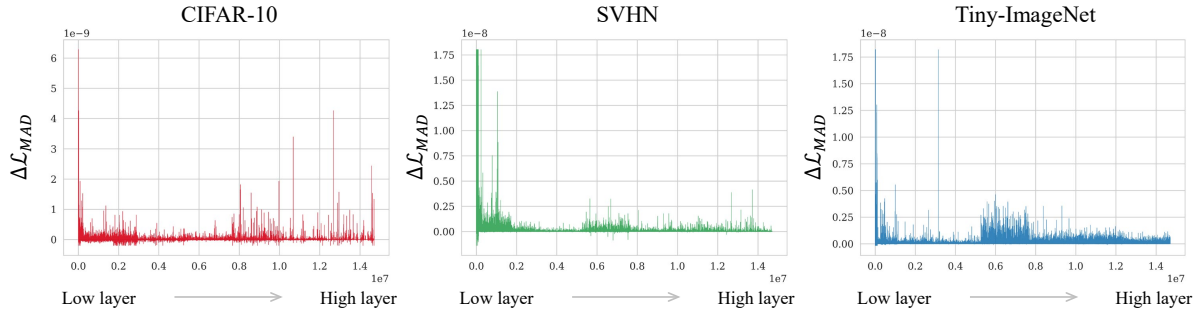Based on the perspective of adversarial training consid-

Figure 1. The shape of adversarial saliency to the whole model parameters in VGG-16 for CIFAR-10, SVHN, and Tiny-ImageNet.
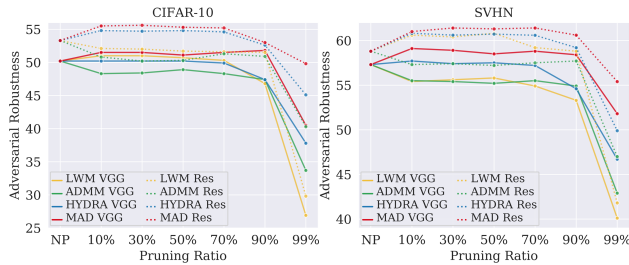

Figure 2. Ablation study results of adversarial robustness across varying pruning ratio. The experiments are conducted on CIFAR-10 and SVHN for VGG-16 and ResNet-18. Note that NP indicates AT with pruning ratio of 0%, "Non-pruned".

ered as ultimate data augmentation [51], adversarial perturbation set is regarded as an invariant factor that the robust network should be independent of. From our pruning method, we observe that the model parameters of the stem layer are highly fragile in the parameter space, and they easily affect adversarial loss the most. Concurrently, it also indicates that training stem parameters invariant to the adversarial perturbation set is another decisive key of pruning approaches in adversarially trained networks and enhancing adversarial robustness.

4.4. Understanding Compressed Features

We have demonstrated the effectiveness of MAD in the model compression while retaining adversarial robustness. Then, we want to know if the compressed features can be identified in the intermediate feature space. Recently, several works [8, 23] have introduced a way of understanding adversarial feature representation using direct feature visualization methods [35, 36]. We adopt Olah *et al.* [36] to analyze the feature representation of our pruned model and compare them with non-pruned model (AT) in Fig. 3.

As seen in this figure, we have observed that MAD can retain the semantic information of the intermediate feature representation, even with the high pruning ratio. Compared with the visualization results of the non-pruned model, MAD seemingly loses some feature representation, but still shows the recognizable quality in the semantic information of target objects regardless of the high pruning ratio.

Figure 3. Feature visualization results of Tiny-ImageNet that include semantic information for target objects. We compare MAD with non-pruned model (AT) for VGG-16. We set the pruning ratio of MAD as 90% and illustrate successfully defended cases.

5. Discussion and Conclusion

Discussion. Adversarial attacks can potentially cause negative impacts on various DNN applications due to high computation and its fragility. By pruning model parameters without weakening adversarial robustness, our work contributes important societal impacts in this research area. Furthermore, in our promising observation that model parameters of initial layer are highly sensitive to adversarial loss, we hope to progress in another future direction of utilizing such property to enhance adversarial robustness.

Conclusion. To achieve adversarial robustness and model compression concurrently, we propose a novel adversarial pruning method, Masking Adversarial Damage (MAD). By exploiting second-order information with mask optimization and Block-wise K-FAC, we can precisely estimate adversarial saliency of network parameters. Through extensive validations, we corroborate that pruning network parameters in order of low adversarial saliency retains adversarial robustness while alleviating performance degradation compared with previous adversarial pruning methods.

