Synergy-of-Experts: Collaborate to Improve Adversarial Robustness

Sen Cui[1*], Jingfeng Zhang[2*], Jian Liang[3], Bo Han[4], Masashi Sugiyama[2,5], Changshui Zhang[1]
[1]Institute for Artificial Intelligence, Tsinghua University (THUAI),
Beijing National Research Center for Information Science and Technology (BNRist),
Department of Automation,Tsinghua University, Beijing, P.R.China
[2] RIKEN Center for Advanced Intelligence Project, Tokyo, Japan
[3] Alibaba Group, China
[4] Hong Kong Baptist University, Hong Kong SAR, China
[5] The University of Tokyo, Tokyo, Japan
cuis19@mails.tsinghua.edu.cn jingfeng.zhang@riken.jp
xuelang.lj@alibaba-inc.com bhanml@comp.hkbu.edu.hk
sugi@k.u-tokyo.ac.jp zcs@mail.tsinghua.edu.cn

Abstract

Learning adversarially robust models requires invariant predictions to a small neighborhood of its natural inputs, often encountering *insufficient model capacity*. There is research showing that learning multiple sub-models in an *ensemble* could mitigate this insufficiency, further improving the generalization and the robustness. However, the ensemble's voting-based strategy excludes the possibility that *the true predictions remain with the minority*. Therefore, this paper further improves the ensemble through a *collaboration* scheme—Synergy-of-Experts (SoE). Compared with the voting-based strategy, the SoE enables the possibility of correct predictions even if there exists a single correct sub-model. In SoE, every sub-model fits its specific vulnerability area and reserves the rest of the sub-models to fit other vulnerability areas, which effectively optimizes the utilization of the model capacity. Empirical experiments verify that SoE outperforms various ensemble methods against white-box and transfer-based adversarial attacks. The source codes are available at `https://github.com/cuis15/synergy-of-experts`.

1 Introduction

Deep models have been widely applied in various real-world applications including high-stakes scenarios (such as in healthcare, finance, and autonomous driving). An increasing concern is whether these models make *adversarially robust* decisions [1, 2]. Recently, there are research revealing that an adversarially robust method requires invariant predictions to a small neighborhood of its natural inputs, thus often encountering insufficient model capacity [3, 4]. This limits the further improvement of robustness and has the undesirable degradation of generalization [5].

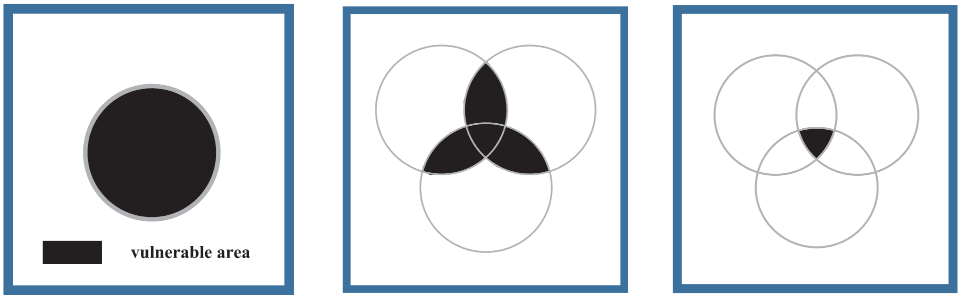
Learning multiple sub-models in an ensemble [6, 7] can mitigate this insufficiency. Remarkably, there are research [8, 9, 10] proposing to minimize the vulnerability overlaps between each pair of sub-models and improving both robustness and generalization over a single model. However, the voting-based ensemble may waste 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*

[*] The first two authors have made equal contributions.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

https://doi.org/10.52202/068431-2359



(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.

predictions remain with the minority. In other words, 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.

Inspired by mixture-of-experts (MoE) [11], we propose to *learn a collaboration* among multiple sub-models to optimally utilize the limited capacity. 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. Besides, learning a collaboration could enable every sub-model to fit its vulnerability area, which could collectively fix broader vulnerability areas. Then, sub-models could collaboratively choose a trustworthy one to make the final predictions.

Classic MoE methods assume that the problem space is separable, and the separability is irrelevant to the learned model [11]. However, the non-i.i.d adversarial inputs [12] depend on the learned models and are hard to be classified by a learned gate in MoE. To tackle the above challenge, we propose Synergy-of-Experts (SoE), which explicitly builds the relationship between learned models and adversarial inputs. Specifically, each sub-model has dual heads: one outputs a vector of predicted probability $f_\theta(\cdot)$; another outputs a scalar that measures the confidence of the prediction. In the adversarial training phase, given an adversarial input x, each sub-model chooses an easy one(s) to feed itself. The other 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, SoE chooses a sub-model with the largest confidence as the representative to output the overall prediction.

We highlight our key contributions as follows.

