# More or Less (MoL): Defending against Multiple Perturbation Attacks on Deep Neural Networks through Model Ensemble and Compression

Hao Cheng[1], Kaidi Xu[2], Zhengang Li[1], Pu Zhao[1], Chenan Wang[1], Xue Lin[1],
Bhavya Kailkhura[4], and Ryan Goldhahn[4]
[1]Northeastern University,[2]Drexel University,[3]Lawrence Livermore National Laboratory
{cheng.hao,li.zhen,zhao.pu,wang.chena,xue.lin}@northeastern.edu,kx46@drexel.edu,
{kailkhura1, goldhahn1}@llnl.gov

## Abstract

*Deep neural networks (DNNs) have been adopted in many application domains due to their superior performance. However, their susceptibility under test-time adversarial perturbations and out-of-distribution shifts has attracted extensive research efforts. The adversarial training provides an effective defense method withstanding evolving attacking methods. However, DNNs obtained by adversarial training are usually robust to only a single type of adversarial perturbation that they are trained with. To tackle this problem, improvements have been made to incorporate multiple perturbation types into adversarial training process, but with limited flexibility in terms of perturbation types. This work investigates the design problem of deep learning (DL) systems robust to multiple perturbation attacks. To maximize flexibility, we adopt the model ensemble approach, where an ensemble of expert models dealing with various perturbation types are integrated through a trainable aggregator module. Expert models are obtained in parallel through adversarial training, targeting at respective perturbation types. Then, the aggregator module is (adversarially) trained together with fine-tuning of expert models, addressing the obfuscated gradients issue in adversarial robustness. Furthermore, in order to practically implement the robust ensemble model onto edge devices, the model compression approach is leveraged to reduce the ensemble model size. By exploring the most suitable model compression scheme, we significantly reduce the overall model size without compromising robustness. Proposed* More or Less (MoL) *defense outperforms state-of-the-art defenses against multiple perturbations.*

## 1. Introduction

Deep learning (DL) has proliferated in many application domains [21, 43, 23, 10]. However, it has been noticed that deep neural networks (DNNs) are susceptible to adversarial perturbations (or examples) [44, 3, 46], where adversaries add sophisticatedly crafted perturbations onto clean images to produce adversarial examples. Adversarial perturbations may incur potential safety hazard in a range of application domains, particularly, self-driving or ADAS (Advanced Driver Assistant Systems) application. A major component of the self-driving or ADAS is to let DL system automate the vehicle control in the real-world road conditions. Undoubtedly, if the adversarial attack is applied to the traffic sign classification or obstacle detection system, it may have serious consequences [8, 42, 34, 4, 48]. From the attack perspective, various attack algorithms have been developed such as FGSM [17], C&W [5], EAD [7], PGD [31], StrAttack [47], Universal Attack [51], etc. Those attack algorithms usually set an $\ell_p$-norm constraint on the added adversarial perturbations to deceive human perception, besides misleading the DNN predictions. Besides, naturally occurring out-of-distribution (OoD) shifts such as weather [36, 9] and geometric transformations [14, 15] can also significantly degrade the accuracy of DNNs, which should be accounted for when designing robust DL systems.

From the defense side, several seemingly effective defenses have been proposed, such as defensive distillation [38], image denoising [18], stochastic activation pruning [11], etc. However, those methods are subject to the obfuscated gradients [2], a gradient masking phenomenon that leads to a false sense of security under adversarial perturbation attacks. On the other hand, adversarial training [17, 31] was proposed that is resilient to the obfuscated gradients issue, and therefore provides the notion of security against the adversarial perturbation attacks through a min-max optimization. Adversarial training can be considered as an augmented DNN training process, where a DNN model is trained with an adversarial perturbation type until the model learns to classify those adversarial examples correctly.

Although adversarial training provides a principled de-

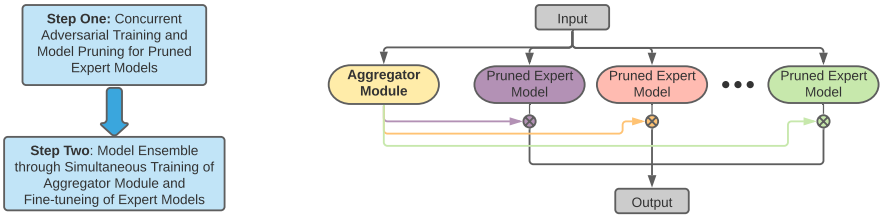



Figure 1. (a) Two-step procedure of the proposed *MoL* defense framework. (b) *MoL* system diagram.

fense, DNNs obtained from it are usually only robust to a single type of adversarial perturbation that they are trained with. To tackle this problem, the adversarial training has been improved such that DNNs are trained with a union of multiple types of $\ell_p$ norm constrained adversarial perturbations. For example, Tramer and Boneh [45] proposed adversarial training on all types of adversarial perturbations (i.e., the AVG strategy) and on only the worst case type (i.e., the MAX strategy). Maini et al. [33] extended over the AVG and MAX strategies by proposing Multi Steepest Descent (MSD) for adversarial training, which incorporates different gradient-based perturbation models into a single unified adversary to directly solve the inner maximization. [33].

This paper investigates the design problem of robust DL systems against multiple perturbation attacks. Existing works based on adversarial training such as [45, 33] mainly focus on adversarial perturbations. Differently in this work, we further incorporate the naturally occurring OoD perturbations such as weather transformations [36, 9] and geometric transformations [14, 15] in our defense method. Note that OoD perturbations are ubiquitous and may significantly degrade the DL performance in real-world. Specifically, for the self-driving and ADAS applications, the performance of various classification and detection systems will face serious degradation when encountering bad weather or geometric shifts. Therefore, we propose to design robust DL systems with a broad coverage of both adversarial and natural perturbations. However, the existing adversarial training based methods [45, 33] lack the flexibility to incorporate various perturbation types, especially the natural perturbations. Because unlike adversarial perturbations, the natural perturbations are DNN model-agnostic, and therefore it cannot achieve satisfying robustness performance by directly integrating natural perturbations within the gradient-based adversarial perturbation models in the inner maximization of adversarial training. Therefore, we adopt the model ensemble approach to achieve broad coverage of adversarial and natural perturbations with desirable performance. Additionally, although the model ensemble approach has been used for DNN robustness [37, 49, 24], they only defend against transferred attacks or different attack algorithms, but for the same perturbation type.

