COLLABORATE TO DEFEND AGAINST ADVERSARIAL ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Adversarially robust learning methods require invariant predictions to a small neighborhood of its natural inputs, thus often encountering insufficient model capacity. Learning multiple sub-models in an *ensemble* can mitigate this insufficiency, further improving both generalization and robustness. However, an *ensemble* still wastes the limited capacity of multiple models. To optimally utilizing the limited capacity, this paper proposes to learn a *collaboration* among multiple sub-models. Compared with the *ensemble*, the *collaboration* enables the possibility of correct predictions even if there exists a single correct sub-model. Besides, learning a collaboration could enable every sub-model to fit its vulnerability area and reserve the rest of the sub-models to fit other vulnerability areas. To implement the idea, we propose a collaboration framework—CDA^2 the abbreviation for Collaborate to Defend against Adversarial Attacks. CDA^2 could effectively minimize the vulnerability overlap of all sub-models and then choose a representative sub-model to make correct predictions. Empirical experiments verify that CDA^2 outperforms various ensemble methods against black-box and white-box adversarial attacks.

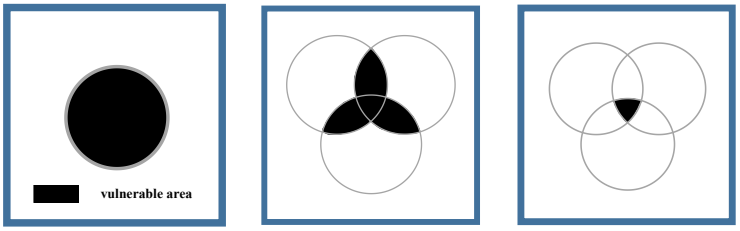
1 INTRODUCTION

Safety-critical applications (such as in medicine and finance) require the *adversarial robustness* of deep models (Goodfellow et al., 2015; Szegedy et al., 2014). An adversarially robust learning method requires invariant predictions to a small neighborhood of its natural inputs, thus often encountering insufficient model capacity (Zhang et al., 2021; Yu et al., 2021a). This limits the further improvement of robustness and has the undesirable degradation of generalization (Madry et al., 2018).

Learning multiple sub-models in an ensemble (Breiman, 1996; Freund et al., 1996) can mitigate this insufficiency (Pang et al., 2019; Kariyappa & Qureshi, 2019; Yang et al., 2020a). Remarkably, Pang et al. (2019), Kariyappa & Qureshi (2019) and Yang et al. (2020a) minimized the vulnerability overlaps between each pair of sub-models and improved both robustness and generalization over a single model.

However, an ensemble wastes the limited capacity of multiple models. In the example of three sub-models (see Figure 1(b)), the adversarial input that lies in the black areas can fool the ensemble successfully, i.e., more than half of sub-models must correctly classify the adversarial input. Therefore, the ensemble's voting-based strategy excludes the possibility that *true predictions remain with the minority*. Besides, learning an ensemble requires more than half of the sub-models to fit the same vulnerability areas, which leaves the following question unanswered whether we could only leverage a single sub-model to fit a vulnerability area and reserve the rest of the sub-models to fit other vulnerability areas.

To optimally utilizing the limited capacity, this paper proposes to *learn a collaboration* among multiple sub-models. As shown in Figure 1(c), the adversarial input that lies in the vulnerability overlaps of all sub-models can undoubtedly fool the collaboration. Compared with the ensemble in Figure 1(b)), collaboration enables the possibility of correct predictions even if there exists a single correct sub-model merely. Besides, learning a collaboration could enable every sub-model to fit its vulnerability area, which could collectively fix broader vulnerability areas than the ensemble does. Then, sub-models could collaboratively choose trustworthy ones to make the final predictions.



(a) Single model (b) Ensemble (c) Collaboration

Figure 1: Illustrations of the vulnerability area of (a) Single model (b) Ensemble, and (c) Collaboration. The black area represents the vulnerability area in which the model is undoubtedly fooled.

To realize the idea, we propose a collaboration framework—Collaborate to Defend against Adversarial Attacks (CDA^2) (Algorithms 1 and 2). In CDA^2, each sub-model has dual heads: one outputs a vector of predicted probability $f_\theta(\cdot)$; another outputs a scalar that measures posterior probability density (PPD) of the prediction. In the training phase, given a natural input x, the M sub-models generate a set of M adversarial variants $\{\tilde{x}^{\theta_i}\}_{i=1}^M$; then, each sub-model chooses an easy one(s) to feed itself. The PPD head is meanwhile updated by comparing the predicted probability on the true label—$f_\theta^y(\cdot)$ (a scalar). In the inference phase, given an input, CDA^2 chooses a sub-model with the largest PPD value as the representative to output the prediction.

We highlight our key contributions as follows.

- We provide a new perspective on learning multiple sub-models for defending against adversarial attacks. We theoretically show the collaboration makes better decisions than the ensemble, which implies collaboration may fix broader vulnerability areas.
- We propose a novel collaboration framework—CDA^2 (see Section 3.2). In the training phase, CDA^{2}could effectively minimize the vulnerability overlap of all sub-models; In the inference phase, CDA^2 could effectively choose a representative sub-model to make correct predictions. We also provide a comprehensive analysis illustrating the rationale of CDA^2.
- Empirical experiments verify that CDA^2 outperforms various ensemble methods against black-box and white-box adversarial attacks.

2 Related Works

Adversarial attack Adversarial attacks aim to craft the human-imperceptible adversarial input to fool the deep models. Adversarial attacks could be roughly divided into white-box attacks in which the adversary is fully aware of the model's structures (Goodfellow et al., 2015; Moosavi-Dezfooli et al., 2016; Carlini & Wagner, 2017b; Chen et al., 2018; Athalye et al., 2018; Xiao et al., 2018; Zheng et al., 2019; Wong et al., 2019; Mopuri et al., 2019; Alaifari et al., 2019; Sriramanan et al., 2020; Wu et al., 2020b; Croce & Hein, 2020; Yu et al., 2021b) and black-box attacks in which the deep models are treated as black boxes to the adversary (Cheng et al., 2019; 2020; Wu et al., 2020a; Chen et al., 2020a; Li et al., 2020a; Rahmati et al., 2020; Yan et al., 2021b; Hendrycks et al., 2021; Dong et al., 2018; Xie et al., 2019). This paper focuses on building effective defense and select both white-box and black-box attack methods as our robustness evaluation metrics.