- We provide a new perspective on learning multiple sub-models for defending against adversarial attacks. We show that the collaboration could make better decisions than the ensemble (Proposition 1), which implies collaboration may fix broader vulnerability areas.
- We propose a novel collaboration framework—SoE (see Section 3.2). In the training phase, SoE minimizes the vulnerability overlap of all sub-models; In the inference phase, SoE could effectively choose a representative sub-model to make correct predictions. We also provide a comprehensive analysis illustrating the rationale of SoE in Appendix.
- Empirical experiments corroborate the SoE outperforms various ensemble methods [10, 8, 9, 13] against white-box and transfer attacks.

2 Related Work

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 [1, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26] and black-box attacks in which the deep models are treated as black boxes to the adversary [27, 28, 29, 30, 31, 32, 33, 34, 35, 36]. 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 [37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49]. *Empirical defense.* leverages adversarial data to build effective defense such as *adversary detection* [50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68] 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 [5, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 3], fixing AT's undesirable robust overfitting [90, 91, 92], improving AT's training efficiency [93, 94, 95, 96, 97, 98], understanding/interpreting AT's unique traits [99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 49, 109, 110], applying AT into applications [111, 112], etc. Besides, researchers have also actively investigated robust-structured models [113, 114, 115, 116, 117, 118, 119]. 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 [6] and boosting [7] have been investigated for significantly improving the model's generalization. Motivated by the benefits of ensemble methods in improving generalization, researchers introduced an ensemble to improve the model robustness [10, 9, 8, 120]. Tramèr *et al.* [120] proposed to reduce the adversarial transferability by training a single model with adversarial examples from multiple pretrained sub-models. Pang *et al.* [8] introduce a regularization method—ADP—to encourage high diversity in the non-maximal predictions of sub-models. Kariyappa *et al.* [9] improved the ensemble diversity by maximizing the introduced cosine distance between the gradients of sub-models with respect to the input. Yang *et al.* [10] proposed to distill non-robust features in the input and diversify the adversarial vulnerability. These methods reduced overlaps of vulnerability areas between sub-models [10]. Compared with voting strategy in ensemble, mixture-of-experts (MoE) assumes that the problem space can be divided into multiple sub-problems through a gate module [11, 121]. However, in adversarial training, the adversarial samples, which depend on the learned models, are not i.i.d. A vanilla MoE is hard to identify the best performing sub-models for each adversarial sample without the information about the learned models.

3 Collaboration to Defend Against Adversarial Attacks

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 $\mathrm{ensemble}(x, f_{\theta_1}, ..., f_{\theta_M})$ by the voting-based strategy:

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

where $\mathbb{1}$ is the indicator function and K denotes the number of classes. 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:*

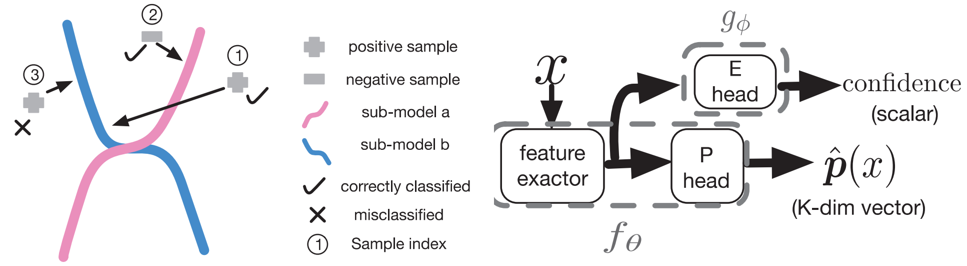
$$f_{\theta_{\mathrm{best}}}(x) = \arg \min_{f_{\theta_i}\in\left\{f_{\theta_1},...,f_{\theta_M}\right\}} \ell(f_{\theta_i}(x), y). \tag{2}$$

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

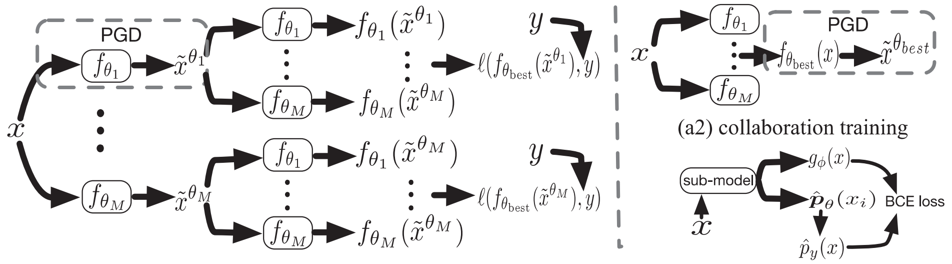
$$\mathrm{collaboration}(x, f_{\theta_1}, ..., f_{\theta_M}) = f_{\theta_{\mathrm{best}}}(x). \tag{3}$$

https://doi.org/10.52202/068431-2359



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

Figure 2: (a) The blue and pink lines denote the decision boundaries of 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—*P head* that outputs the predicted probability (vector) and *E head* approximates the predicted probability of the true label in the prediction (scalar).