To achieve a broad coverage of perturbation types, we propose to integrate an ensemble of expert models through a trainable aggregator module. Each expert model is obtained individually through adversarial training to deal with a particular perturbation type, which is followed by training of the aggregator module together with fine-tuning of the expert models. The aggregator training and expert fine-tuning process uses adversarial training to essentially eliminate the obfuscated gradients issue. A shortcoming of the model ensemble approach is the overlarge model size, making it impractical to be implemented onto resource limited edge devices for efficient inference execution. To address this shortcoming, we leverage the DL model compression techniques, specifically, the model pruning approach. We propose to explore various model pruning schemes such as irregular pruning [20, 19], filter pruning [28, 22], and block-based pruning [12, 29] to reduce the storage and computation requirements of the whole ensemble model without compromising the robustness performance.

Our proposed More or Less (*MoL*) defense framework is shown in Figure 1 (a). "More" here denotes the adopted model ensemble approach that designs robust DL systems against a variety of perturbation types, and "Less" denotes the utilized model compression method that reduces the overall ensemble model size. The *MoL* defense framework uses a two-step procedure, where step one is to perform the concurrent adversarial training and model pruning to obtain pruned expert models each targeting at a particular perturbation type, and step two is to integrate the pruned expert models through simultaneous training of the aggregator module and fine-tuning of expert models. To further decrease the overall model size and in turn enable efficient edge execution, the aggregator module also adopts a DNN architecture with reduced size. To eliminate obfuscated gradients, step two uses well designed adversarial training to accommodate both adversarial and natural perturbations.

Figure 1 (b) shows the overall *MoL* system. The input to the system is fed to each expert model as well as the aggregator module. The aggregator module adaptively assigns weights to the expert models on the fly, and the outputs from expert models are aggregated as the final output of the system. Our main contributions can be summarized as follows:

- We propose a model ensemble based defense against multiple perturbation attacks. Different from the adversarial training based works, our approach achieves broad coverage for both adversarial and natural perturbations with desirable robustness.
- It is the first work exploring model compression to significantly reduce the size of robust ensemble model, without compromising accuracy.
- Comparing with state-of-the-art defenses, our proposed *MoL* defense achieves the highest robustness performance. Specifically, using expert models at 1/8 of the original size (with an overall model size the same as the baselines) when defending against both adversarial and natural perturbations on CIFAR-10, our *MoL* increases the clean accuracy, $\ell_1$ , $\ell_2$ & $\ell_\infty$ adversarial accuracy, natural perturbation accuracy and all attack accuracy by up to 12.2%, 15.6%, 19.7%, 17.2%, 7.2%, and 7.8%, respectively.

# 2. Related Work

## 2.1. Adversarial Training against Multiple Perturbation Attacks

Adversarial training [31] uses an inner maximization problem presenting the first-order adversary embedded within the outer minimization on training loss. However, such adversarial training process only provides robustness to a single type of adversarial perturbation that is presented by its inner maximization. To generalize onto multiple adversarial perturbation types, improvements have been made onto the inner maximization with a union of multiple types of $\ell_p$ norm constrained adversarial perturbations. Tramer and Boneh [45] proposed the adversarial training for multiple perturbations. Specifically, they devised the AVG strategy that trains on all types of perturbations and the MAX strategy that for each input trains on the strongest perturbation among all the types. Maini, Wong, and Kolter [33] proposed the Multi Steepest Descent (MSD) algorithm that incorporates the different perturbation types within each step of projected steepest descent, rather than generating adversarial perturbations in different types with separate projected gradient descent adversaries. This paper compares with these two works, as they represent state-of-the-arts for adversarial robustness against multiple perturbation attacks.

Furthermore, Classify Then Predict (CTP) [32] uses a two-stage pipeline to improve the robustness against the union of multiple perturbation types. Stutz, et. al. [41] proposed the Confidence-Calibrated Adversarial Training (CCAT) method by biasing the model towards low confidence predictions on adversarial examples. Adversarial Distributional Training (ADT) [13] was proposed for robustness against unseen attacks. Attribute-Guided Adversarial Training (AGAT) [16] leverage generative models conditioned on perturbation attributes.

Existing adversarial training based works mainly focus on adversarial perturbations. This work targets a broad coverage of both adversarial and natural perturbations, which are ubiquitous and can degrade DNN performance as adversarial perturbations do. But direct integration of natural perturbations with adversarial perturbations into adversarial training process cannot achieve satisfying robustness performance, as demonstrated in our experiments.

## 2.2. Model Ensemble Learning for Adversarial Robustness

Ensemble learning with multiple models has also been applied for improving adversarial robustness. Adaptive Diversity Promoting (ADP) regularizer [37] was proposed to encourage diversity, leading to robustness to different attack algorithms including FGSM [17], PGD [31], C&W [5], EAD [7], etc. But, this work still targets a single perturbation type. In [49] an ensemble model to adopt diversity among sub-models for robustness to transferred attacks was proposed. Kariyappa and Qureshi [24] showed that an ensemble of models with misaligned loss gradients can provide an effective defense against transfer-based attacks and proposed Diversity Training, a method to train an ensemble of models with uncorrelated loss functions.

The current model ensemble approaches for adversarial robustness can defend against different attack algorithms or transferred attacks, but with the same perturbation type. We propose to use model ensemble learning defending against multiple perturbation types, with adversarial training exploited in individual expert model training and in model ensemble process, to essentially eliminate obfuscated gradients. Furthermore, we propose to use model compression i.e., model pruning to address the shortcoming of model ensemble learning, namely, the overlarge model size issue.

# 3. Preliminaries

## 3.1. Perturbation Attacks on Deep Learning

This work defends against two major categories of perturbation attacks i.e., $\ell_p$ adversarial perturbations and natural perturbations.

