Inclusion-Exclusion Enhanced Ensemble Adversarial Training

Henry Shugart Yeshashvi Munagala Judy Chao Mikhal Ben-Joseph
University of North Carolina at Chapel Hill
Department of Statistics and Operations Research
{hshugart, munagala}@email.unc.edu
{judychao, mikbenjo}@live.unc.edu

Abstract

Adversarial training is a state-of-the-art technique used to enhance the robustness of deep learning models against testing-time attacks in the form of perturbed images designed to fool target models. Ensemble Adversarial Training, originally proposed by Tramer et al., creates adversarial images utilizing min-max optimization over a probability simplex on multiple models simultaneously. Due to the transference property of adversarial images, this process tends to select images in the shared adversarial perturbation space of the models. This paper describes a novel application of Alternating Projected Gradient Descent Ascent method to find adversarial examples in the non-overlapping perturbation spaces of the various models. On CIFAR-10, our method yielded greater robustness against both white box and black box attacks, as well as non-perturbed images, than traditional Adversarial Training and Ensemble Adversarial Training. Code for the project can be found at `https://github.com/mikhalbj/ModEnsAdvTrain.git`

1 Introduction

As Deep Neural Networks (DNNs) become increasingly prevalent across a wide array of societal domains, adversarial attacks against them are a gradually more relevant threat. Adversarial attacks consist of perturbed image inputs that are impossible for a human to distinguish from the original and are deployed with the goal of fooling the model into incorrectly labeling the image.

Readying models to defend against these attack consists of creating adversarial neighbors and training the model to ignore these neighbors' “tricks" so that it correctly labels perturbed data. There are two main ways to create such adversarial examples: white box attacks, which can access the model parameters and perform effectively on a small scale, and black box attacks, which do not have any information about the model's architecture and create adversarial examples based on an arbitrary external model and the target model's output labels alone. It has been shown that certain types of black box-based training can be scaled well, but only a limited number of “single-step" attacks have been explored through this method. Moreover, if a certain black-box example generator model does not fit the true models' loss function well, the adversary may optimize to a degenerate minimum, which in turn can lead to overfitting and potentially poor outcomes on new models, as explained in [3].

Ensemble Adversarial Training (hereby referred to as EAT), as proposed by Tramer, seeks to mitigate some of these problems by training a model with perturbed images transferred from multiple other models. One notable advantage of EAT is its increased robustness against transfer black box attacks from holdout models. Additionally, this method, through its training of enriched data, decouples the adversarial attack generation from the training time of the target model. However, this technique results in decreased robustness to white-box attacks like Fast Gradient Sign Method and Projected

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

Gradient Descent, wherein the source and target models are the same. Another drawback is that EAT also results in a slightly lower accuracy on natural images. Finally, and of most relevance to our project, traditional EAT tends to produce adversarial examples which exist in the shared perturbation space of the models. As explained in Bao et al., a lack of diversity in the adversarial training image collection may inhibit the performance of the adversarially trained model [1].

Our team seeks to remedy this issue by finding adversarial perturbations from the non-overlapping areas in the perturbation spaces of the models. To this end, we utilize a min-max optimized attack method called **A**lternating One-Step **P**rojected **G**radient **D**escent-**A**scent (APGDA) to create modified EAT examples. We compare the robustness of models trained with these examples to three baseline models: a model trained with regular AT, a model trained with traditional EAT, and a model which was not adversarially trained. We hope to gain insight into the effectiveness of our method in defending DNNs against both white box and black box attacks.

In summary, we aimed to simultaneously reap the benefits of EAT and address some of its drawbacks. Our goal was to build on ensemble adversarial training with min-max optimization and inclusion-exclusion methods, as described below.