PGD
x
f_{θ_1}
$\tilde{x}^{\theta_1}$
f_{θ_M}
$\tilde{x}^{\theta_M}$
$f_{\theta_1}(\tilde{x}^{\theta_1})$
$f_{\theta_M}(\tilde{x}^{\theta_M})$
y
$\ell(f_{\theta_{\text{best}}}(\tilde{x}^{\theta_1}), y)$
$f_{\theta_1}(\tilde{x}^{\theta_M})$
$\ell(f_{\theta_{\text{best}}}(\tilde{x}^{\theta_M}), y)$
PGD
$f_{\theta_{\text{best}}}(x)$
$\tilde{x}^{\theta_{best}}$
sub-model
$g_\phi(x)$
$\hat{\boldsymbol{p}}_\theta(x_i)$
BCE loss
$\hat{p}_y(x)$

(a2) collaboration training

(a1) optimizing the module f_θ (sub-model training) (b) optimizing g_ϕ

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

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's prediction is correct, there exists a case that the majority of sub-models make 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 could achieve an equal or higher performance than an ensemble. Compared with the ensemble, the collaboration requires the identification of the best-performing sub-models using label information. Next, we will introduce a realization of our collaboration framework.

3.2 Realization of collaboration for defending against adversarial attacks

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 radius $\epsilon > 0$ centered at x. To search for adversarial data within norm ball $\mathcal{B}_\epsilon[x]$, [5] 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., $\hat{\boldsymbol{p}}(x) = [\hat{p}_1(x), ..., \hat{p}_K(x)]$.

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 predictions.

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"

https://doi.org/10.52202/068431-2359

Algorithm 1 training phase I: the sub-model training

```
Input: the sub-models with dual heads $\{f_{\theta_i}\}_{i=1}^{M}$ and $\{g_{\phi_i}\}_{i=1}^{M}$, where $f_{\theta_i}$ outputs the label prediction and $g_{\phi_i}$ outputs the approximated confidence, the training dataset $D$, and the hyperparameter $\sigma$
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 $f_{\theta_i}$ using the PGD method;
4:         for each sub-model $f_{\theta_j}, j = 1, 2, ..., M$ do
5:             Calculate the approximated confidence, 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 the module $g_{\phi_j}$;
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:         end for
9:         Calculate surrogate loss on data $\tilde{x}^{\theta_i}$: $\hat{\ell}_m = -\sigma \ln \sum_{j=1}^{M} \exp\left(\frac{-\ell\left(f_{\theta_j}\left(\tilde{x}^{\theta_i}\right), y\right)}{\sigma}\right)$;
10:        Update $\{f_{\theta_i}\}_{i=1}^{M}$ by minimizing $\hat{\ell}_m$. //choose the best-performing sub-model to fit $\tilde{x}^{\theta_i}$
11:    end for
12: end for
13: Output: the learned sub-models with dual heads $\{f_{\theta_i}\}_{i=1}^{M}$ and $\{g_{\phi_i}\}_{i=1}^{M}$.
```

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. 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 by 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) *predictor* (P) head: f_θ predicts the label probability $f_\theta(x) = \hat{\boldsymbol{p}}(x) = [\hat{p}_1(x), ..., \hat{p}_K(x)]$ (a vector); 2) *evaluator* (E) head: g_ϕ approximates the confidence of the prediction $\hat{\boldsymbol{p}}(x)$.

Note that we use g_ϕ to approximate the true label probability $\hat{p}_y(x)$, which denotes the true confidence of a given prediction $\hat{\boldsymbol{p}}(x)$. Meanwhile, the largest confidence corresponds to the lowest objective loss, and vice versa (see theoretical proof in Proposition 2). Therefore, the best-performing sub-models could be identified using the approximated confidence ($g_\phi(x)$) by the E head.

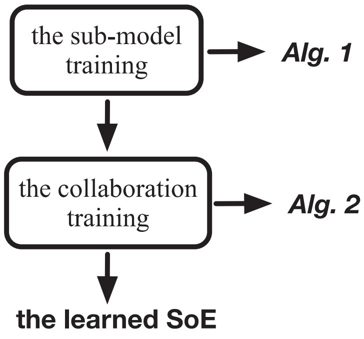


Figure 4: Training process of SoE.

To defend against adversarial attacks, the collaboration needs to learn from adversarial data. Algorithm 1,2 along with Figure 3 articulates how to learn such the collaboration. In particular, the training of our framework SoE shown in Figure 4 is as follows:

1. the sub-model training: the sub-models fit adversarial samples from their own or other sub-models;
2. the collaboration training: the sub-models fit adversarial samples from the collaboration.