### 3.1.1 Adversarial Perturbations

An adversarial perturbation attack is to find adversarial perturbation $\boldsymbol{\delta}$ with minimized $\ell_p$ norm (denoted by $\|\cdot\|_p$) for a clean input $\boldsymbol{x}$ (with its correct one-hot label $\boldsymbol{y}$), such that a DNN model $f_{\boldsymbol{\theta}}(\cdot)$ (with parameters $\boldsymbol{\theta}$) predicts the adversarial example $\boldsymbol{x} + \boldsymbol{\delta}$ differently from its correct label. Therefore, the problem of generating an $\ell_p$ adversarial per-

turbation for a clean input $\boldsymbol{x}$ can be cast as:

$$\begin{aligned} \min \quad & \|\boldsymbol{\delta}\|_p \\ s.t. \quad & \arg\max(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta})) \neq \arg\max(\boldsymbol{y}). \\ & \boldsymbol{x}+\boldsymbol{\delta} \in [0,1]^n \end{aligned} \tag{1}$$

#### 3.1.2 Natural Perturbations

Different from adversarial perturbations, the naturally occurring perturbations indistinctly degrade the DNN accuracy performance, i.e., they are DNN model-agnostic. We introduce three types of naturally occurring perturbations, i.e., Fog, Snow and Rotation here. For Fog perturbation, there are two hyperparameters to control the strength i.e., thickness factor $t$ and atmospheric light factor $light$. The Fog generation process can be described as:

$$\boldsymbol{x}_{fog} = \frac{\boldsymbol{x}}{(1+\mathrm{e}^{-t})} + (1 - \frac{1}{(1+\mathrm{e}^{-t})}) \cdot light. \tag{2}$$

For Snow perturbation, we define function $T$ that can convert RGB image to HLS image, and $T^{-1}$ vice versa. The strength of snow perturbation is determined by the factor of $darkness$ and a random noise $\alpha \sim U(0, 255/2)$:

$$\begin{aligned} \boldsymbol{x}_{HLS_i} &= \begin{cases} T(\boldsymbol{x})_i & T(\boldsymbol{x})_i < \alpha, \\ T(\boldsymbol{x})_i \cdot darkness & otherwise \end{cases} \\ \boldsymbol{x}_{snow} &= T^{-1}(\boldsymbol{x}_{HLS}). \end{aligned} \tag{3}$$

For Rotation perturbation, we simply utilize affine transformation to rotate the input $\boldsymbol{x}$ with a random degree in $[-180°, +180°]$.

### 3.2. Adversarial Training for Expert Models

We propose to leverage model ensemble learning, where each expert model targets a particular perturbation type. We need three categories of expert models i.e., (i) a normally trained expert model to deal with clean inputs, (ii) adversarially trained expert models to deal with different $\ell_p$ adversarial perturbation types, and (iii) adversarially trained expert models to deal with different natural perturbation types.

#### 3.2.1 Expert for Clean Inputs

The normal training is used to obtain the expert model targeting at clean inputs as follows:

$$\min_{\boldsymbol{\theta}} \mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}), \boldsymbol{y}), \tag{4}$$

where $\mathcal{L}(\cdot)$ is the loss function.

#### 3.2.2 Expert for Adversarial Perturbations

Adversarial training uses min-max optimization to defend against an $\ell_p$ norm constrained perturbation:

$$\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\delta} \in \boldsymbol{\Delta}_p} \mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), \boldsymbol{y}), \tag{5}$$

where $\boldsymbol{\delta}$ denotes the adversarial perturbation bounded by the $\ell_p$ norm ball $\boldsymbol{\Delta}_p$. The inner maximization problem can be solved by projected gradient descent (PGD).

#### 3.2.3 Expert for Natural Perturbations

The naturally occurring perturbations are DNN model-agnostic, therefore we use a slightly modified adversarial training process to obtain those expert models. We first transform the inputs according to a natural perturbation type such as Fog, Snow, or Rotation as explained in Section 3.1.2 to obtain $\boldsymbol{x}_{\text{nat}}$. Then we minimize the following loss to obtain an expert model robust to a type of natural perturbation.

$$\min_{\boldsymbol{\theta}} \mathcal{L}(f_{\boldsymbol{\theta}}(x_{\text{nat}}), y). \tag{6}$$

### 3.3. Concurrent Adversarial Training and Model Pruning for Pruned Expert Models

We propose to leverage the DNN model pruning technique to reduce the size of the expert models, such that the whole ensemble model can be practically implemented on edge devices for efficient inference execution.

#### 3.3.1 Pruning Schemes

We explore three representative pruning schemes i.e., irregular pruning [20, 19], filter pruning [28, 22], and block-based pruning [12, 29]. Block-based pruning is of particular importance here, since it can achieve negligible accuracy loss as the irregular pruning does, but are far more friendly for inference execution on computing devices. Details of the pruning schemes are presented in Appendix A.

#### 3.3.2 Concurrent Adversarial Training and Model Pruning

We adopt the concurrent adversarial training and model pruning approach [50] to obtain pruned expert models with reduced size, targeting at $\ell_p$ adversarial perturbation types. We improve the concurrent adversarial training and model pruning by incorporating the novel block-based pruning scheme as a sweet spot across the accuracy and inference speed performance.

Generally, the concurrent adversarial training and model pruning can be cast as the following min-max optimization process

$$\min_{\boldsymbol{\theta}} \left[ \max_{\boldsymbol{\delta} \in \boldsymbol{\Delta}_p} \mathcal{L}(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}), \boldsymbol{y}) + g(\boldsymbol{\theta}) \right], \tag{7}$$

where $g(\boldsymbol{\theta}) = \sum_l g_l(\boldsymbol{\theta}_l)$ and $g_l(\boldsymbol{\theta}_l)$ denotes whether the weights $\boldsymbol{\theta}_l$ of the $l$-th layer satisfy the particular pruning scheme requirement

$$g_l(\boldsymbol{\theta}_l) = \begin{cases} 0 & if \ \ \boldsymbol{\theta}_l \in S_l, \\ +\infty & otherwise. \end{cases} \tag{8}$$

$S_l$ is the pruning scheme requirement for the $l$-th layer.

We explore three pruning schemes as described in Section 3.3.1, namely, irregular pruning, filter pruning, and block-based pruning. Here, we provide an example of block-based pruning [25], for which the pruning scheme requirement can be formulated as

$$S_l = \{\boldsymbol{\theta}_l \mid ||\boldsymbol{\theta}_l||_{N0}^b \leq \alpha_l\}, \tag{9}$$

where $||\boldsymbol{\theta}_l||_{N0}^b$ denotes the number of remaining non-zero blocks in the $l$-th layer, and $\alpha_l$ is the pruning threshold.