Adversarial defense Defending adversarial attacks is a challenging task and researchers have proposed various solutions. *Certified defense* tries to learn provably robust deep models against norm-bounded (e.g., ℓ_2 and ℓ_∞) perturbations (Wong & Kolter, 2018; Tsuzuku et al., 2018; Weng et al., 2018; Mirman et al., 2018; Hein & Andriushchenko, 2017; Lécuyer et al., 2019; Xiao et al., 2019; Cohen et al., 2019; Balunovic & Vechev, 2020a; Zhang et al., 2020a; Singla & Feizi, 2020; Balunovic & Vechev, 2020b; Zou et al., 2021). *Empirical defense* leverages adversarial data to build effective defense such as *adversary detection* (Metzen et al., 2017; Li & Li, 2017; Carlini & Wagner, 2017a; Tian et al., 2018; Ma et al., 2018b; Lee et al., 2018; Pang et al., 2018; Smith & Gal, 2018; Roth et al., 2019; Liu et al., 2019; Yin & Rohde, 2020; Sperl et al., 2020; Cohen et al., 2020; Sheikholeslami et al., 2021; Chen et al., 2021a; Yang et al., 2020b; Qin et al., 2020; Tian et al., 2021; Wu et al., 2021) and *adversarial training* (AT), in which AT stands out as the most effective defense. Researchers have investigated various aspects of AT, such as improving AT's robustness or generalization (Madry

et al., 2018; Yan et al., 2018; Wu et al., 2018; Cai et al., 2018; Najafi et al., 2019; Alayrac et al., 2019; Carmon et al., 2019; Dong & Lin, 2019; Farnia et al., 2019; Song et al., 2019; Zhang et al., 2019b; Wang et al., 2019; Tramèr & Boneh, 2019; Zhang & Wang, 2019; Stutz et al., 2020; Pang et al., 2020; Gan et al., 2020; Dong et al., 2020; Zhang et al., 2020b; Chen et al., 2020b; Song et al., 2020; Ding et al., 2020; Wang et al., 2020b; Zhang et al., 2021), fixing AT's undesirable robust overfitting (Rice et al., 2020; Chen et al., 2021b), improving AT's training efficiency (Zhang et al., 2019a; Shafahi et al., 2019; Zheng et al., 2020; B.S. & Babu, 2020; Andriushchenko & Flammarion, 2020; Wong et al., 2020), understanding/interpreting AT's unique traits (Nakkiran, 2019; Yin et al., 2019; Gao et al., 2019; Cranko et al., 2019; Zhang et al., 2019c; Liu et al., 2020; Roth et al., 2020; Wang et al., 2020a; Zhang et al., 2020c; Li et al., 2020b; Zou et al., 2021; Mehrabi et al., 2021; Xu et al., 2021), etc. Besides, researchers have alao actively investigated robust-structured models (Cisse et al., 2017; Xie et al., 2020; Moosavi-Dezfooli et al., 2019; Xie & Yuille, 2020; Yan et al., 2021a; Du et al., 2021; Pang et al., 2021). Nevertheless, the above research thoroughly investigated a single model; this paper focuses on the collaboration among multiple models for adversarial defense.

Ensemble methods for adversarial robustness The most relevant studies are the ensemble methods. Ensemble methods such as bagging (Breiman, 1996) and boosting (Freund et al., 1996) have been investigated for significantly improving the model's generalization. Motivated by benefits of ensemble methods in improving generalization, researchers introduced an ensemble to improve the model robustness (Yang et al., 2020a; Kariyappa & Qureshi, 2019; Pang et al., 2019; Tramèr et al., 2018). Tramèr et al. (2018) proposed to reduce the adversarial transferability by training a single model with adversarial examples from multiple pretrained sub-models. Pang et al. (2019) introduce a regularization method—ADP—to encourage high diversity in the non-maximal predictions of sub-models. Kariyappa & Qureshi (2019) improved the ensemble diversity by maximizing the introduced cosine distance between the gradients of sub-models with respect to the input. Yang et al. (2020a) proposed to distill non-robust features in the input and diversify the adversarial vulnerability. These methods reduced overlaps of vulnerability areas between sub-models (Yang et al., 2020a).

To further improve the ensembles, mixture-of-experts (MOE) assume that the problem space can be divided into multiple sub-problems through a gate module; the gate module specifies each sub-model on a specific sub-problem (Jacobs et al., 1991; Ma et al., 2018a). Nevertheless, to the best of our knowledge, MOE-based methods have been not applied to help adversarial robustness. Inspired by MOE, we propose the collaboration framework to defend against adversary attacks.

3 Collaboration to Defend Against Adversarial Attack

3.1 Superiority of Collaboration

This section shows a *collaboration*, in theory, could make better decisions than an *ensemble*.

Ensemble Suppose that there are M learned sub-models $\{f_{\theta_1}, f_{\theta_2}, ..., f_{\theta_M}\}$, given an input x, M sub-models make predictions $\left\{f_{\theta_1}(x), f_{\theta_2}(x), ..., f_{\theta_M(x)}\right\}$. The ensemble outputs a final prediction $\text{ensemble}(x, f_{\theta_1}, ..., f_{\theta_M})$ by the voting-based strategy:

$$\text{ensemble}(x, f_{\theta_1}, ..., f_{\theta_M}) = \arg\max_{y \in \{1,...,K\}} \left(\sum_{i=1}^{M} \mathbb{1}_{y = f_{\theta_i}(x)} \right), \tag{1}$$

where $\mathbb{1}$ is the indicator function. Note that the ensemble outputs the predicted label y that agrees with the majority predictions of the sub-models.

Definition 1 (best-performing sub-model)**.** *Given an input x and its label y, the best-performing sub-model achieves the lowest objective loss on the data (x, y) among all M sub-models:*

$$f_{\theta_{\text{best}}}(x) = \min_{f_{\theta_i} \in \left\{f_{\theta_1}, .., f_{\theta_M}\right\}} \ell(f_{\theta_i}(x), y). \tag{2}$$

Note that the best-performing sub-model is w.r.t. the input data (x, y), i.e., different input data correspond to different best-performing sub-models.

Collaboration Suppose that there are M learned sub-models $\{f_{\theta_1}, f_{\theta_2}, ..., f_{\theta_M}\}$. Given an input x, sub-models make predictions $\left\{f_{\theta_1}(x), f_{\theta_2}(x), ..., f_{\theta_M(x)}\right\}$. The collaboration tries to output a final

prediction $\text{collaboration}(x, f_{\theta_1}, ..., f_{\theta_M})$ by the best-performing sub-model:

$$\text{collaboration}(x, f_{\theta_1}, ..., f_{\theta_M}) = f_{\theta_{\text{best}}}(x). \tag{3}$$

Proposition 1. *Given M learned sub-models, the predicted accuracy of the collaboration is upper-bounded that of the ensemble, i.e.,*

$$\mathbb{E}_{(x,y)\in D}\left[\mathbb{1}_{\text{collaboration}(x,f_{\theta_1},...,f_{\theta_M})=y}\right] \geq \mathbb{E}_{(x,y)\in D}\left[\mathbb{1}_{\text{ensemble}(x,f_{\theta_1},...,f_{\theta_M})=y}\right]. \tag{4}$$

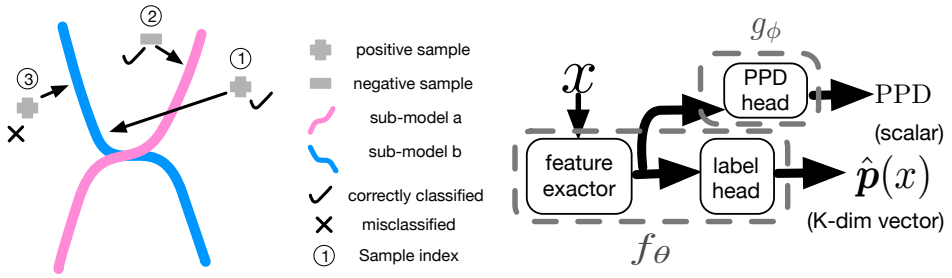
Proof. Given an $(x, y) \in D$, if the ensemble's prediction is correct, at least one sub-model makes correct prediction, i.e., $\mathbb{1}_{f_{\theta_{\text{best}}}(x)=y}$ holds; therefore, the collaboration' prediction is correct. If the collaboration' prediction is correct, there exists a case that the majority of sub-models make the consistent but wrong predictions, while a single sub-model's prediction is correct; then, ensemble's prediction is wrong. Therefore, Proposition 1 holds. □

From Proposition 1, a collaboration can theoretically achieve an equal or higher performance than an ensemble. Next, we will introduce a realization of our collaboration framework.