The sub-model training. As shown in Figure 3(a1), given a 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 as existing baselines do. 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 1–8 and Lines 9–10 in Algorithm 1. Note that in Line 9, we use a surrogate loss to approximate the sub-model assignment process (reasons see Eq. 5 and 6). 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

Algorithm 2 training phase II: the collaboration training

```
Input: sub-models with dual heads $\{f_{\theta_i}\}_{i=1}^{M}$ and $\{g_{\phi_i}\}_{i=1}^{M}$, where $f_{\theta_i}$ outputs the label prediction and $g_{\phi_i}$ outputs the approximated confidence, training dataset $D$, hyperparameter $\sigma$;
for each data $(x, y) \in D$ do
    for each sub-model $f_{\theta_i}, i = 1, 2, ..., M$ do
        Calculate the approximated confidence, i.e., $g_{\phi_i}(x)$;
        Calculate the prediction i.e., $f_{\theta_i}(x)$;
    end for
    Output the prediction $\hat{\boldsymbol{p}}'(x)$ with the highest confidence;
    Obtain $\tilde{x}$ by perturbing $x$ to worsen the prediction; // generate the adversarial samples of the collaboration
    Minimize BCE loss $\ell_{\phi}(\tilde{x})$ to update the module $g_{\phi}$ of all sub-models;
    Update $\{f_{\theta_i}\}_{i=1}^{M}$ to fit $\tilde{x}$ by minimizing the surrogate loss $\hat{\ell}_m(\tilde{x})$;
end for
Output: the learned sub-models $\{f_{\theta_i}\}_{i=1}^{M}$ with $\{g_{\phi_i}\}_{i=1}^{M}$.
```

the predicted label probability on the true label (i.e., $\hat{p}_y(\tilde{x}^{\theta_i})$) and the approximated confidence (i.e., $g_{\phi}(\tilde{x}^{\theta_i})$) to update each sub-model's E head. This process corresponds to Lines 5–6 in Algorithm 1.

The collaboration training. During the sub-model training, we learn the most adversarial data from each sub-model using the sub-models performing best. The most adversarial samples cannot attack the collaboration successfully. However, there may exist harmful samples (which may not be the most adversarial for any sub-model) that are unexplored and can attack all sub-models. Therefore, the sub-model training may converge without a full exploration of the adversarial samples. (The experimental verification about this could be found in Appendix.) To defend these potential adversarial samples, we propose the collaboration training shown in Figure 3 (a2) and Algorithm 2. Firstly, we propose to generate the adversarial samples that can worsen the outputs of the collaboration. In particular, for each data sample x, we output the prediction $\hat{\boldsymbol{p}}(x)$ whose confidence $g_{\phi}(x)$ is the highest. We perturb x to fool the prediction using PGD method. Then we minimize a surrogate loss to fit this adversarial data $\tilde{x}$. This process is detailly shown in Algorithm 2. We use Algorithm 1 and Algorithm 2 in sequence to train our collaboration. Algorithm 2 is proposed to explore the adversarial samples which could be not the most adversarial samples of any sub-model, but could fool all sub-models. In Algorithm 2, we attack the collaboration iteratively to obtain the adversarial samples. However, in each iteration we update the adversarial sample $\tilde{x}$, the best-performing sub-model could be different. For example, in the first iteration, given the input x, the best-performing sub-model is f_{θ_1}. We obtain an adversarial sample $\tilde{x}'$ by attacking f_{θ_1}. However, in the second iteration, given the input $\tilde{x}'$, the best-performing sub-model is another sub-model (e.g., f_{θ_2}). We attack the best-performing sub-model f_{θ_2} to obtain the adversarial sample $\tilde{x}''$. Therefore, by attacking the collaboration following Algorithm 2, we could obtain the adversarial sample $\tilde{x}''$ which is not the most adversarial samples but could fool all sub-models and is unseen in Algorithm 1.

During the inference phase shown in Algorithm 3 in Appendix, once M sub-models are properly learned, SoE chooses a representative sub-model whose confidence is the highest among all sub-models, and then outputs this sub-model's prediction.

3.3 Analyses of SoE

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: $\min_{\theta_i \in \{\theta_1,...,\theta_M\}} \ell(f_{\theta_i}(\tilde{x}, y)) > \delta, \quad \tilde{x} \in \tilde{D}$, where $\tilde{D}$ denotes the vulnerability overlap of all sub-models.

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.,

$$\begin{aligned} &\min_{\{\theta_1,\theta_2,\ldots,\theta_M\}} \mathbb{E}_{(x,y)\in D}\left(\mathbb{E}_{i\in\{1,2,\ldots,M\}}\,\ell_{best}(\tilde{x}^{\theta_i},y)\right) \\ &\text{where} \quad \ell_{best}(\tilde{x}^{\theta_i},y) = \min_{j\in\{1,2,\ldots,M\}} \ell\left(f_{\theta_j}(\tilde{x}^{\theta_i}),y\right) \end{aligned} \tag{5}$$

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.(5) may cause a trivial solution (e.g., there is only one optimized sub-model), for ease of the optimization of Eq.(5) (Corresponding to Lines 9–10 in Algorithm 1), we provide a surrogate objective as follows.

$$\min\ {}_{\{\theta_1,\theta_2,\ldots,\theta_M\}}\ \mathbb{E}_{(x,y)\in D}\left(\mathbb{E}_{i\in\{1,2,\ldots,M\}}\hat{\ell}_m(\tilde{x}^{\theta_i},y)\right), \tag{6}$$

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.(6), 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

$$\min_{j\in\{1,2,\ldots,M\}} \ell\left(f_{\theta_j}(\tilde{x}^{\theta_i}),y\right) - \delta\cdot\ln(M) \leq \hat{\ell}_m(\tilde{x}^{\theta_i}),y) \leq \min_{j\in\{1,2,\ldots,M\}} \ell\left(f_{\theta_j}(\tilde{x}^{\theta_i}),y\right). \tag{7}$$

The proof of Eq.(7) is in Appendix.

