Synergy-of-Experts: Collaborate to Improve Adversarial Robustness

Sen Cui1*, Jingfeng Zhang2*, Jian Liang3, Bo Han4, Masashi Sugiyama2,5, Changshui Zhang1

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.
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—$f_{\theta}(\cdot)$ (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 [11, 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., $\ell_2$ and $\ell_\infty$) 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][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20], improving AT’s training efficiency [21][22][23][24][25][26][27][28][29][30], fixing AT’s undesirable robust overfitting [31][32][33][34][35][36], enhancing AT’s unique traits [37][38][39][40][41][42][43][44][45][46][47][48][49][50][51][52][53][54][55][56][57][58][59][60][61][62][63][64][65][66][67][68][69][70][71][72][73][74][75][76][77][78][79][80][81][82][83][84][85][86][87][88][89][90][91][92], improving AT’s training efficiency [93][94][95][96][97][98], understanding/interpreting AT’s 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 $\{f_{\theta_1}(x), f_{\theta_2}(x), ..., f_{\theta_M}(x)\}$. The ensemble outputs a final prediction $\text{ensemble}(x, f_{\theta_1}, ..., f_{\theta_M})$ by the voting-based strategy:

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

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:

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

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 $\{f_{\theta_1}(x), f_{\theta_2}(x), ..., f_{\theta_M}(x)\}$. The collaboration tries to output a final prediction $\text{collaboration}(x, f_{\theta_1}, ..., f_{\theta_M})$ by the best-performing sub-model:

$$\text{collaboration}(x, f_{\theta_1}, ..., f_{\theta_M}) = f_{\theta_{\text{best}}}(x).$$
To search for adversarial data within norm ball \( D \), if the ensemble’s prediction is correct, at least one sub-model makes 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-model to make predictions. 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 \( \mathcal{Y} = \{1, \ldots, K\} \) 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 \( B_{\epsilon}[x] = \{x' \in \mathcal{X} \mid d_{\text{inf}}(x, x') \leq \epsilon\} \) is the closed ball of radius \( \epsilon > 0 \) centered at \( x \). To search for adversarial data within norm ball \( B_{\epsilon}[x] \), \(^3\) 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., \( \hat{p}(x) = [\hat{p}_1(x), \ldots, \hat{p}_K(x)] \).

**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"
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) Data 3 is misclassified by both sub-models. The blue sub-model is near Data 3 and has the lowest objective loss. We assign the blue sub-model to fit Data 3, because the collaboration can correctly be classified Data 2 by selecting the pink model as the representative. ii) Data 1 is correctly classified by both models. The blue model is far from Data 1 and takes the lowest effort on fitting it; therefore, we assign the blue model to fit Data 1.

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 \( \hat{y}(x) = \hat{p}(x) = [\hat{p}_1(x), ..., \hat{p}_K(x)] \) (a vector); 2) evaluator (E) head: \( g_θ \) approximates the confidence of the prediction \( \hat{p}(x) \).

Note that we use \( g_θ \) to approximate the true label probability \( \hat{p}_j(x) \), which denotes the true confidence of a given prediction \( \hat{p}(x) \). 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 \( g_θ(x) \) by the E head.

To defend against adversarial attacks, the collaboration needs to learn from adversarial data. Algorithm 1 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.

**Algorithm 1** training phase I: the sub-model training

**Input:** the sub-models with dual heads \( \{f_θ_i\}_{i=1}^M \) and \( \{g_θ_i\}_{i=1}^M \), where \( f_θ \) outputs the label prediction and \( g_θ \) outputs the approximated confidence, the training dataset \( D \), and the hyperparameter \( σ \)