3.2 Realization of Collaboration for Defending Against Adversarial Attack

This section realizes the framework of Collaboration for Defending against Adversarial Attack (CDA^2).

Notation We firstly introduce the needed notations. Suppose $\mathcal{X}$ and $\mathcal{Y}$ denote input space and output space, where $\mathcal{Y} = \{1, ..., K\}$ for a K-class classification problem. There are N samples in the dataset $D = \{(x, y)\}$, where $x \in \mathcal{X}$ and $y \in \mathcal{Y}$. Let $d_{\text{inf}}(x, x') = \|x - x'\|_\infty$ denotes the infinity distance metric, and $\mathcal{B}_\epsilon[x] = \{x' \in \mathcal{X} \mid d_{\text{inf}}(x, x') \leq \epsilon\}$ is the closed ball of of radius $\epsilon > 0$ centered at x. To search for adversarial data within norm ball $\mathcal{B}_\epsilon[x]$, Madry et al. (2018) proposed a projected gradient descent (PGD) method that iteratively searches for adversarial data $\tilde{x}$ (x refers to natural data). $f_\theta(x)$ outputs a K-dimensional predicted probability, i.e., $\hat{\boldsymbol{p}}(x) = [\hat{p}_1(x), ..., \hat{p}_k(x)]$.

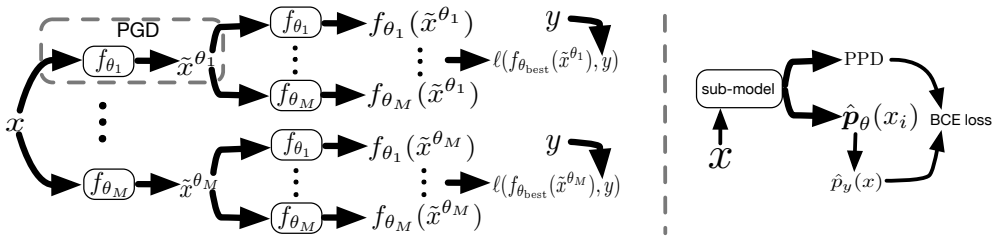


(a) Assign data to their best-performing sub-models (b) Dual-head structured sub-models

Figure 2: (a) The blue and pink lines denote two sub-models. Each sub-model makes negative predictions ($-$) on its left and makes positive predictions on its right ($+$). The given data will be assigned to the sub-model that has the lowest objective loss. The arrows represent the data assignment. (b) Each sub-model has two heads—*label head* that outputs the predicted probability (vector) and *PPD head* that approximates the value of posterior probability density (scalar) of the predicted probability.

Goal of collaboration 1) ensure the correct prediction of the best-performing sub-model for a given input, and 2) select the best-performing sub-model among all sub-models to make prediction.

First, intuitively, every sub-model in a collaboration should maximize its expertise to fit its areas and leave the remaining areas fitted by others. As a result, the collaboration can minimize the vulnerability overlaps of all sub-models. Section 4 shows "minimizing the vulnerability overlap of all sub-models" is "minimizing the objective loss of the best-performing sub-models". Therefore, during the training phase, the given data should always be allocated to the sub-model that has the lowest objective loss.



(a) optimizing the module f_θ (b) optimizing g_ϕ

Figure 3: Optimization process of the M sub-models in a collaboration

In other words, the sub-models always choose the easiest data to learn. In the example of Figure 2(a), $i)$ Data③ is misclassified by both sub-models. The blue sub-model is near Data③ and has the lowest objective loss. We assign the blue sub-model to fit Data③. $ii)$ Data② is correctly classified by the pink model but wrongly classified by the blue model; for ease of effort, we assign the pink model to fit Data②, because the collaboration can correctly be classified Data② through selecting the pink model as the representative. $iii)$ Data① is correctly classified by both models. The blue model is far from Data① and takes the lowest effort on fitting it; therefore, we assign the blue model to fit Data①.

Second, to select the best-performing sub-model, we construct *dual-head structured sub-models*. As shown in Figure 2(b), our sub-model has dual heads: 1) f_θ contains the feature extractor module and the label head and predicts the label probability $f_\theta(x) = \hat{\boldsymbol{p}}(x) = [\hat{p}_1(x), ..., \hat{p}_K(x)]$ (a vector); 2) the PPD head g_ϕ approximates the *posterior probability density* (PPD) (a scalar) of each prediction $\hat{\boldsymbol{p}}(x)$.

Note that the PPD is the likelihood that a given prediction $\hat{\boldsymbol{p}}(x)$ equals the true label distribution $\boldsymbol{p}(x)$. Thus, the PPD value measures the quality of the sub-model's predictions. In other words, the best-performing sub-models can be decided by the PPD values because the largest PPD value corresponds to the lowest objective loss, and vice versa (see theoretical proof in Proposition 2).

Collaborate to defend against adversarial attack To defend against adversarial attack, the collaboration needs to learn from adversarial data. Algorithm 1 along with Figure 3 articulates how to learn such the collaboration, and Algorithm 2 articulates how to use the collaboration to make predictions.

As shown in Figure 3(a), given natural training data x and its label y, we use the PGD method to obtain M adversarial variants $\{\tilde{x}^{\theta_i}\}_{i=1}^M$ of M sub-models. For each adversarial variant $\tilde{x}^{\theta_i}$, we assign the best-performing sub-model to learn and update its feature extractor and label head. This process corresponds to Lines 1–3 and Lines 8–9 in Algorithm 1. Note that in Line 8, we use a surrogate loss to approximate the sub-model assignment process (reasons see Eq. 6 and 7).

As shown in Figure 3(b), given an adversarial variant $\tilde{x}^{\theta_i}$, we propose to use the binary-cross-entropy (BCE) loss between the predicted label probability on the true label (i.e., $\hat{p}_y(\tilde{x}^{\theta_i})$) and the approximated PPD (i.e., $g_\phi(\tilde{x}^{\theta_i})$) to update each sub-model's PPD head. This process corresponds to Lines 5–6 in Algorithm 1.

As shown in Algorithm 2, once M sub-models are properly learned, CDA^2 chooses a representative sub-model whose PPD value is highest among all sub-models, and then outputs this sub-model's prediction.

3.3 Analyses of CDA^2

Optimizing the best-performing sub-models We firstly show that minimizing the vulnerability overlap of all sub-models is equal to minimizing the objective loss of the best-performing sub-models. For ease of optimization of the best-performing sub-model, we provide a surrogate loss.

The vulnerability overlap of all sub-models refers to the set of adversarial data $(\tilde{x}, y)$ that are misclassified by all sub-models, i.e., all sub-models' objective loss is higher than a certain degree δ:

$$\min_{\theta_i \in \{\theta_1, ..., \theta_M\}} \ell(f_{\theta_i}(\tilde{x}, y)) > \delta, \quad \text{where } \tilde{x}_i \in \tilde{D}, \tag{5}$$

where $\tilde{D}$ denotes the vulnerability overlap of all sub-models.