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

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

$$\arg\max_{j\in\{1,2,\ldots,M\}} \text{confidence}(f_{\theta_j}(x)) = \arg\min_{j\in\{1,2,\ldots M\}} \ell(f_{\theta_j}(x),y). \tag{8}$$

where $\mathrm{confidence}$ corresponds to the probability of the true label in a given $\hat{\boldsymbol{p}}(x)$. To learn the E head to approximate the confidence of $\hat{\boldsymbol{p}}(x)$, we could simply compare its output with the predicted probability on the true label (i.e.,$\hat{p}_y(x)$), then update the E head by gradient descent (corresponds to Lines 6–7 in Algorithm 1). Given an input to the collaboration, our dual-head structured sub-models can collaboratively decide the best-performing sub-model by comparing the values of the E head (corresponds to Algorithm 3 in Appendix).

Note that the E head may be susceptible to adversarial attacks in the white-box setting. In our implementation, we use a simple linear structure to regress the confidence. Experimental results on adaptive attacks demonstrate the reliability of our framework.

4 Experiments

In this section, we conduct experiments on a benchmark dataset to verify the effectiveness of our method in defending against white-box and transfer attacks. Then we provide ablation studies to demonstrate the significance of the collaboration training described in Algorithm 2. The experimental results about black-box attacks and more discussions could be found in Appendix.

4.1 Experimental Setup

Following the work in [10], we compare our method with various related methods, including ADP [8], GAL [9], DVERGE [10] and MoRE [13]. We use ResNet-20 [122] as sub-models in all methods for fair comparisons, and we use CIFAR10 as the data set, a classical image dataset [123] that has 50,000 training images and 10,000 test images.

4.2 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.

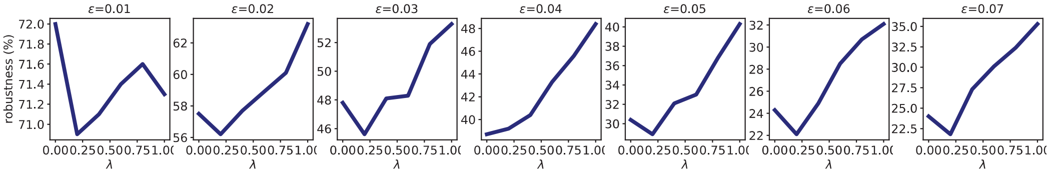


Figure 5: SoE robustness with varying λ under adaptive attacks with $\epsilon \in \{0.01, 0.02, ..., 0.07\}$.

We compare our method with 4 baselines on defending against white-box attacks using a subset of **CIFAR10**. We use 50-step PGD with five random starts and the step size of $\epsilon/5$ to attack all methods as in [10]. We learn the models in adversarial training and evaluate the robustness under various attacks with the same ϵ. For example, to evaluate the robustness under white-box attack with $\epsilon = 0.01$, we first learn the models in adversarial training with $\epsilon = 0.01$. We evaluate all methods following the setting in [10]. In particular, we randomly select 1000 samples under different ϵ. For the PGD attack, we select the cross-entropy loss to update the perturbations to search for adversarial samples. In addition to the robustness, we also report the performance of all methods on clean data with the adversarial training under different ϵ.

Table 1: Robustness and clean data accuracy (%) under white-box attack.

ϵ(robust/clean)	0.01	0.02	0.03	0.04	0.05	0.06	0.07
GAL	49.5/87.8	31.4/85.4	25.4/81.2	22.7/78.7	18.4/77.3	13.4/76.2	9.0/76.0
DVERGE	67.3/85.4	52.3/83.0	41.1/79.7	29.9/77.6	22.5/76.7	14.2/75.8	10.0/75.3
MoRE	67.9/88.0	49.9/85.3	37.8/82.0	31.3/79.5	24.0/78.2	15.6/77.1	12.3/77.8
ADP	67.7/89.0	52.9/86.8	40.8/85.4	30.8/83.3	25.8/76.0	23.4/66.4	20.3/63.0
SoE	**72.0**/88.8	**57.5**/85.6	**47.8**/80.2	**38.7**/80.0	**30.4**/79.1	**24.3**/76.7	**24.0**/74.1
SoE (adaptive)	70.9/88.8	56.2/85.6	45.6/80.2	38.7/80.0	28.9/79.1	22.1/76.7	21.8/74.1

From Table 1, SoE achieves a better robustness performance under white-box attack. The results verify that collaboration significantly improves the utilization of the limited model capacity. Therefore, SoE can fit more adversarial data and have a relatively smaller vulnerable area.