For pruning expert models dealing with clean inputs and natural perturbations, the following losses are minimized,

$$\min_{\boldsymbol{\theta}} \left[\mathcal{L}(f_{\boldsymbol{\theta}}(x), y) + g(\boldsymbol{\theta})\right], \tag{10}$$

$$\min_{\boldsymbol{\theta}} \left[\mathcal{L}(f_{\boldsymbol{\theta}}(x_{\text{nat}}), y) + g(\boldsymbol{\theta})\right]. \tag{11}$$

To solve the above two optimization problems satisfying various pruning scheme requirements, we adopt ADMM (alternating direction method of multipliers) based pruning algorithms [30].

# 4. Proposed MoL Defense Framework

## 4.1. MoL Defense System

Inspired by Expert Gate [1] in lifelong learning, we propose *MoL* defense architecture as in Figure 1 (b). A trainable aggregator module is employed to adaptively assign weights to the expert models during inference. The inputs to the system are simultaneously fed to the aggregator and experts. And the outputs from experts are aggregated as the final output of the system.

We adopt the ResNet-18 [21] model for both the aggregator and experts as a reasonable choice considering both accuracy and training time. Note that the dimension of the output layer of the aggregator needs to match the number of experts in the ensemble. For the case that expert models are in the pruned size (i.e., $1/K$ of the original size), we reduce the model size of the aggregator accordingly (i.e., $1/K$ too).

We use $\boldsymbol{\phi}$ to represent the aggregator and its output can be denoted by $f_{\boldsymbol{\phi}}(\boldsymbol{x})$ with the input $\boldsymbol{x}$. Then, the output of the whole ensemble model can be expressed as

$$f(\boldsymbol{x}) = \sum_i \left[f_{\boldsymbol{\phi}}(\boldsymbol{x})\right]_i \cdot f_{\boldsymbol{\theta}_i}(x) \tag{12}$$

where $\left[f_{\boldsymbol{\phi}}(\boldsymbol{x})\right]_i$ denotes the $i$-th element of the aggregator output and $f_{\boldsymbol{\theta}_i}(x)$ is the output of the $i$-th expert.

## 4.2. Model Ensemble Learning

Our proposed *MoL* consists of two steps: (i) Concurrent adversarial training and model pruning to obtain individual expert models, each targeting at a particular perturbation type; and (ii) Model ensemble through simultaneous training of the aggregator and fine-tuning of experts. We consider seven expert models i.e., an expert to deal with clean inputs, three experts to deal with $\ell_1$, $\ell_2$, and $\ell_\infty$ adversarial perturbations, respectively, and three experts to deal with Fog, Snow, and Rotation natural perturbations, respectively. We highlight that thanks to the ADMM based method, we can concurrently perform adversarial training and model pruning to obtain the pruned experts, instead of training a robust expert first and then pruning it, to save training time and to preserve the robustness performance at pruned size.

For step (ii), the aggregator module and the last fully connected (FC) layers of experts are updated to minimize the classification loss. We highlight that (1) the other layers except for the last FC layer in an expert model are not updated to keep the expertise of the expert, and (2) the last FC layers of the expert models are fine-tuned together with training of the aggregator module to achieve a higher level of cooperation between the aggregator and experts rather than a comprise of the aggregator with the fixed experts.

Specifically, as the aggregator needs to recognize certain adversarial or natural perturbation type to assign a larger weight to the corresponding expert model, we need to feed all kinds of images (i.e., $\boldsymbol{x}_i$ for $i = 1, ..., 7$, corresponding to the clean, three adversarially perturbed and three naturally perturbed images) to the aggregator. For each type

**Algorithm 1** MoL Defense Framework

**Input**: dataset $\mathcal{D}$, aggregator module $\boldsymbol{\phi}$, expert models $\{\boldsymbol{\theta}_i\}_{i=1}^m$, number of training steps $T$.
**Output**: aggregator module $\boldsymbol{\phi}$ and expert models $\{\boldsymbol{\theta}_i\}_{i=1}^m$
# Train the expert model.
**for** $i = 1, 2, \ldots, m$ **do**
    Train the $i$-th expert $\boldsymbol{\theta}_i$ with the corresponding expertise type.
**end for**
# Train the aggregator
**for** $t = 1, 2, \ldots, T$ **do**
    Sample mini-batch data $(\boldsymbol{x}, \boldsymbol{y})$ from $\mathcal{D}$
    **for** $i = 1, 2, \ldots, m$ **do**
        Get perturbed images $x_i$ corresponding to the $i$-th perturbation type
        **for** $j = 1, 2, \ldots, m$ **do**
            obtain $\mathcal{L}_j(\boldsymbol{\phi}, \boldsymbol{\theta}_j) = \mathcal{L}([f_{\boldsymbol{\phi}}(\boldsymbol{x}_i)]_j \cdot f_{\boldsymbol{\theta}_j}(x_i), y)$
        **end for**
        # Update weights $(\boldsymbol{\phi}, \boldsymbol{\theta}_j)$
        $\mathcal{L}^*(\boldsymbol{\phi}, \boldsymbol{\theta}_j) = \max_j(\mathcal{L}_j(\boldsymbol{\phi}, \boldsymbol{\theta}_j))$
        update $\boldsymbol{\phi}, \boldsymbol{\theta}_j$ to minimize $\mathcal{L}^*(\boldsymbol{\phi}, \boldsymbol{\theta}_j)$
    **end for**
**end for**

of image $\boldsymbol{x}_i$, the objective of the aggregator is to generate a large weight for the corresponding expert model output. To achieve this, inspired by the adversarial training to optimize the worst case, we compute the weighted output of each expert model and its loss, i.e. $\mathcal{L}_j(\boldsymbol{\phi}, \boldsymbol{\theta}_j) = \mathcal{L}([f_{\boldsymbol{\phi}}(\boldsymbol{x}_i)]_j \cdot f_{\boldsymbol{\theta}_j}(x_i), y), j = 1, ..., 7$, and select the largest loss (the worst case) i.e., $\mathcal{L}^*(\boldsymbol{\phi}, \boldsymbol{\theta}_j) = \max_j(\mathcal{L}_j(\boldsymbol{\phi}, \boldsymbol{\theta}_j))$. Then we minimize the largest loss to update the aggregator and the last FC layers of expert models as shown below

$$\min_{\boldsymbol{\phi}, \boldsymbol{\theta}_j} \max_j \mathcal{L}([f_{\boldsymbol{\phi}}(\boldsymbol{x}_i)]_j \cdot f_{\boldsymbol{\theta}_j}(x_i), y). \tag{13}$$

For the aggregator training, we also need to reduce the aggregator size to obtain an overall model size similar to other works for a fair comparison. But as the aggregator training is coupled with fine-tuning the last FC layers of experts, it could be complex to perform ADMM pruning. To reduce the complexity, we simply use a model with $1/4$ or $1/8$ size of the original Resnet18 model, by reducing the number of intermediate channels in the model to $1/4$ or $1/8$. Thus the size of the obtained aggregator can be reduced without incorporating relatively complex pruning method.