Algorithm 1 Collaboration for Defending against Adversarial Attack (CDA^2) (training phase)

Input: sub-models with duel heads $\{f_{\theta_i}\}_{i=1}^M$ and $\{g_{\phi_i}\}_{i=1}^M$, where f_{θ_i} outputs the label prediction and g_{ϕ_i} outputs the approximated PPD, training dataset D, hyperparameter σ;

1: **for** each data $(x, y) \in D$ **do**
2: **for** each sub-model $f_{\theta_i}, i = 1, 2, ..., M$ **do**
3: Obtain the adversarial data $\tilde{x}^{\theta_i}$ of the sub-model f_{θ_i} using PGD method;
4: **for** each sub-model $f_{\theta_j}, j = 1, 2, ..., M$ (in parallel) **do**
5: Calculate the approximated PPD, i.e., $g_{\phi_j}(\tilde{x}^{\theta_i})$;
6: Minimize BCE loss $\ell_\phi = \mathrm{BCE}(g_{\phi_j}(\tilde{x}^{\theta_i}), \hat{p}_y(\tilde{x}^{\theta_i}))$ to update module g_{ϕ_j};
7: Collect sub-model i's cross entropy (CE) loss on data $\tilde{x}^{\theta_i}$: $\ell\left(f_{\theta_j}\left(\tilde{x}^{\theta_i}\right), y\right)$;
8: Calculate surrogate loss on data $\tilde{x}^{\theta_i}$: $\hat{\ell}_m = -\sigma \ln \sum_{j=1}^M \exp\left(\frac{-\ell\left(f_{\theta_j}\left(\tilde{x}^{\theta_i}\right), y\right)}{\sigma}\right)$;
9: Update $\{f_{\theta_i}\}_{i=1}^M$ by minimizing $\hat{\ell}_m$; // choose the best-performing sub-model to fit $\tilde{x}^{\theta_i}$
10: **return** the learned sub-models $\{f_{\theta_i}\}_{i=1}^M$ with $\{g_{\phi_i}\}_{i=1}^M$.

To reduce the vulnerability overlap of all sub-models, we only need to reduce objective loss of a single model, which is equal to minimizing the loss of the best-performing sub-model, i.e.,

$$\min_{\{\theta_1,\theta_2,\ldots,\theta_M\}} \mathbb{E}_{(x,y)\in D}\left(\mathbb{E}_{\theta_i\in\{\theta_1,\theta_2,\ldots,\theta_M\}} \min_{\theta_j\in\{\theta_1,\theta_2,\ldots,\theta_M\}} \ell\left(f_{\theta_j}(\tilde{x}^{\theta_i}), y\right)\right), \tag{6}$$

where $\tilde{x}^{\theta_i}$ is the adversarial data generated by the sub-model f_{θ_i}.

While directly performing the outer minimization in Eq.(6) may cause a trivial solution (e.g., there is only one optimized sub-model), for ease of the optimization of Eq.(6) (Corresponding to Lines 8–9 in Algorithm 1), we provide a surrogate objective as follows.

$$\min_{\{\theta_1,\theta_2,\ldots,\theta_M\}} \mathbb{E}_{(x,y)\in D}\left(\mathbb{E}_{\theta_i\in\{\theta_1,\theta_2,\ldots,\theta_M\}} \hat{\ell}_m(\tilde{x}^{\theta_i}, y)\right), \tag{7}$$

where $\hat{\ell}_m(\tilde{x}^{\theta_i}, y) = -\sigma \ln \sum_{j=1}^M \exp\left(\frac{-\ell\left(f_{\theta_j}(\tilde{x}^{\theta_i}), y\right)}{\sigma}\right)$ and $\sigma > 0$ is a pre-defined hyper-parameter.

In Eq.(7), we approximate the objective $\min_{\theta_j\in\{\theta_1,\theta_2,\ldots,\theta_M\}} \ell\left(f_{\theta_j}(\tilde{x}^{\theta_i}), y\right)$ using a smooth surrogated maximum function due to

$$\min_{\theta_j\in\{\theta_1,\theta_2,\ldots,\theta_M\}} \ell\left(f_{\theta_j}(\tilde{x}^{\theta_i}), y\right) - \delta\cdot\ln(M) \le \hat{\ell}_m(\tilde{x}^{\theta_i}), y) \le \min_{\theta_j\in\{\theta_1,\theta_2,\ldots,\theta_M\}} \ell\left(f_{\theta_j}(\tilde{x}^{\theta_i}), y\right). \tag{8}$$

Proof. Since we have

$$\max_j \exp\left(\frac{-\ell\left(f_{\theta_j}(\tilde{x}^{\theta_i}), y\right)}{\sigma}\right) \le \sum_{j=1}^M \exp\left(\frac{-\ell\left(f_{\theta_j}(\tilde{x}^{\theta_i}), y\right)}{\sigma}\right) \le M\cdot\max_j \exp\left(\frac{-\ell\left(f_{\theta_j}(\tilde{x}^{\theta_i}), y\right)}{\sigma}\right), \tag{9}$$

considering that $-\delta\ln(x)$ monotonically decreases w.r.t x, we have

$$-\delta\ln\left(M\cdot\max_j \exp\left(\frac{-\ell\left(f_{\theta_j}(\tilde{x}^{\theta_i}), y\right)}{\sigma}\right)\right) \le \hat{\ell}_m(f_{\theta_j}(\tilde{x}^{\theta_i}, y)) \le -\delta\ln\max_j \exp\left(\frac{-\ell\left(f_{\theta_j}(\tilde{x}^{\theta_i}), y\right)}{\sigma}\right), \tag{10}$$

where $-\delta\ln\left(M\cdot\max_j \exp\left(\frac{-\ell\left(f_{\theta_j}(\tilde{x}^{\theta_i}), y\right)}{\sigma}\right)\right) = \min_{\theta_j\in\{\theta_1,\theta_2,\ldots,\theta_M\}} \ell\left(f_{\theta_j}(\tilde{x}^{\theta_i}), y\right) - \delta\cdot\ln(M)$ and $-\delta\ln\exp\left(\max_j \frac{-\ell\left(f_{\theta_j}(\tilde{x}^{\theta_i}), y\right)}{\sigma}\right) = \min_{\theta_j\in\{\theta_1,\theta_2,\ldots,\theta_M\}} \ell\left(f_{\theta_j}(\tilde{x}^{\theta_i}), y\right)$, so Eq.(8) holds. □

The best-performing sub-model has the highest PPD. We show that a sub-model with the highest PPD achieves the minimum of the objective loss among all sub-models, i.e., the best-performing sub-model.

Algorithm 2 Collaboration for Defending against Adversarial Attack (CDA^2) (inference phase)

Input: the learned sub-models $\{f_{\theta_i}\}_{i=1}^M$ with $\{g_{\phi_i}\}_{i=1}^M$, test input x.
1: **for** all sub-models $f_{\theta_i}, i = 1, ..., M$ (in parallel) **do**
2: make label predictions $\hat{\boldsymbol{p}}(x) = f_{\theta_i}(x)$ and output approximated PPD value $g_{\phi_i}(x)$;
3: **return** prediction $\hat{\boldsymbol{p}}(x)$ whose PPD value $g_\phi(x)$ is the highest among M sub-models.