References

[1] Shun-Ichi Amari. Natural gradient works efficiently in learning. *Neural computation*, 10(2):251–276, 1998. 5

[2] Giovanni Apruzzese, Michele Colajanni, Luca Ferretti, and Mirco Marchetti. Addressing adversarial attacks against security systems based on machine learning. In *International Conference on Cyber Conflict (CyCon)*, volume 900, pages 1–18. IEEE, 2019. 1

[3] Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, pages 274–283, 2018. 1

[4] 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 Computer Society, 2017. 1, 6

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision (ECCV)*, pages 801–818, 2018. 1

[6] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning (ICML)*, 2020. 6

[7] Xin Dong, Shangyu Chen, and Sinno Jialin Pan. Learning to prune deep neural networks via layer-wise optimal brain surgeon. *arXiv preprint arXiv:1705.07565*, 2017. 3, 5

[8] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations. *arXiv preprint arXiv:1906.00945*, 2019. 8

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015. 1, 2, 6

[10] Alex Graves. Practical variational inference for neural networks. *Neural Information Processing Systems (NeurIPS)*, 24, 2011. 5

[11] Roger Grosse and James Martens. A kronecker-factored approximate fisher matrix for convolution layers. In *International Conference on Machine Learning (ICML)*, pages 573–582. PMLR, 2016. 5

[12] Shupeng Gui, Haotao N Wang, Haichuan Yang, Chen Yu, Zhangyang Wang, and Ji Liu. Model compression with adversarial robustness: A unified optimization framework. *Neural Information Processing Systems (NeurIPS)*, 32:1285–1296, 2019. 1, 3

[13] Yiwen Guo, Chao Zhang, Changshui Zhang, and Yurong Chen. Sparse dnns with improved adversarial robustness. *Neural Information Processing Systems (NeurIPS)*, 31:242–251, 2018. 3

[14] 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. 1, 3

[15] Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural networks. *arXiv preprint arXiv:1506.02626*, 2015. 1, 3

[16] Babak Hassibi and David G Stork. *Second order derivatives for network pruning: Optimal brain surgeon*. Morgan Kaufmann, 1993. 5

[17] Babak Hassibi, David G Stork, and Gregory J Wolff. Optimal brain surgeon and general network pruning. In *IEEE International Conference on Neural Networks*, pages 293–299. IEEE, 1993. 2, 3, 4

[18] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017. 1

[19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6

[21] Yihui He, Xiangyu Zhang, and Jian Sun. Channel pruning for accelerating very deep neural networks. In *International Conference on Computer Vision (ICCV)*, pages 1389–1397, 2017. 3

[22] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *Neural Information Processing Systems (NeurIPS)*, volume 32, 2019. 1

[23] Junho Kim, Byung-Kwan Lee, and Yong Man Ro. Distilling robust and non-robust features in adversarial examples by information bottleneck. In *Neural Information Processing Systems (NeurIPS)*, 2021. 1, 6, 8

[24] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[25] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 6

[26] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems (NeurIPS)*, volume 25, 2012. 1

[27] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. 2017. 1, 2

[28] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7, 2015. 6

[29] Yann LeCun, John S Denker, and Sara A Solla. Optimal brain damage. In *Neural Information Processing Systems (NeurIPS)*, pages 598–605, 1990. 2, 3, 4

[30] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017. 1

[31] Divyam Madaan, Jinwoo Shin, and Sung Ju Hwang. Adversarial neural pruning with latent vulnerability suppression.

In *International Conference on Machine Learning (ICML)*, pages 6575–6585. PMLR, 2020. 1, 3

[32] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018. 1, 2, 6

[33] James Martens and Roger Grosse. Optimizing neural networks with kronecker-factored approximate curvature. In *International Conference on Machine Learning (ICML)*, pages 2408–2417. PMLR, 2015. 5

[34] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011. 6

[35] Anh Nguyen, Jason Yosinski, and Jeff Clune. Understanding neural networks via feature visualization: A survey. In *Explainable AI: interpreting, explaining and visualizing deep learning*, pages 55–76. Springer, 2019. 8

[36] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2(11):e7, 2017. 8

[37] Ozan Özdenizci and Robert Legenstein. Training adversarially robust sparse networks via bayesian connectivity sampling. In *International Conference on Machine Learning (ICML)*, pages 8314–8324. PMLR, 2021. 1