We show the overall training framework in Algorithm 1. We first train the expert models (Lines 3-4). Next, we train the aggregator. Specifically, in each training batch, we generate all kinds of clean, adversarially and naturally perturbed images $\boldsymbol{x}_i$ for $i = 1, ..., 7$ (Line 9). For each $\boldsymbol{x}_i$, we compute the loss of the weighted output of each expert model (Lines 10-12) and minimize the largest loss to update the aggregator and the last FC layers of expert models (Lines 13-14). The aggregator training can be treated as an adversarial training method against multiple perturbations.

# 5. Experiments

## 5.1. Experimental Setup

### 5.1.1 Architecture and Hyperparameters of Experts

In this section, we use the image classification task to evaluate our methods. Two popular datasets are included, i.e., CIFAR-10 and Tiny-ImageNet. ResNet-18 [21] is used as our base architecture for both the aggregator module and 7 different expert networks. The training process includes two phases: expert training/pruning and aggregator training.

We need to train a clean expert, three experts robust to adversarial attacks, two weather robust experts, and an image rotation robust expert. For adversarially robust experts, we perform Projected Gradient Descent (PGD) [31] to generate three kinds of $\ell_p$ perturbations, i.e., $\ell_1$, $\ell_2$ and $\ell_\infty$. The perturbation strength value $\epsilon$ is set to be 16 ($\epsilon = 12$ in Tiny-ImageNet), 1.0, and $8/255$ for $\ell_1$, $\ell_2$, and $\ell_\infty$ experts, respectively. The numbers of PGD attack steps $k$ are 20 and 7 for CIFAR-10 and Tiny-ImageNet, respectively. For perturbation step size, we adopt different values in three $\ell_p$ experts, i.e., $2.5 * \epsilon/k$ for $\ell_1$ expert, $\epsilon/5$ for $\ell_2$ expert, and $0.01$ for $\ell_\infty$ expert. For weather robust experts, we use two kinds of natural perturbations, i.e., fog and snow, following [36]. Here, we set $t = 0.4$ and $light = 1.2$ for the fog perturbation and $darkness = 8$ for the snow perturbation. For the image rotation robust expert, we follow [15] with the rotation angle within $[-180°, +180°]$.

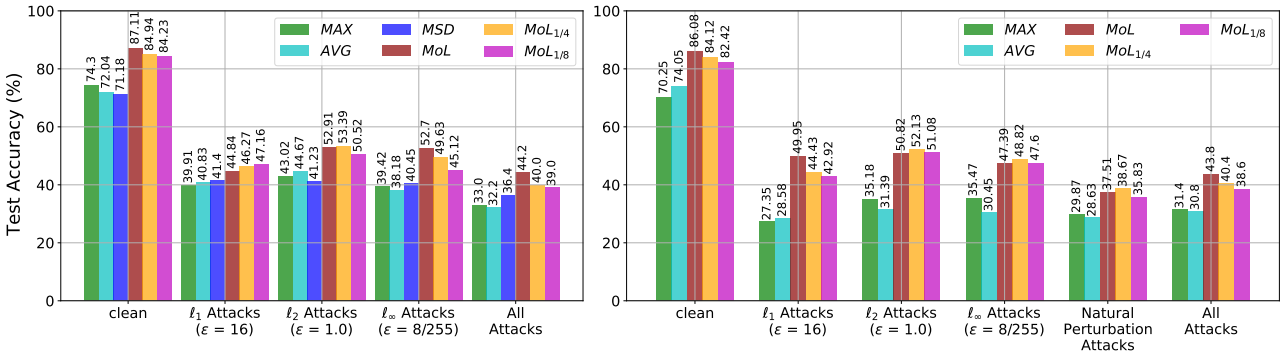
### 5.1.2 Training and Pruning Setup for Experts

For the expert training/pruning, ADMM regularization method [50] is conducted. We use SGD optimizer with a learning rate ($lr$) updating scheduler following a cosine function [27]. We set $lr = 0.01$ initially for CIFAR-10 ($lr = 0.001$ for Tiny-ImageNet) with a decay factor as $5 \times 10^{-4}$. For the pruning step, the ADMM pruning penalty factor $\rho$ is set to $0.001$ in the whole process. $200$ epochs with fixed batch size 128 are used in both individual expert ADMM pruning and masked retraining. For the pruning, we set the pruning rate (the rate between the number of non-zero elements in the original model and that of the pruned model) to $4\times$ or $8\times$ such that we can obtain a pruned expert model with $1/4$ or $1/8$ non-zero weights. We also experiment with different pruning schemes including irregular pruning [20], filter pruning [19] and block pruning [12]. For aggregator training, we also reduce the aggregator size. As discussed in Section 4.2, we train a smaller Resnet-18 as the aggregator by tuning the number of intermediate channels in the model. The number of parameters in the aggregator can be reduced to $1/4$ or $1/8$ of the original number.

### 5.1.3 Attack Scenarios for Evaluation

We validate the *MoL* under two attack scenarios: (i) mixing $\ell_p$ attack; and (ii) comprehensive attack . In the mixing $\ell_p$ attack, we only attack the ensemble with the $\ell_1$, $\ell_2$ and $\ell_\infty$ attacks. The ensemble is only composed of the clean expert, the $\ell_1$, $\ell_2$ & $\ell_\infty$ experts, and the aggregator, while we do not use other natural perturbation experts. This setting is well aligned with other baselines which focus on $\ell_p$ adversarial attacks without considering natural perturbations. In the comprehensive attack, we use both adversarial attacks and natural perturbations to attack our ensemble. The ensemble is composed of all seven experts, including clean expert, three adversarial robust experts, and three natural perturbation robust experts, with the aggregator.