According to Bayesian theory, for a given input x in a K-class classification problem, the true probability of the corresponding label $\boldsymbol{p}(x) = [p_1(x), ..., p_K(x)]$ is agnostic. Usually we assume $\boldsymbol{p}(x)$ comes from a prior probability distribution, e.g., $\boldsymbol{p}(x) \sim \mathrm{Dir}(\boldsymbol{\alpha})$, where $\mathrm{Dir}(\boldsymbol{\alpha})$ is the Dirichlet distribution with the pre-defined parameter vector $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, ..., \alpha_K]$. The prior probability density function of $\mathrm{Dir}(\boldsymbol{\alpha})$ is

$$\begin{aligned} f(\boldsymbol{p}(x), \boldsymbol{\alpha}) &= \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^{K} p_i(x)^{\alpha_i - 1} \\ \mathrm{B}(\boldsymbol{\alpha}) &= \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma(\alpha_0)}, \quad \alpha_0 = \sum_{i=1}^{K} \alpha_i, \end{aligned} \tag{11}$$

where $\Gamma(\cdot)$ denotes the Gamma function. According to the Bayesian views, when we learn the model in a supervised manner, the label y happens n times after n observations given the data $(x, y) \in D$. Therefore, the posterior probability distribution is $\mathrm{Dir}(\boldsymbol{\alpha'})$ where $\boldsymbol{\alpha'} = [\alpha_1, ..., \alpha_y + n, ..., \alpha_K]$, and the posterior probability density (PPD) function is

$$\mathrm{PPD}(\boldsymbol{p}(x), \boldsymbol{\alpha'}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha'})} \prod_{i=1, i \neq y}^{K} p(x)^{\alpha_i - 1} \cdot p_y(x)^{\alpha_y + n - 1}, \tag{12}$$

where PPD corresponds to the likelihood of a given $\hat{\boldsymbol{p}}(x)$ being the probability of the true label $\boldsymbol{p}(x)$. In other words, the predictions $\hat{\boldsymbol{p}}(x)$ from the sub-model f_{θ_i} with a higher PPD is more likely to be the true label probability. Therefore, we propose to use PPD to measure the quality of predictions.

In Eq.(12), when n (i.e., the number of posterior observations) is sufficiently large, $p_y(x)^{\alpha_y + n - 1}$ becomes a dominant part; therefore, *the value of PPD monotonically increases w.r.t. the value of* $p_y(x)$. Consequently, we have the following proposition.

Proposition 2. *Given an input x, the sub-model that has the highest PPD corresponds to the best-performing sub-model, i.e.,*

$$\underset{j \in \{1,2,...,M\}}{\arg\max} \ \mathrm{PPD}(f_{\theta_j}(x), \boldsymbol{\alpha'}) = \underset{j \in \{1,2,...M\}}{\arg\min} \ \ell(f_{\theta_j}(x), y). \tag{13}$$

Proof. The objective loss depends on the label probability (e.g., cross-entropy loss $\ell = -\ln p_y(x)$), and the loss monotonically decreases with respect to $p_y(x)$, so the best-performing sub-model has the highest label probability. Combined the fact that $\mathrm{PPD}(\boldsymbol{p}(x))$ monotonically increases w.r.t. $p_y(x)$, the best-performing sub-model corresponds to the highest PPD; therefore, Eq.(13) holds. □

From Proposition 2, the PPD value can decide the best-performing sub-model. Given an input to the collaboration, our dual-head structured sub-models can collaboratively decide the best-performing sub-model by just comparing the values of the PPD head (corresponds to Algorithm 2). To learn the PPD head, we could simply compare its output with the predicted probability on the true label (i.e.,$\hat{p}_y(x)$), then update the PPD head by gradient descent (corresponds to Lines 5–6 in Algorithm 1).

Note that the PPD head may be susceptible to adversarial attack in the white-box setting. We can use a simple linear structure to regress the PPD head.

4 Experiments

In this section, we firstly provide a synthetic experiment to illustrate the behavior of our method compared with prior ensemble methods. Then, we provide a series of experiments on a benchmark dataset to verify the effectiveness of our method in defending against adversarial attacks.

4.1 EXPERIMENTAL SETUP

Following the work in (Yang et al., 2020a), we compare our method with various related methods, including ADP (Pang et al., 2019), GAL (Kariyappa & Qureshi, 2019), DVERGE (Yang et al., 2020a). We use ResNet-20 (He et al., 2016) as sub-models in all methods for fair comparisons, and we use CIFAR10 as the data set, a classical image dataset (Krizhevsky et al., 2009) that has 50,000 training images and 10,000 test images.

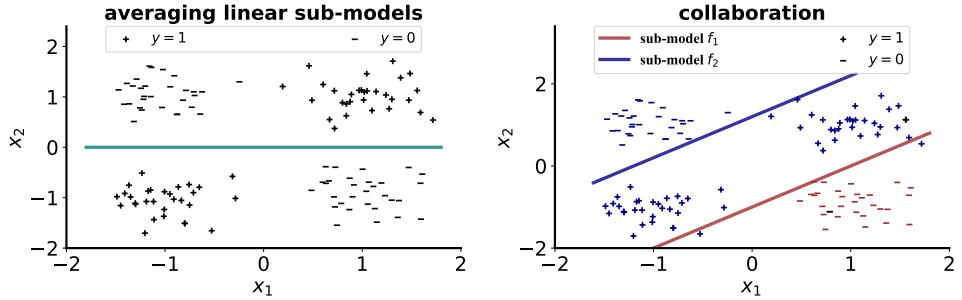
4.2 SYNTHETIC DATA EXPERIMENTS

To demonstrate the behavior of collaboration, we apply our collaboration framework on the well-known **XOR** problem as an example.

In **XOR** problem, There is a binary training set $D = \{x_i, y_i)\}_{i=1}^n$, where $y_i \in \{\pm 1\}$. For the samples with the label $y_i = 1$, the input feature x is independently sampled from the two Gaussian distributions $x \in \mathcal{N}(\boldsymbol{\mu} = [1, 1], \boldsymbol{\sigma} = 0.1 \cdot \boldsymbol{I}_2)$ and $x \in \mathcal{N}(\boldsymbol{\mu} = [-1, -1], \boldsymbol{\sigma} = 0.1 \cdot \boldsymbol{I}_2)$, where $\mathcal{N}(\boldsymbol{\mu} = [1, 1], \boldsymbol{\sigma} = 0.1 \cdot \boldsymbol{I}_2)$ denotes 2-d Gaussian distribution with the mean vector $\boldsymbol{\mu} = [1, 1]$ and covariance matrix $\boldsymbol{\sigma} = 0.1 \cdot \boldsymbol{I}_2$. For the samples with the label $y_i = -1$, we sample the input feature independently $x_i \in \mathcal{N}(\boldsymbol{\mu} = [-1, 1], \boldsymbol{\sigma} = 0.1 \cdot \boldsymbol{I}_2)$ and $x_i \in \mathcal{N}(\boldsymbol{\mu} = [1, -1], \boldsymbol{\sigma} = 0.1 \cdot \boldsymbol{I}_2)$.