Performance on Adaptive Attacks. We follow the suggestions in [124] and conduct two different adaptive attacks to fool the dual heads simultaneously. For the first adaptive attack, we attack the E head to minimize the confidence of the best-performing sub-model. In particular, we maximize $l_1 = \text{BCE}(g_{\phi_j}(x), 1)$, where $j = \arg\max_{i \in [M]} g_{\phi_i}(x)$ means the j-th sub-model is identified as the best-performing sub-model. For the second adaptive attack, we try to achieve a mismatch between the correct predictions and the highest confidence. Specifically, we maximize $l_2 = -\log\left[f_{\theta_j}(x)_y * g_{\phi_j}(x) + 1 - g_{\phi_j}(x)\right]$, where $j = \arg\max_{i \in [M]} -\log\left[f_{\theta_i}(x)_y * g_{\phi_i}(x) + 1 - g_{\phi_i}(x)\right]$. We conduct experiments by maximizing the weighted loss $\ell_1^{adp} = \ell(f_\theta(x), y) + \lambda \cdot \ell_1$ and $\ell_2^{adp} = \ell(f_\theta(x), y) + \lambda \cdot \ell_2$ with varying λ. The robustness with respect to λ on the stronger adaptive attack (i.e., the attack achieves a higher success rate) is shown in Figure 5. Though both adaptive attacks could attack the predictor and the evaluator simultaneously, the evaluator in our method is robust to the adversarial samples because of its sample structure. As seen in Figure 5 and Table 1, our method is slightly degraded by adaptive attacks and still outperforms baseline models.

Table 2: Transfer attack with 3 adversarial variants (%).

methods \ ϵ	0.01	0.02	0.03	0.04	0.05	0.06	0.07
GAL	64.2$_{\pm 4.2}$	48.7$_{\pm 2.7}$	50.2$_{\pm 3.5}$	49.9$_{\pm 3.2}$	52.3$_{\pm 4.5}$	48.7$_{\pm 3.2}$	42.2$_{\pm 4.1}$
ADP	**85.6**$_{\pm .2}$	82.9$_{\pm .2}$	78.3$_{\pm .3}$	73.2$_{\pm .1}$	69.6$_{\pm .2}$	60.4$_{\pm .2}$	57.4$_{\pm .1}$
MoRE	84.8$_{\pm .3}$	82.1$_{\pm .1}$	78.4$_{\pm .2}$	74.3$_{\pm .1}$	73.2$_{\pm .1}$	70.3$_{\pm .2}$	69.1$_{\pm .3}$
DVERGE	83.4$_{\pm .3}$	80.1$_{\pm .2}$	77.3$_{\pm .1}$	72.4$_{\pm .1}$	71.9$_{\pm .2}$	68.8$_{\pm .3}$	66.2$_{\pm .2}$
SoE	85.2$_{\pm .1}$	**83.4**$_{\pm .1}$	**78.8**$_{\pm .1}$	**76.6**$_{\pm .2}$	**74.6**$_{\pm .1}$	**72.3**$_{\pm .2}$	**70.2**$_{\pm .2}$

4.3 Visualization of the our Collaboration Scheme

To intuitively understand the collaboration mechanism of our proposed SoE, we show the decision boundaries of the ensemble and the collaboration in Figure 6. In particular, we learn the ensemble and the collaboration with 3 ResNet-20 sub-models on CIFAR10 dataset with adversarial training.

https://doi.org/10.52202/068431-2359

Table 3: Transfer attack with 30 adversarial variants (%).

methods \ ϵ	0.01	0.02	0.03	0.04	0.05	0.06	0.07
GAL	57.8$_{\pm 3.2}$	64.1$_{\pm 3.7}$	46.3$_{\pm 2.8}$	56.0$_{\pm 3.0}$	43.9$_{\pm 3.1}$	44.5$_{\pm 2.9}$	41.4$_{\pm 3.2}$
ADP	**84.2**$_{\pm .2}$	80.2$_{\pm .2}$	73.7$_{\pm .1}$	69.4$_{\pm .2}$	65.2$_{\pm .2}$	56.7$_{\pm .2}$	54.4$_{\pm .2}$
MoRE	83.7$_{\pm .3}$	79.6$_{\pm .2}$	74.4$_{\pm .2}$	70.2$_{\pm .3}$	67.6$_{\pm .1}$	63.8$_{\pm .3}$	59.3$_{\pm .2}$
DVERGE	81.5$_{\pm .3}$	78.1$_{\pm .3}$	73.5$_{\pm .3}$	68.4$_{\pm .1}$	67.2$_{\pm .2}$	63.8$_{\pm .1}$	57.1$_{\pm .2}$
SoE	83.1$_{\pm .3}$	**80.4**$_{\pm .2}$	**75.1**$_{\pm .3}$	**70.8**$_{\pm .2}$	**69.0**$_{\pm .2}$	**64.0**$_{\pm .3}$	**61.1**$_{\pm .2}$