To make a comprehensive study, although our adversarial experts are trained with PGD attack, in the evaluation of the ensemble, we employ various attack methods besides PGD attack to attack our *MoL*. For $\ell_1$ attacks, besides the $\ell_1$ PGD [31] attack, we also use two digital attacks Gaussian Noise and Salt&Pepper [39] as the supplements. For $\ell_2$ attacks, we validate our *MoL* system under more attack methods in additional to $\ell_2$ PGD adversary. Specifically, we




(a) Defense against only adversarial perturbations

(b) Defense against both adversarial and natural perturbations

Figure 2. Accuracy comparisons on CIFAR-10 for (a) defending against only the adversarial perturbations and (b) defending against both adversarial and natural perturbations (decomposed results are shown in Table A5). Our *MoL* system at different sizes (original size, 1/4, and 1/8 of original size) are compared with baselines including MAX [45], AVG [45], and MSD [33].

Table 1. **Test accuracy (%) of unpruned expert models on CIFAR-10 dataset under various attack methods.**

| | $M_{clean}$ | $M_{\ell_1}$ | $M_{\ell_2}$ | $M_{\ell_\infty}$ | $M_{fog}$ | $M_{snow}$ | $M_{RT}$ |
|---|---|---|---|---|---|---|---|
| clean accuracy | **95.48** | 80.91 | 84.43 | 83.33 | 72.43 | 90.95 | 88.57 |
| PGD $\ell_1$ ($\epsilon = 16$) | 2.43 | **55.90** | 56.44 | 49.84 | 0.64 | 2.98 | 1.32 |
| PGD $\ell_2$ ($\epsilon = 1.0$) | 0.00 | 45.33 | **49.71** | 32.22 | 0.00 | 0.00 | 0.00 |
| PGD $\ell_\infty$ ($\epsilon = 8/255$) | 0.00 | 33.00 | 40.11 | **50.09** | 0.00 | 0.00 | 0.00 |
| Fog ($\{t, light\} = \{0.4, 1.2\}$) | 18.09 | 14.71 | 10.08 | 10.01 | **90.01** | 27.62 | 14.42 |
| Snow ($darkness = 8$) | 30.10 | 24.31 | 25.91 | 26.62 | 17.79 | **72.88** | 15.58 |
| Rotation ($max = 180°$) | 31.24 | 31.56 | 31.36 | 31.77 | 24.98 | 30.51 | **83.13** |

use FGSM [17], DNN based attack [40], RayS attack [6], Deepfool [35] following the Foolbox implementations with $\epsilon$ the same as the $\ell_2$ PGD attack to evaluate our model. For $\ell_\infty$ attacks, similarly, we additionally adopt Foolbox implementations to generate $\ell_\infty$ FGSM [17], RayS and Deepfool to test the $\ell_\infty$ robustness of *MoL*. Besides, to further highlight the superior performance of *MoL*, we also adopt All Attack [33] to validate the robustness performance.

#### 5.1.4 Baselines

For the mixing $\ell_p$ attack scenario, we use MAX [45], AVG [45], and MSD [33] as the baselines, since they are only designed for $\ell_p$ adversaries without consideration of natural perturbations. For the comprehensive attack scenario, *MoL* is compared with MAX and AVG. MSD is not used as it can not extend to defend natural perturbations.

### 5.2. Individual Expert Model Performance

#### 5.2.1 Unpruned Expert Model Performance

Table 1 shows the performance of unpruned expert models in the original size of ResNet-18 on CIFAR-10 dataset. Each column shows the performance of one expert under various adversarial attacks or natural perturbations. For example, Column 2 shows the accuracy of the clean expert model under each perturbation type. Each row demonstrates the accuracy of one perturbation type under various expert models. For example, Row 3 shows the accuracy of all seven experts under the $\ell_1$ PGD attack. The attack hyperparameters are also listed together with each attack method, following the common realistic setting. We can observe that the accuracy under the corresponding attacks or transformations (**bold** values) are usually the highest in each column, demonstrating that the expert models can achieve certain expertise by training with the corresponding attacks or perturbations. An outlier is found on the $\ell_2$ expert model, where its accuracy under $\ell_1$ attack is a bit higher than the $\ell_1$ expert model. This can be attributed to the high correlations between $\ell_1$ and $\ell_2$ norms. The results on Tiny-ImageNet can be found in Table A1 of Appendix.

#### 5.2.2 Pruned Expert Model Performance

Next, we investigate the effect of model pruning on all experts. As shown in Table A2, we perform concurrent training and pruning to obtain pruned experts with a uniform pruning rate $4\times$ for every layer. For each attack or natural perturbation, we use three different pruning schemes, i.e., irregular pruning (I), filter pruning (F) and block pruning (B), to train and prune three corresponding models to investigate the effects of pruning schemes. Compared with unpruned models, under irregular or block pruning, we can observe that the pruned expert models exhibit negligible accuracy degradation, and some experts can even achieve a higher accuracy with pruning thank to the dedicated retraining process during the pruning. For filter pruning, we observe a more obvious accuracy degradation compared with irregular or block pruning. The results on Tiny-ImageNet can be found in Table A3 of Appendix.

Furthermore, we explore expert pruning with a larger pruning rate $8\times$ and show the results in Table A4 of Appendix. Similarly, the accuracy degradation of the pruned expert models are negligible with irregular or block pruning, though the number of non-zero parameters is reduced to $1/8$ in pruned expert models, demonstrating the advan-

tages of the concurrent training and pruning method.

In summary, block pruning obtains pruned experts with $1/4$ or $1/8$ model size while achieving competitive robustness. It builds a firmer foundation of our *MoL* framework robust to multiple perturbations with moderate overhead.

## 5.3. MoL Performance

As discussed in Section 5.1.3, besides PGD attacks, we adopt other attacks during evaluation and plot the average accuracy in Figure 2. The accuracy of each attack method is presented in Table A5 of Appendix. Here we mainly report the results on CIFAR-10. More results on Tiny-ImageNet are shown in Table A6 of Appendix. Specific performance of pruned *MoL* on edge devices are in Appendix B.