Suppose there are two linear sub-models $f_1(x) = \boldsymbol{a}_1 \cdot x + b_1$ and $f_2(x) = \boldsymbol{a}_2 \cdot x + b_2$. Ensemble methods output a prediction $\hat{\boldsymbol{p}}(x)$ by a voting-based strategy e.g., averaging the predictions as the output, i.e., $\hat{\boldsymbol{p}}(x) = \frac{1}{M} \cdot \sum_{i=1}^{M} f_{\theta_i}(x)$.

Collaboration aims to 1). minimize the mean square error (MSE) of the best-performing sub-models; 2). optimize the module g_1 and g_2 to measure the quality of the multiple predictions during inference. From Figure 4(a), learning multiple linear sub-models and averaging the predictions (ensemble)



(a) ensemble (b) collaboration

Figure 4: Illustration of the behavior of the ensemble and the collaboration.

is still a linear model, so it cannot tackle **XOR** problem. Collaboration can address **XOR** which classifies each sample by learning to specify the sub-tasks to different sub-models. From Figure 4(b), the samples are assigned to two sub-models, in which the blue samples are assigned to f_1 and the red samples are assigned to f_2. Finally, it output the prediction by identifying the best-performing sub-models.

4.3 PERFORMANCE ON WHITE-BOX ATTACK

As there are mainly two threat modes in the adversarial attack setting: white-box attack and black-box attack. White-box attack refers to that attackers know all the information about the models, including training data, model architectures, and parameters, while black-box attackers have no access to the information about the model's structures and parameters and rely on surrogate models to generate transferable adversarial examples.

We compare our method with baselines (GAL (Kariyappa & Qureshi, 2019), DVERGE Yang et al. (2020a) and ADP (Pang et al., 2019)) on defending white-box attacks using a subset of **CIFAR10**. In particular, we randomly select 1000 samples under different ϵ. We apply **PGD-10** (Madry et al., 2018) which updates 10 iterations with the step size being 0.01. For **PGD-10** attack, we select the cross-entropy loss to update the perturbations to search for adversarial samples. From Table 1,

Table 1: Robustness results (%) under white-box attack.

methods \ ϵ	0.01	0.02	0.03	0.04	0.05	0.06	0.07
GAL	49.5	31.4	25.5	28.6	19.2	14.1	10.3
DVERGE	63.4	49.6	36.9	33.7	26.4	18.1	12.9
ADP	62.9	48.7	38.9	33.5	26.7	**26.7**	24.3
CDA^2	**64.8**	**51.5**	**44.5**	**38.3**	**34.6**	26.5	**25.5**

CDA^2 achieves a better performance under white-box attack. The results verify that collaboration significantly improves the utilization of the limited model capacity. Therefore, CDA^2 can fit more adversarial data and has a relatively smaller vulnerable area.

4.4 Performance on Black-box attack

Due to the transferability of adversarial examples, black-box adversaries can craft adversarial examples based on surrogate models and perform an attack on the target model. Following the black-box attack setting in (Yang et al., 2020a), we select 1000 test samples randomly and use hold-out baseline ensembles with three ResNet-20 sub-models as the surrogate models to generate adversarial samples. We apply two attack methods in our black-box attack experiments: PGD with momentum (Dong et al., 2018) and M-FGSM (Xie et al., 2019), which are the two commonly used attack methods, and use cross-entropy loss and CW (Carlini & Wagner, 2017b) loss respectively in the two methods. The results are shown in Table 2 and Table 3. In our experiments, GAL is hard to optimize in adversarial training. From Table 2 and Table 3, we show the black-box robustness of ensemble methods across a wide range of attack radius $0.01 \leq \epsilon \leq 0.07$. When $0.01 \leq \epsilon \leq 0.05$, CDA^2 addresses more adversarial data and outperforms the other methods with higher accuracy. With the increase of ϵ, the volume of ϵ-ball increases exponentially, the performances of all methods get worse significantly because of insufficient model capacity.

Table 2: Robustness results (%) under black-box attack (PGD).

methods \ ϵ	0.01	0.02	0.03	0.04	0.05	0.06	0.07
GAL	65.1	49.9	49.7	47.3	53.4	51.1	42.2
DVERGE	77.8	73.6	70.5	67.9	65.4	64.4	**63.5**
ADP	78.3	74.1	69.4	68.7	64.8	63.8	61.2
CDA^2	**79.3**	**76.0**	**72.4**	**69.2**	**67.5**	**65.2**	62

Table 3: Robustness results (%) under black-box attack (M-FGSM).

methods \ ϵ	0.01	0.02	0.03	0.04	0.05	0.06	0.07
GAL	64.8	49	49.9	46.9	53.8	51.6	42.7
DVERGE	77.1	72.7	69.3	67.1	64.4	64.5	**62.3**
ADP	77.3	73.9	68.7	67.5	65.0	**64.5**	60.8
CDA^2	**78.9**	**75.0**	**71.4**	**68.1**	**65.8**	63.4	60.3

5 Conclusion

This paper philosophically answers an essential question that when we should collaborate. (i) If a single model can handle everything, there is no need for multiple models. (ii) If a single model can only handle a part of the whole, collaboration among multiple models makes sense. Adversarial defense is a typical task that falls into the circumstance (ii) because a single model hardly fits adversarial data. We provided a collaboration framework—CDA^2—as the defense strategy over ensemble methods, and empirical experiments indeed verified the efficacy of CDA^2. Future work includes applying our collaboration framework to other areas such as kernel methods, fairness, and global federated model, etc.

ETHICS STATEMENT

In this work, we investigate the robustness issue in deep neural networks (DNNs). We propose a collaboration mechanism for reducing the vulnerability of existing DNNs to enhance their trustworthiness. Having reading the codes of ethics, we make sure that our work conforms to them. In our experiments, we use a public data set which complies with requirements.

REPRODUCIBILITY STATEMENT

Our instructions and experimental settings are illustrated in the maintext. For all theoretical results, we provide rigorous analysis and detailed proofs in maintext. The source code of our method will be made public when it is published.

REFERENCES

Rima Alaifari, Giovanni S. Alberti, and Tandri Gauksson. Adef: an iterative algorithm to construct adversarial deformations. In *ICLR*, 2019.

Jean-Baptiste Alayrac, Jonathan Uesato, Po-Sen Huang, Alhussein Fawzi, Robert Stanforth, and Pushmeet Kohli. Are labels required for improving adversarial robustness? In *NeurIPS*, 2019.

Maksym Andriushchenko and Nicolas Flammarion. Understanding and improving fast adversarial training. In *NeurIPS*, 2020.

Anish Athalye, Nicholas Carlini, and David A. Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

Mislav Balunovic and Martin Vechev. Adversarial training and provable defenses: Bridging the gap. In *ICLR*, 2020a.

Mislav Balunovic and Martin T. Vechev. Adversarial training and provable defenses: Bridging the gap. In *ICLR*, 2020b.

Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

Vivek B.S. and R. Venkatesh Babu. Single-step adversarial training with dropout scheduling. In *CVPR*, 2020.

Qi-Zhi Cai, Chang Liu, and Dawn Song. Curriculum adversarial training. In *IJCAI*, 2018.

Nicholas Carlini and David A. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, AISec@CCS*, 2017a.

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy, SP*, 2017b.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, Percy Liang, and John C. Duchi. Unlabeled data improves adversarial robustness. In *NeurIPS*, 2019.

Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *IEEE Symposium on Security and Privacy, SP 2020*, 2020a.