2 Related Works

In the paper "Ensemble Adversarial Training: Attacks and Defenses," [3] the authors address the weaknesses of adversarial training using basic single-step attack methods such as the fast gradient sign method (FGSM) by proposing a technique called Ensemble Adversarial Training. This technique is more robust to black-box attacks because it augments training data with perturbations transferred from other static, pre-trained models. These perturbations crafted from external models act as appropriate approximations with the goal of minimizing risk over the adversarial examples. Together, this strategy is capable of separating the processes of adversarial example generation from the training of the model itself. As a result, a model produced using EAT is projected to defend itself more robustly against its black-box adversaries. However, trade-offs do exist when EAT is applied to clean data. Additionally, it should be noted that models trained with EAT did not fare well against white-box single step attacks, due to the nature of the training process. Regardless, there exists great potential for EAT to increase robustness in black-box settings.

Wang et al. employ min-max optimization to improve the design of different types of adversarial attacks in their paper, "Adversarial Attack Generation Empowered by Min-Max Optimization" [4]. The authors propose a unified alternating one-step projected gradient descent-ascent (APGDA) attack method. At each iteration, only one-step projected gradient descent and one-step projected gradient ascent are completed for outer minimization and inner maximization, respectively. Of the three attack tasks in the paper, the one of primary interest to our team is the generation of adversarial examples that simultaneously fool several models. We aimed to use this min-max approach to improve the Attack Success Rates (ASR) against multiple models; our team applied the APGDA method to the EAT technique in [3].

In 2020, Yang et al. [5] proposed a different method to tackle the lack of diversity in ensemble adversarially-crafted images. The DVERGE process incentivizes diversity in the adversarial perturbation creation process by identifying and isolating vulnerabilities, or non-robust features which are sensitive to noise, within each sub-model from EAT. By altering the loss function to maximize diversity and reduce the likelihood that vulnerabilities are shared across sub-models, the DVERGE method lessens the transferability of attacks. This, in turn, was shown to increase robustness against both white box and black box attacks relative to traditional EAT methods. The conclusions from this work informed our baseline expectations for how our modified EAT method would perform.

3 Proposed Methods

3.1 Notation

This work considers an image classification problem with data $\mathbf{X}_i \in \mathbb{R}^{n\times n}$ and associated classification $Y \in \mathbb{R}^s$. Our work relies heavily on pre-trained models $f : \mathbb{R}^{n\times n} \to \mathbb{R}^s$, which we label as $f_1, f_2, ..., f_m$. We consider the standard m-dimensional simplex $\Delta^m = \{w \mid \mathbf{1}^T w = 1, w \geq 0\}$.

We define the adversarial subspace of a function to be $A(f,(x_0,y_0)) = \{x \mid \arg\max_i\{f(x_0)\} \neq y_0, ||x_0 - x|| \leq \varepsilon\}$.

3.2 Methods

We propose an Inclusion-Exclusion Enhanced Ensemble Adversarial Training method. The performance of EAT is found to be limited due to largely shared adversarial spaces between standardly trained models. Our method attempts to resolve this issue by enhancing the diversity of adversarial examples created for model training. We achieve this by creating adversarial examples which are designed to fool specific subsets of the pre-trained models while ensuring the other pre-trained models are not fooled.

It is shown in [4] that adversarial examples which are able to fool multiple models can be crafted by solving the following min-max problem.

$$\max_{\delta:||\delta||_\infty \leq \varepsilon} \sum_{i=1}^{M} \min_{w \in \Delta^m} w_i \ell_i(f_i(x+\delta), y). \tag{1}$$

We utilize a similar approach to formulate models which trick only a subset of models by considering an alternative loss function for models which are not meant to be fooled. For pre-trained models we attempt to fool we use $\hat{\ell}_i(\delta) = \ell(f_i(X+\delta), Y)$. We consider ℓ to be the standard categorical cross entropy loss function. For models which we attempt to not fool we define

$$\hat{\ell}_i(\delta) = \alpha - \beta\ell(f_i(X+\delta), Y) \quad \text{s.t.} \quad \alpha,\ \beta > 0. \tag{2}$$