### 5.3.1 Performance under Mixing $\ell_p$ Attack Scenario

In the mixing $\ell_p$ attack scenario, the ensemble only contains the clean, $\ell_1$, $\ell_2$ & $\ell_\infty$ experts, and we use $\ell_1$, $\ell_2$, $\ell_\infty$ and all attacks [33] to evaluate the ensemble performance to achieve a fair comparison with other baselines only focusing on $\ell_p$ attacks. With block pruning, we train three ensembles corresponding to three pruning rate configurations, i.e., $1\times$, $4\times$ and $8\times$, respectively. $1\times$ pruning rate means that the experts are not pruned. $4\times$ or $8\times$ pruning rate means that the aggregator and each expert after pruning only has $1/4$ or $1/8$ non-zero parameters compared with the unpruned model. As demonstrated in Figure 2 (a), on CIFAR-10 we use MAX [45], AVG [45], and MSD [33] as the baselines, and report the performance of our three ensembles. Note that in this scenario, as we have 4 experts and one aggregator in the ensemble, $1/4$ expert size leads to about $5/4$ overall model size which is a bit larger than the model size in other baselines, and $1/8$ expert size leads to about $5/8$ overall model size which is smaller than the models in the baselines. Note that to avoid the challenge of obfuscated gradient [2], we adopt black-box attack, RayS (Table A5), demonstrating the effectiveness of the proposed method to deal with the obfuscated gradient issue.

Figure 2 (a) shows that our three ensembles can achieve higher accuracy on clean data or under $\ell_1$, $\ell_2$ and $\ell_\infty$ attacks compared with MAX, AVG, and MSD. Generally pruning more weights with a larger pruning rate can result in certain accuracy drop. However, even with $8\times$ pruning rate where each pruned expert only has $1/8$ non-zero elements compared with the unpruned expert, the performance of our ensemble is still better than all baselines, demonstrating the effectiveness of the proposed ensemble framework. In this case, the overall model size (about $5/8$) is smaller than that of other baselines and we can still achieve higher accuracy.

### 5.3.2 Performance under Comprehensive Attack

In the comprehensive attack scenario, the ensemble contains the clean expert, three adversarial attack and three natural perturbation experts. On CIFAR-10, we evaluate our *MoL* under clean data, adversarial attacks, natural perturbations and all attack [33]. Similar to the previous scenario, we train three *MoL* corresponding to three pruning rate configurations ($1\times$, $4\times$ and $8\times$). As each ensemble has 7 experts and one aggregator, $8\times$ pruning rate leads to an overall model size almost the same as the models in the baselines. We compare our ensembles with MAX [45] and AVG [45]. MSD [33] is specifically designed for $\ell_p$ adversarial attacks and not able to be extended to natural perturbations.

As shown in Figure 2 (b), our three ensembles under different pruning rates can achieve higher accuracy on clean data or under $\ell_1$, $\ell_2$, $\ell_\infty$ attacks or natural perturbations, compared with MAX and AVG. Pruning more weights generally can lead to further accuracy drop. But sometimes the ensemble with $4\times$ pruning rate can achieve slightly better accuracy than the ensemble with unpruned experts which is consistent with the phenomenon we observed in experts pruning, demonstrating there may be certain redundancy in the experts and removing the redundancy can further improve their performance. We can observe that for the ensemble with $8\times$ pruning rate, where lead to an overall ensemble model size almost the same as the model in the baselines, our method can achieve better accuracy than the baselines, demonstrating powerful of *MoL*. Note that the improvement is not marginal as the *MoL* can increase the clean accuracy, $\ell_1$, $\ell_2$ & $\ell_\infty$ adversarial accuracy, natural perturbation accuracy and all attack accuracy by up to 12.2%, 15.6%, 19.7%, 17.2%, 7.2%, and 7.8%, respectively.

## 6. Conclusion

This work investigates the design problem of deep learning systems robust to multiple perturbations. The model ensemble approach is adopted, where an ensemble of experts dealing with various perturbation types are integrated through a trainable aggregator. Our *MoL* enables great flexibility against various perturbation types. Furthermore, we explore model compression (pruning) to address the overlarge model size problem in model ensembling. Compared with state-of-the-art defenses against multiple perturbation attacks, our proposed *MoL* defense based on DNN ensemble and compression achieves the highest performance.

## 7. Acknowledgements

This work was performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore National Laboratory under Contract No. DE-AC52-07NA27344 and LDRD Project No. 20-SI-005 (LLNL-CONF-820522).

## References


[1] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3366–3375, 2017.

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

[3] Saikiran Bulusu, Bhavya Kailkhura, Bo Li, Pramod K Varshney, and Dawn Song. Anomalous example detection in deep learning: A survey. *IEEE Access*, 8:132330–132347, 2020.

[4] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z Morley Mao. Adversarial sensor attack on lidar-based perception in autonomous driving. In *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, pages 2267–2281, 2019.

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pages 39–57. IEEE, 2017.

[6] Jinghui Chen and Quanquan Gu. Rays: A ray searching method for hard-label adversarial attack. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1739–1747, 2020.

[7] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[8] Alesia Chernikova, Alina Oprea, Cristina Nita-Rotaru, and BaekGyu Kim. Are self-driving cars secure? evasion attacks against deep neural networks for steering angle prediction. In *2019 IEEE Security and Privacy Workshops (SPW)*, pages 132–137. IEEE, 2019.

[9] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3213–3223, 2016.

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[11] Guneet S Dhillon, Kamyar Azizzadenesheli, Zachary C Lipton, Jeremy D Bernstein, Jean Kossaifi, Aran Khanna, and Animashree Anandkumar. Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations*, 2018.

[12] Peiyan Dong, Siyue Wang, Wei Niu, Chengming Zhang, Sheng Lin, Zhengang Li, Yifan Gong, Bin Ren, Xue Lin, and Dingwen Tao. Rtmobile: Beyond real-time mobile acceleration of rnns for speech recognition. In *2020 57th ACM/IEEE Design Automation Conference*, pages 1–6. IEEE, 2020.

[13] Yinpeng Dong, Zhijie Deng, Tianyu Pang, Hang Su, and Jun Zhu. Adversarial distributional training for robust deep learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

[14] Logan Engstrom, Brandon Tran, Dimitris Tsipras, Ludwig Schmidt, and Aleksander Madry. Exploring the landscape of spatial robustness. In *International Conference on Machine Learning (ICML)*, pages 1802–1811. PMLR, 2019.