Kejiang Chen, Yuefeng Chen, Hang Zhou, Chuan Qin, Xiaofeng Mao, Weiming Zhang, and Nenghai Yu. Adversarial examples detection beyond image space. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2021a.

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 *AAAI*, 2018.

Tianlong Chen, Sijia Liu, Shiyu Chang, Yu Cheng, Lisa Amini, and Zhangyang Wang. Adversarial robustness: From self-supervised pre-training to fine-tuning. In *CVPR*, 2020b.

Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *ICLR*, 2021b.

Minhao Cheng, Thong Le, Pin-Yu Chen, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. In *ICLR*, 2019.

Minhao Cheng, Simranjit Singh, Patrick H. Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack. In *ICLR*, 2020.

Moustapha Cisse, Piotr Bojanowski, Edouard Grave, Yann Dauphin, and Nicolas Usunier. Parseval networks: Improving robustness to adversarial examples. In *ICML*, 2017.

Gilad Cohen, Guillermo Sapiro, and Raja Giryes. Detecting adversarial samples using influence functions and nearest neighbors. In *CVPR*, 2020.

Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. In *ICML*, 2019.

Zac Cranko, Aditya Krishna Menon, Richard Nock, Cheng Soon Ong, Zhan Shi, and Christian J. Walder. Monge blunts bayes: Hardness results for adversarial training. In *ICML*, 2019.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

Gavin Weiguang Ding, Yash Sharma, Kry Yik Chau Lui, and Ruitong Huang. Mma training: Direct input space margin maximization through adversarial training. In *ICLR*, 2020.

Jinhao Dong and Tong Lin. Margingan: Adversarial training in semi-supervised learning. In *NeurIPS*, pp. 10440–10449, 2019.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 9185–9193, 2018.

Yinpeng Dong, Zhijie Deng, Tianyu Pang, Jun Zhu, and Hang Su. Adversarial distributional training for robust deep learning. In *NeurIPS*, 2020.

Xuefeng Du, Jingfeng Zhang, Bo Han, Tongliang Liu, Yu Rong, Gang Niu, Junzhou Huang, and Masashi Sugiyama. Learning diverse-structured networks for adversarial robustness. In *Proceedings of the 38th International Conference on Machine Learning*, 2021.

Farzan Farnia, Jesse M. Zhang, and David Tse. Generalizable adversarial training via spectral normalization. In *ICLR*, 2019.

Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *ICML*, 1996.

Zhe Gan, Yen-Chun Chen, Linjie Li, Chen Zhu, Yu Cheng, and Jingjing Liu. Large-scale adversarial training for vision-and-language representation learning. In *NerIPS*, 2020.

Ruiqi Gao, Tianle Cai, Haochuan Li, Cho-Jui Hsieh, Liwei Wang, and Jason D. Lee. Convergence of adversarial training in overparametrized neural networks. In *NeurIPS*, 2019.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *NeurIPS*, 2017.

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, 2021.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

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

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

Mathias Lécuyer, Vaggelis Atlidakis, Roxana Geambasu, Daniel Hsu, and Suman Jana. Certified robustness to adversarial examples with differential privacy. In *Symposium on Security and Privacy (SP)*, 2019.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *NeurIPS*, 2018.

Huichen Li, Xiaojun Xu, Xiaolu Zhang, Shuang Yang, and Bo Li. QEBA: query-efficient boundary-based blackbox attack. In *CVPR*, 2020a.

Xin Li and Fuxin Li. Adversarial examples detection in deep networks with convolutional filter statistics. In *ICCV*, 2017.

Yan Li, Ethan X. Fang, Huan Xu, and Tuo Zhao. Implicit bias of gradient descent based adversarial training on separable data. In *ICLR*, 2020b.

Chen Liu, Mathieu Salzmann, Tao Lin, Ryota Tomioka, and Sabine Süsstrunk. On the loss landscape of adversarial training: Identifying challenges and how to overcome them. In *NeurIPS*, 2020.

Jiayang Liu, Weiming Zhang, Yiwei Zhang, Dongdong Hou, Yujia Liu, Hongyue Zha, and Nenghai Yu. Detection based defense against adversarial examples from the steganalysis point of view. In *CVPR*, 2019.

Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *SIGKDD*, 2018a.

Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi N. R. Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *ICLR*, 2018b.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Mohammad Mehrabi, Adel Javanmard, Ryan A. Rossi, Anup B. Rao, and Tung Mai. Fundamental tradeoffs in distributionally adversarial training. In *ICML*, 2021.

Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. In *ICLR*, 2017.

Matthew Mirman, Timon Gehr, and Martin T. Vechev. Differentiable abstract interpretation for provably robust neural networks. In *ICML*, 2018.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *CVPR*, 2016.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. In *CVPR*, 2019.

Konda Reddy Mopuri, Aditya Ganeshan, and R. Venkatesh Babu. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(10): 2452–2465, 2019. doi: 10.1109/TPAMI.2018.2861800.

Amir Najafi, Shin-ichi Maeda, Masanori Koyama, and Takeru Miyato. Robustness to adversarial perturbations in learning from incomplete data. In *NeurIPS*, 2019.

Preetum Nakkiran. Adversarial robustness may be at odds with simplicity. *arXiv:1901.00532*, 2019.

Tianyu Pang, Chao Du, Yinpeng Dong, and Jun Zhu. Towards robust detection of adversarial examples. In *NeurIPS*, 2018.

Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In *ICLR*, 2019.

Tianyu Pang, Xiao Yang, Yinpeng Dong, Taufik Xu, Jun Zhu, and Hang Su. Boosting adversarial training with hypersphere embedding. In *NeurIPS*, 2020.

Tianyu Pang, Huishuai Zhang, Di He, Yinpeng Dong, Hang Su, Wei Chen, Jun Zhu, and Tie-Yan Liu. Adversarial training with rectified rejection. *arXiv preprint arXiv:2105.14785*, 2021.

Yao Qin, Nicholas Frosst, Sara Sabour, Colin Raffel, Garrison W. Cottrell, and Geoffrey E. Hinton. Detecting and diagnosing adversarial images with class-conditional capsule reconstructions. In *ICLR*, 2020.

Ali Rahmati, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Huaiyu Dai. Geoda: A geometric framework for black-box adversarial attacks. In *CVPR*, 2020.

Leslie Rice, Eric Wong, and J Zico Kolter. Overfitting in adversarially robust deep learning. In *ICML*, 2020.

Kevin Roth, Yannic Kilcher, and Thomas Hofmann. The odds are odd: A statistical test for detecting adversarial examples. In *ICML*, 2019.

Kevin Roth, Yannic Kilcher, and Thomas Hofmann. Adversarial training is a form of data-dependent operator norm regularization. In *NeurIPS*, 2020.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! In *NeurIPS*, 2019.

Fatemeh Sheikholeslami, Ali Lotfi, and J. Zico Kolter. Provably robust classification of adversarial examples with detection. In *ICLR*, 2021.

Sahil Singla and Soheil Feizi. Second-order provable defenses against adversarial attacks. In *ICML*, 2020.

Lewis Smith and Yarin Gal. Understanding measures of uncertainty for adversarial example detection. In *UAI*, 2018.

Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Improving the generalization of adversarial training with domain adaptation. In *ICLR*, 2019.