1. for each data \((x, y) \in D \) do
2. for each sub-model \( f_θ_i, i = 1, 2, ..., M \) do
3. Obtain the adversarial data \( \tilde{x}_θ^i \) of the sub-model \( f_θ_i \) using the PGD method;
4. for each sub-model \( f_θ_i, j = 1, 2, ..., M \) do
5. Calculate the approximated confidence, i.e., \( g_θ(\tilde{x}_θ^i) \);
6. Minimize BCE loss \( \ell_θ = \text{BCE}(g_θ(\tilde{x}_θ^i), \hat{y}_θ(\tilde{x}_θ^i)) \) to update the module \( g_θ \);
7. Collect sub-model \( i \)'s cross entropy (CE) loss on data \( \tilde{x}_θ^i: \ell \left( f_θ_j(\tilde{x}_θ^i), y \right) \);
8. end for
9. Calculate surrogate loss on data \( \tilde{x}_θ^i: \hat{\ell}_m = -σ\ln \sum_{j=1}^M \exp \left( \frac{-\ell(f_θ_i(\tilde{x}_θ^i), y)}{σ} \right) \);
10. Update \( \{f_θ_i\}_{i=1}^M \) by minimizing \( \hat{\ell}_m \). //choose the best-performing sub-model to fit \( \tilde{x}_θ^i \)
11. end for
12. end for
13. Output: the learned sub-models with dual heads \( \{f_θ_i\}_{i=1}^M \) and \( \{g_θ_i\}_{i=1}^M \).
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 \( \sigma \);

1: for each data \((x, y) \in D \) do
2: for each sub-model \( f_{\theta_i}, i = 1, 2, ..., M \) do
3: Calculate the approximated confidence, i.e., \( g_{\phi_i}(x) \);
4: Calculate the prediction i.e., \( f_{\theta_i}(x) \);
5: end for
6: Output the prediction \( \tilde{p}(x) \) with the highest confidence;
7: Obtain \( \tilde{x} \) by perturbing \( x \) to worsen the prediction; // generate the adversarial samples of the collaboration
8: Minimize BCE loss \( \ell_{\phi}(\tilde{x}) \) to update the module \( g_{\phi} \) of all sub-models;
9: Update \( \{f_{\theta_i}\}_{i=1}^M \) to fit \( \tilde{x} \) by minimizing the surrogate loss \( \ell_{\eta}(\tilde{x}) \);
10: end for
11: Output: the learned sub-models \( \{f_{\theta_i}\}_{i=1}^M \) with \( \{g_{\phi_i}\}_{i=1}^M \).

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 \( \hat{p}(x) \) 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 detailed 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 \( f_{\theta_1} \). We obtain an adversarial sample \( \tilde{x}' \) by attacking \( f_{\theta_1} \). However, in the second iteration, given the input \( \tilde{x}' \), the best-performing sub-model is another sub-model (e.g., \( f_{\theta_2} \)). We attack the best-performing sub-model \( f_{\theta_2} \) 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 \( \delta \): \( \min_{\theta_i \in \{\theta_1, ..., \theta_M\}} \ell(\theta_i, (\tilde{x}, y)) > \delta \), \( \tilde{x} \in \bar{D} \), where \( \bar{D} \) 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 equivalent to minimizing the loss of the best-performing sub-model, i.e.,

\[
\min_{\{\theta_1, \theta_2, \ldots, \theta_M\}} \mathbb{E}_{(x,y) \in D} \left( \mathbb{E}_{i \in \{1,2,\ldots,M\}} \ell_{\text{best}}(\tilde{x}^{\theta_i}, y) \right) 
\]

where \( \tilde{x}^{\theta_i} \) is the adversarial data generated by the sub-model \( \theta_i \).

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.

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

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

In Eq. (6), we approximate the objective \( \min_{\theta_j \in \{\theta_1, \theta_2, \ldots, \theta_M\}} \ell(f_{\theta_j}(\tilde{x}^{\theta_i}), y) \) using a smooth gating maximum function as follows

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

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.,

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

where confidence corresponds to the probability of the true label in a given \( \hat{p}(x) \). To learn the E head to approximate the confidence of \( \hat{p}(x) \), 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 1. 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.
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 $\epsilon$. 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 $\epsilon$. 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 $\epsilon$.

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 $\ell^2_{\text{dp}} = \ell(f_0(x), y) + \lambda \cdot \ell_1$ and $\ell^2_{\text{dp}} = \ell(f_0(x), y) + \lambda \cdot \ell_2$ with varying $\lambda$. The robustness with respect to $\lambda$ 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.

### 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.
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 $\epsilon$, the volume...
of $\epsilon$-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 $\epsilon$ increases.

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

To explore the robustness of the collaboration under different $\epsilon$ 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 $\epsilon$, more sub-models could achieve a more significant robustness improvement with the increase of $\epsilon$. 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 Foundation 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


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


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]