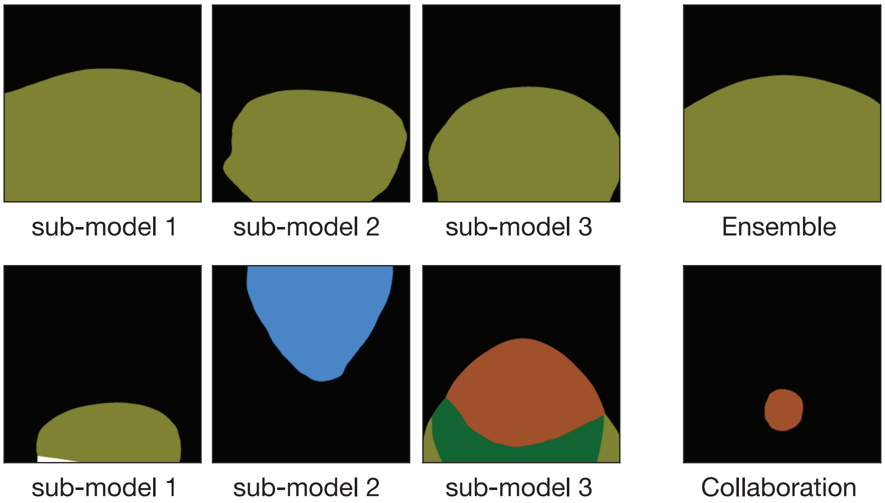


Figure 6: The visualization of the decision boundaries of the ensemble and the collaboration. The same color areas denotes the same predicted label. The black regions means that the models predict correctly while other color regions means that the models are fooled and predict incorrectly. The vertical axis is along the adversarial direction and the horizontal axis is along a random Rademacher vector.

The same color areas in Figure 6 denotes the same predicted label. The black regions means that the models predict correctly while other color regions means that the models are fooled and predict incorrectly. The ensemble defending against attacks requires more than half of the sub-models fit the same vulnerability areas. From the top of Figure 6, all sub-models fit similar vulnerability areas and there is a boarder vulnerability overlap between sub-models. Our proposed collaboration proposes to minimize the overlap of all sub-models. From the bottom of Figure 6, different sub-models defend against different vulnerability areas collaboratively and our collaboration achieves a smaller vulnerability areas.

4.4 Performance on Transfer Attack

Due to the transferability of adversarial examples, transfer adversaries can craft adversarial examples based on surrogate models and perform an attack on the target model. In our experiments, we follow the transfer attack setting in [10] and select 1000 test samples randomly. We use hold-out baseline ensembles with three ResNet-20 sub-models as the surrogate models to generate adversarial samples. In particular, we use three attack methodologies: PGD with momentum [35], SGM [29] which adds weight to the gradient through the skip connections of the model, and M-FGSM [36] which randomly augments the input images in each step. For each sample, three adversarial variants are using the three attack methods. Only when the model can classify all kinds of adversarial variants can the model successfully defend against adversarial attacks. We show the results of all methods in Table 2. Furthermore, we also use a more challenging setting following the work in [10]. We use hold-out baseline models with 3, 5, and 8 ResNet-20 sub-models as the surrogate models. Meanwhile, we generate adversarial samples with cross-entropy loss and CW loss [15]. For each sample, we generate 30 adversarial variants, and only if the model classifies all the 30 variants can the model defend the transfer attack successfully. The results are shown in Table 3.

In our experiments, GAL is hard to learn stably in adversarial training. In Table 2, when $0.01 \leq \epsilon$, SoE achieves a better performance compared with baselines. With the increase of ϵ, the volume

https://doi.org/10.52202/068431-2359

of ϵ-ball increases exponentially. The performances of all methods get worse significantly because of insufficient model capacity. Similar results can also be found in Table 3. Since SoE addresses more adversarial data using a collaboration mechanism, it achieves a relatively better robustness performance as ϵ increases.

4.5 The Robustness of SoE under Different Number of Sub-models

To explore the robustness of the collaboration under different ϵ with different number of sub-models, we conduct experiments under transfer attacks with different number of sub-models ($1 \leq N \leq 5$). The detailed information could be found in Section 4.3 in Appendix. In summary, we have the following findings. For different ϵ, more sub-models could achieve a more significant robustness improvement with the increase of ϵ. For different number of sub-models, more sub-models are more likely to achieve a higher robustness, but the margin gain decreases with more sub-models.

5 Conclusion

In this paper, we study an essential question in the field of adversarial attacks 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—SoE—as the defense strategy over ensemble methods, and empirical experiments indeed verified the efficacy of SoE. Future work includes applying our collaboration framework to other areas such as kernel methods, fairness, and federated model, etc.