[15] Takanori Fujisawa and Masaaki Ikehara. High-accuracy image rotation and scale estimation using radon transform and sub-pixel shift estimation. *IEEE Access*, 7:22719–22728, 2019.

[16] Tejas Gokhale, Rushil Anirudh, Bhavya Kailkhura, Jayaraman J Thiagarajan, Chitta Baral, and Yezhou Yang. Attribute-guided adversarial training for robustness to natural perturbations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

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

[18] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens van der Maaten. Countering adversarial images using input transformations. In *International Conference on Learning Representations (ICLR)*, 2018.

[19] Yiwen Guo, Anbang Yao, and Yurong Chen. Dynamic network surgery for efficient dnns. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1379–1387, 2016.

[20] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. In *International Conference on Learning Representations (ICLR)*, 2016.

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

[22] Yang He, Ping Liu, Ziwei Wang, Zhilan Hu, and Yi Yang. Filter pruning via geometric median for deep convolutional neural networks acceleration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4340–4349, 2019.

[23] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdelrahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6):82–97, 2012.

[24] Sanjay Kariyappa and Moinuddin K Qureshi. Improving adversarial robustness of ensembles with diversity training. *arXiv preprint arXiv:1901.09981*, 2019.

[25] Hao Li, Asim Kadav, Igor Durdanovic, Hanan Samet, and Hans Peter Graf. Pruning filters for efficient convnets. *arXiv preprint arXiv:1608.08710*, 2016.

[26] Ning Liu, Xiaolong Ma, Zhiyuan Xu, Yanzhi Wang, Jian Tang, and Jieping Ye. Autocompress: An automatic dnn structured pruning framework for ultra-high compression rates. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 34, pages 4876–4883, 2020.

[27] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *International Conference on Learning Representations (ICLR)*, 2017.

[28] Jian-Hao Luo, Jianxin Wu, and Weiyao Lin. Thinet: A filter level pruning method for deep neural network compression. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 5058–5066, 2017.

[29] Xiaolong Ma, Zhengang Li, Yifan Gong, Tianyun Zhang, Wei Niu, Zheng Zhan, Pu Zhao, Jian Tang, Xue Lin, Bin Ren, et al. Blk-rew: A unified block-based dnn pruning framework using reweighted regularization method. *arXiv preprint arXiv:2001.08357*, 2020.

[30] Xiaolong Ma, Sheng Lin, Shaokai Ye, Zhezhi He, Linfeng Zhang, Geng Yuan, Sia Huat Tan, Zhengang Li, Deliang Fan, Xuehai Qian, et al. Non-structured dnn weight pruning–is it beneficial in any platform? *IEEE Transactions on Neural Networks and Learning Systems (T NNLS)*, 2021.

[31] 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.

[32] Pratyush Maini, Xinyun Chen, Bo Li, and Dawn Song. Classifying perturbation types for robustness against multiple adversarial perturbations. In *International Conference on Machine Learning (ICML) Workshop on Uncertainty and Robustness in Deep Learning*, 2020.

[33] Pratyush Maini, Eric Wong, and Zico Kolter. Adversarial robustness against the union of multiple perturbation models. In *International Conference on Machine Learning (ICML)*, pages 6640–6650. PMLR, 2020.

[34] Roland Meier, Thomas Holterbach, Stephan Keck, Matthias Stähli, Vincent Lenders, Ankit Singla, and Laurent Vanbever. (self) driving under the influence: Intoxicating adversarial network inputs. In *Proceedings of the 18th ACM Workshop on Hot Topics in Networks*, pages 34–42, 2019.

[35] 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 Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2574–2582, 2016.

[36] Mesut Ozdag, Sunny Raj, Steven Fernandes, Laura L Pullum, and Sumit Kumar Jha. On the susceptibility of deep neural networks to natural perturbations. Technical report, Oak Ridge National Lab. (ORNL), Oak Ridge, TN (United States), 2019.

[37] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning (ICML)*, pages 4970–4979. PMLR, 2019.

[38] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016.

[39] Jonas Rauber, Wieland Brendel, and Matthias Bethge. Foolbox: A python toolbox to benchmark the robustness of machine learning models. In *Reliable Machine Learning in the Wild Workshop, 34th International Conference on Machine Learning (ICLR)*, 2017.

[40] Jérôme Rony, Luiz G Hafemann, Luiz S Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4322–4330, 2019.

[41] David Stutz, Matthias Hein, and Bernt Schiele. Confidence-calibrated adversarial training: Generalizing to unseen attacks. In *International Conference on Machine Learning (ICML)*, pages 9155–9166. PMLR, 2020.

[42] Jiachen Sun, Yulong Cao, Qi Alfred Chen, and Z Morley Mao. Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, pages 877–894, 2020.

[43] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 31, 2017.

[44] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna Estrach, Dumitru Erhan, Ian Goodfellow, and Robert Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

[45] Florian Tramer and Dan Boneh. Adversarial training and robustness for multiple perturbations. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2019.

[46] Kaidi Xu, Hongge Chen, Sijia Liu, Pin-Yu Chen, Tsui-Wei Weng, Mingyi Hong, and Xue Lin. Topology attack and defense for graph neural networks: An optimization perspective. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2019.

[47] Kaidi Xu, Sijia Liu, Pu Zhao, Pin-Yu Chen, Huan Zhang, Quanfu Fan, Deniz Erdogmus, Yanzhi Wang, and Xue Lin. Structured adversarial attack: Towards general implementation and better interpretability. In *International Conference on Learning Representations (ICLR)*, 2019.

[48] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. In *European Conference on Computer Vision (ECCV)*, pages 665–681. Springer, 2020.

[49] Huanrui Yang, Jingyang Zhang, Hongliang Dong, Nathan Inkawhich, Andrew Gardner, Andrew Touchet, Wesley Wilkes, Heath Berry, and Hai Li. Dverge: Diversifying vulnerabilities for enhanced robust generation of ensembles. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 33, 2020.

[50] 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 (ICCV)*, pages 111–120, 2019.

[51] Pu Zhao, Sijia Liu, Yanzhi Wang, and Xue Lin. An admm-based universal framework for adversarial attacks on deep neural networks. In *Proceedings of the 26th ACM international conference on Multimedia (MM)*, pages 1065–1073, 2018.