[38] Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. Standalone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019. 7

[39] Vivek Ramanujan, Mitchell Wortsman, Aniruddha Kembhavi, Ali Farhadi, and Mohammad Rastegari. What's hidden in a randomly weighted neural network? In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11893–11902, 2020. 1, 3

[40] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems (NeurIPS)*, volume 28. Curran Associates, Inc., 2015. 1

[41] Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In Hal Daumé III and Aarti Singh, editors, *International Conference on Machine Learning (ICML)*, volume 119, pages 8093–8104. PMLR, 13–18 Jul 2020. 6

[42] Hippolyt Ritter, Aleksandar Botev, and David Barber. A scalable laplace approximation for neural networks. In *International Conference on Learning Representations (ICLR)*, volume 6. International Conference on Representation Learning, 2018. 5

[43] Sebastian Ruder. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*, 2016. 6

[44] Y. E. Sagduyu, Y. Shi, and T. Erpek. Iot network security from the perspective of adversarial deep learning. In *International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9, 2019. 1

[45] Vikash Sehwag, Shiqi Wang, Prateek Mittal, and Suman Jana. Towards compact and robust deep neural networks. *CoRR*, abs/1906.06110, 2019. 1, 3, 6

[46] Vikash Sehwag, Shiqi Wang, Prateek Mittal, and Suman Jana. Hydra: Pruning adversarially robust neural networks. *Neural Information Processing Systems (NeurIPS)*, 2020. 1, 3, 6

[47] 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. In A. Talwalkar, V. Smith, and M. Zaharia, editors, *Proceedings of Machine Learning and Systems*, volume 1, pages 230–240, 2019. 1

[48] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 6

[49] Sidak Pal Singh and Dan Alistarh. Woodfisher: Efficient second-order approximation for neural network compression. *Neural Information Processing Systems (NeurIPS)*, 33:18098–18109, 2020. 2, 3, 5

[50] 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 (ICLR)*, 2014. 1, 2

[51] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations (ICLR)*, 2019. 8

[52] Manoj-Rohit Vemparala, Nael Fasfous, Alexander Frickenstein, Sreetama Sarkar, Qi Zhao, Sabine Kuhn, Lukas Frickenstein, Anmol Singh, Christian Unger, Naveen-Shankar Nagaraja, et al. Adversarial robust model compression using in-train pruning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 66–75, 2021. 1

[53] Chaoqi Wang, Roger Grosse, Sanja Fidler, and Guodong Zhang. Eigendamage: Structured pruning in the kronecker-factored eigenbasis. In *International Conference on Machine Learning (ICML)*, pages 6566–6575. PMLR, 2019. 2, 3, 5

[54] Luyu Wang, Gavin Weiguang Ding, Ruitong Huang, Yanshuai Cao, and Yik Chau Lui. Adversarial robustness of pruned neural networks. 2018. 3

[55] Xianmin Wang, Jing Li, Xiaohui Kuang, Yu an Tan, and Jin Li. The security of machine learning in an adversarial setting: A survey. *Journal of Parallel and Distributed Computing*, 130:12 – 23, 2019. 1

[56] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations (ICLR)*, 2020. 1, 6

[57] Wei Wen, Chunpeng Wu, Yandan Wang, Yiran Chen, and Hai Li. Learning structured sparsity in deep neural networks. *Neural Information Processing Systems (NeurIPS)*, 29:2074–2082, 2016. 3

[58] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations (ICLR)*, 2020. 6

[59] 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 *International Conference on Computer Vision (ICCV)*, pages 111–120, 2019. 1, 3, 6

[60] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference (BMVC)*, 2016. 6

[61] Guodong Zhang, Shengyang Sun, David Duvenaud, and Roger Grosse. Noisy natural gradient as variational inference. In *International Conference on Machine Learning (ICML)*, pages 5852–5861. PMLR, 2018. 5

[62] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *International Conference on Machine Learning (ICML)*, volume 97, pages 7472–7482, 09–15 Jun 2019. 1, 6

[63] Tianyun Zhang, Shaokai Ye, Kaiqi Zhang, Jian Tang, Wujie Wen, Makan Fardad, and Yanzhi Wang. A systematic dnn weight pruning framework using alternating direction method of multipliers. In *European Conference on Computer Vision (ECCV)*, pages 184–199, 2018. 1, 3