Acknowledgments

We would like to thank the anonymous reviewers of NeurIPS 2022 for their constructive comments. Sen Cui and Changshui Zhang would like to acknowledge the funding by the Natural Science Fundation of China(NSFC. No. 62176132). Bo Han was supported by the RGC ECS No. 22200720, NSFC YSF No. 62006202, Guangdong Basic and Applied Basic Research Foundation No. 2022A1515011652, and RIKEN Collaborative Research Fund. Jingfeng Zhang was supported by JST ACT-X Grant Number JPMJAX21AF and JSPS KAKENHI Grant Number 22K17955, Japan. Masashi Sugiyama was supported by JST AIP Acceleration Research Grant Number JPMJCR20U3 and the Institute for AI and Beyond, UTokyo, Japan.

References

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

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

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

[4] 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*, 2021.

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

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

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

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

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

[10] 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*, 2020.

https://doi.org/10.52202/068431-2359

[11] 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.

[12] Ruize Gao, Feng Liu, Jingfeng Zhang, Bo Han, Tongliang Liu, Gang Niu, and Masashi Sugiyama. Maximum mean discrepancy test is aware of adversarial attacks. In *ICML*, 2021.

[13] Kaidi Xu, Chenan Wang, Hao Cheng, Bhavya Kailkhura, Xue Lin, and Ryan Goldhahn. Mixture of robust experts (more): A robust denoising method towards multiple perturbations. *arXiv preprint arXiv:2104.10586*, 2021.

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

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

[16] 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.

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

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

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

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

[21] 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.

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

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

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

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

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

[27] 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.

[28] 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.

[29] 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*, 2020.

[30] 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.

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

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

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

https://doi.org/10.52202/068431-2359

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

[35] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, 2018.

[36] 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 *CVPR*, 2019.

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

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

[39] 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.

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

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

[42] 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.

[43] 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.

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

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

[46] 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*, 2020.

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

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

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

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

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

[52] 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*, 2017.

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

[54] 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*, 2018.

[55] 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.

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

https://doi.org/10.52202/068431-2359

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

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

[59] 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.

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

[61] 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.

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

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

[64] Kejiang Chen, Yuefeng Chen, Hang Zhou, Chuan Qin, Xiaofeng Mao, Weiming Zhang, and Nenghai Yu. Adversarial examples detection beyond image space. In *ICASSP*, 2021.

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

[66] 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.

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

[68] 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.

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

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

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

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

[73] 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.

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

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

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

[77] 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*, 2019.

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

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

[80] Haichao Zhang and Jianyu Wang. Defense against adversarial attacks using feature scattering-based adversarial training. In *NeurIPS*, 2019.

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

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

[83] 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.

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

[85] 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*, 2020.

[86] 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*, 2020.

[87] 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.

[88] 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.

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

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

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

[92] Jingfeng Zhang, Xilie Xu, Bo Han, Tongliang Liu, Lizhen Cui, Gang Niu, and Masashi Sugiyama. Noilin: Improving adversarial training and correcting stereotype of noisy labels. *Transactions on Machine Learning Research*, 2022.

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

[94] 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.

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

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

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

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

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

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

[101] 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.

[102] 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.

https://doi.org/10.52202/068431-2359

[103] 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*, 2019.

[104] 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.

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

[106] 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*, 2020.

[107] Yi Zhang, Orestis Plevrakis, Simon S. Du, Xingguo Li, Zhao Song, and Sanjeev Arora. Over-parameterized adversarial training: An analysis overcoming the curse of dimensionality. In *NeurIPS*, 2020.

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

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

[110] 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.

[111] Hanshu Yan, Jingfeng Zhang, Jiashi Feng, Masashi Sugiyama, and Vincent Y. F. Tan. Towards adversarially robust deep image denoising. In *IJCAI*, 2022.

[112] Xilie Xu, Jingfeng Zhang, Feng Liu, Masashi Sugiyama, and Mohan Kankanhalli. Adversarial attack and defense for non-parametric two-sample tests. In *ICML*, 2022.

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

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

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

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

[117] 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*, 2021.

[118] 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 *ICML*, 2021.

[119] Tianyu Pang, Huishuai Zhang, Di He, Yinpeng Dong, Hang Su, Wei Chen, Jun Zhu, and Tie-Yan Liu. Two coupled rejection metrics can tell adversarial examples apart. In *CVPR*, 2022.

[120] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *ICLR*, 2018.

[121] 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*, 2018.

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

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

[124] Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

Checklist

1. For all authors...
   - (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] We discuss it in Appendix
   - (b) Did you describe the limitations of your work? [Yes] The limitations are in Appendix
   - (c) Did you discuss any potential negative societal impacts of your work? [Yes] The societal impacts are in Appendix
   - (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   - (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   - (b) Did you include complete proofs of all theoretical results? [Yes] The proofs are in Appendix.
3. If you ran experiments...
   - (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The source codes are made publicly available.
   - (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] The training details are in Appendix.
   - (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   - (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] The devices of computing are in Appendix.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   - (a) If your work uses existing assets, did you cite the creators? [Yes]
   - (b) Did you mention the license of the assets? [Yes]
   - (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   - (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes]
   - (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [Yes] We discuss our used data in Section 4 and Appendix.
5. If you used crowdsourcing or conducted research with human subjects...
   - (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   - (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   - (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

https://doi.org/10.52202/068431-2359