Chuanbiao Song, Kun He, Jiadong Lin, Liwei Wang, and John E. Hopcroft. Robust local features for improving the generalization of adversarial training. In *ICLR*, 2020.

Philip Sperl, Ching-Yu Kao, Peng Chen, Xiao Lei, and Konstantin Böttinger. DLA: dense-layer-analysis for adversarial example detection. In *IEEE European Symposium on Security and Privacy, EuroS&P*, 2020.

Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, and Venkatesh Babu R. Guided adversarial attack for evaluating and enhancing adversarial defenses. In *NeurIPS*, 2020.

David Stutz, Matthias Hein, and Bernt Schiele. Confidence-calibrated adversarial training: Generalizing to unseen attacks. In *ICML*, 2020.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.

Jinyu Tian, Jiantao Zhou, Yuanman Li, and Jia Duan. Detecting adversarial examples from sensitivity inconsistency of spatial-transform domain. In *AAAI*, 2021.

Shixin Tian, Guolei Yang, and Ying Cai. Detecting adversarial examples through image transformation. In *AAAI*, 2018.

F Tramèr, D Boneh, A Kurakin, I Goodfellow, N Papernot, and P McDaniel. Ensemble adversarial training: Attacks and defenses. In *ICLR*, 2018.

Florian Tramèr and Dan Boneh. Adversarial training and robustness for multiple perturbations. In *NeurIPS*, 2019.

Yusuke Tsuzuku, Issei Sato, and Masashi Sugiyama. Lipschitz-Margin training: Scalable certification of perturbation invariance for deep neural networks. In *NeurIPS*, 2018.

Haotao Wang, Tianlong Chen, Shupeng Gui, Ting-Kuei Hu, Ji Liu, and Zhangyang Wang. Once-for-all adversarial training: In-situ tradeoff between robustness and accuracy for free. In *NeurIPS*, 2020a.

Yisen Wang, Xingjun Ma, James Bailey, Jinfeng Yi, Bowen Zhou, and Quanquan Gu. On the convergence and robustness of adversarial training. In *ICML*, 2019.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020b.

Tsui-Wei Weng, Huan Zhang, Pin-Yu Chen, Jinfeng Yi, Dong Su, Yupeng Gao, Cho-Jui Hsieh, and Luca Daniel. Evaluating the robustness of neural networks: An extreme value theory approach. In *ICLR*, 2018.

Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *ICML*, 2018.

Eric Wong, Frank R. Schmidt, and J. Zico Kolter. Wasserstein adversarial examples via projected sinkhorn iterations. In *ICML*, 2019.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. In *ICLR*, 2020.

Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. In *ICLR*, 2020a.

Kaiwen Wu, Allen Houze Wang, and Yaoliang Yu. Stronger and faster wasserstein adversarial attacks. In *ICML*, 2020b.

Xi Wu, Uyeong Jang, Jiefeng Chen, Lingjiao Chen, and Somesh Jha. Reinforcing adversarial robustness using model confidence induced by adversarial training. In *ICML*, 2018.

Yuhang Wu, Sunpreet S. Arora, Yanhong Wu, and Hao Yang. Beating attackers at their own games: Adversarial example detection using adversarial gradient directions. In *AAAI*, 2021.

Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. In *ICLR*, 2018.

Kai Yuanqing Xiao, Vincent Tjeng, Nur Muhammad (Mahi) Shafiullah, and Aleksander Madry. Training for faster adversarial robustness verification via inducing relu stability. In *ICLR*, 2019.

Cihang Xie and Alan L. Yuille. Intriguing properties of adversarial training at scale. In *ICLR*, 2020.

Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2730–2739, 2019.

Cihang Xie, Mingxing Tan, Boqing Gong, Alan Yuille, and Quoc V Le. Smooth adversarial training. *arXiv preprint arXiv:2006.14536*, 2020.

Han Xu, Xiaorui Liu, Yaxin Li, Anil K. Jain, and Jiliang Tang. To be robust or to be fair: Towards fairness in adversarial training. In *ICML*, 2021.

Hanshu Yan, Jingfeng Zhang, Gang Niu, Jiashi Feng, Vincent Tan, and Masashi Sugiyama. Cifs: Improving adversarial robustness of cnns via channel-wise importance-based feature selection. In *ICML*, 2021a.

Ziang Yan, Yiwen Guo, and Changshui Zhang. Deep defense: Training dnns with improved adversarial robustness. In *NeurIPS*, 2018.

Ziang Yan, Yiwen Guo, Jian Liang, and Changshui Zhang. Policy-driven attack: Learning to query for hard-label black-box adversarial examples. In *ICLR*, 2021b.

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 *NeurIPS*, 2020a.

Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael I. Jordan. ML-LOO: detecting adversarial examples with feature attribution. In *AAAI*, 2020b.

Dong Yin, Kannan Ramchandran, and Peter L. Bartlett. Rademacher complexity for adversarially robust generalization. In *ICML*, 2019.

Xuwang Yin and Soheil Kolouri Gustavo K. Rohde. GAT: generative adversarial training for adversarial example detection and robust classification. In *ICLR*, 2020.

Yaodong Yu, Zitong Yang, Edgar Dobriban, Jacob Steinhardt, and Yi Ma. Understanding generalization in adversarial training via the bias-variance decomposition. *arXiv preprint arXiv:2103.09947*, 2021a.

Yunrui Yu, Xitong Gao, and Cheng-Zhong Xu. LAFEAT: piercing through adversarial defenses with latent features. In *CVPR*, 2021b.

Dinghuai Zhang, Tianyuan Zhang, Yiping Lu, Zhanxing Zhu, and Bin Dong. You only propagate once: Accelerating adversarial training via maximal principle. In *NeurIPS*, 2019a.

Haichao Zhang and Jianyu Wang. Defense against adversarial attacks using feature scattering-based adversarial training. In *NeurIPS*, pp. 1829–1839, 2019.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019b.

Huan Zhang, Hongge Chen, Zhao Song, Duane S. Boning, Inderjit S. Dhillon, and Cho-Jui Hsieh. The limitations of adversarial training and the blind-spot attack. In *ICLR*, 2019c.

Huan Zhang, Hongge Chen, Chaowei Xiao, Sven Gowal, Robert Stanforth, Bo Li, Duane Boning, and Cho-Jui Hsieh. Towards stable and efficient training of verifiably robust neural networks. In *ICLR*, 2020a.

Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan S. Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *ICML*, 2020b.

Jingfeng Zhang, Jianing Zhu, Gang Niu, Bo Han, Masashi Sugiyama, and Mohan Kankanhalli. Geometry-aware instance-reweighted adversarial training. In *ICLR*, 2021.

Yi Zhang, Orestis Plevrakis, Simon S. Du, Xingguo Li, Zhao Song, and Sanjeev Arora. Overparameterized adversarial training: An analysis overcoming the curse of dimensionality. In *NeurIPS*, 2020c.

Haizhong Zheng, Ziqi Zhang, Juncheng Gu, Honglak Lee, and Atul Prakash. Efficient adversarial training with transferable adversarial examples. In *CVPR*, 2020.

Tianhang Zheng, Changyou Chen, and Kui Ren. Distributionally adversarial attack. In *AAAI*, 2019.

Difan Zou, Spencer Frei, and Quanquan Gu. Provable robustness of adversarial training for learning halfspaces with noise. In *ICML*, 2021.