Here α, β are constant hyper-parameters selected before the creation of adversarial examples. For our experimentation we used $\alpha = 2.5, \beta = .3$. It is clear that for pre-trained models that are selected to be fooled $\hat{\ell}_i(\delta)$ is maximized by a perturbation δ which maximises the cross entropy loss. For models which are not meant to be fooled $\hat{\ell}_i(\delta)$ is maximized when the perturbation δ minimizes cross entropy loss. Maximizing cross entropy loss should lead to adversarial perturbations which cause models to misclassify the data, while minimized cross entropy loss should lead the models to correctly classify the data. We see that for images which are improperly classified by the pre-trained models which we do not intend to fool, the minimization of the cross entropy loss may lead the newly crafted adversarial example to be correctly classified by the model.

Algorithm 1: Inclusion Exclusion Adversarial Generation

```
for $M_F \in \mathcal{P}(\{f_1, f_2, ..., f_m\})\backslash\{\}$ do
    Selecting set of pre-trained models to be fooled
    for $f_i \in \{f_1, f_2, ..., f_m\}$ do
        if $f_i \in M_F$ then
            Define: $\hat{\ell}_i(\delta) = \ell(f_i(X+\delta), Y)$
        else
            Define: $\hat{\ell}_i(\delta) = \alpha - \beta\ell(f_i(X+\delta), Y)$
        end
    end
    Initialize $w^0 = \mathbf{1}/m, \delta^0 = \mathbf{0}^{n\times n}$
    Define: $\hat{\mathcal{L}}(\delta) = \begin{bmatrix} \hat{\ell}_1(\delta) \\ \vdots \\ \hat{\ell}_m(\delta) \end{bmatrix}$
    for $k = 1, ..., k_{\max}$ do
        Inner Min Update: $w^k = \text{proj}_{\Delta^m}(w^{k-1} - \gamma_1\hat{\mathcal{L}}(\delta^{k-1}))$
        Outer Max Update: $\delta^k = \text{proj}_{||\circ||_\infty \leq \varepsilon}(\delta^{k-1} + \gamma_2 \sum_{i=1}^m w_i^k \nabla\hat{\ell}_i(\delta^k))$
    end
    Save: $\hat{X}_{M_F} = X + \delta^{k_{\max}}$
end
```

We follow algorithm 1 to create $2^m - 1$ data sets of adversarially perturbed images. Using these data sets as the sets of adversarial examples, we follow the work of [3] to perform Ensemble Adversarial Training. Our work differs from that of Tramer et al. by crafting adversarial examples which are necessarily diverse. It is widely known that adversarial attacks are transferable between models of different architectures. It is for that reason that when creating adversarial attacks against each pre-trained model individually, we expect crafted examples to be very similar. By selecting subsets of models which are fooled and ensuring all other models are not fooled, we guarantee that our adversarial examples are to at least some extent different from one another. This diversity should prevent a model from training to be robust to a specific adversarial perturbation while remaining vulnerable to other attacks.

4 Experiments

We proposed an inclusion-exclusion enhanced ensemble adversarial training method. This required four major steps. We completed 256 epochs whenever a model was trained.

First, we identified every possible yes-no combination of our training models, withholding the combination where all models are not fooled, and created 50,000 perturbed images for each combination using APGDA with our modified loss function as defined in (2).

We chose three training models: DenseNet-169, MobileNet v2, and Inception v3 [2]. These models were pre-trained on the CIFAR-10 dataset from modified official TorchVision implementations [2]. There are $2^3 - 1 = 7$ possible combinations of these models, each targeting a different subsection of the total adversarial perturbation space. The adversarial examples were bounded by an ℓ^∞ perturbation with adversarial power $\varepsilon = 0.0625$, which was also used in the experiments for regular EAT as implemented in [3].

Secondly, we exposed a pre-trained VGG-11 model [2] to our 350,000 perturbed images along with one round of regular adversarial training with the original 50,000 clean images. This served as our model developed with our proposed method, Inclusion-Exclusion Enhanced Ensemble Adversarial Training.

Thirdly, we performed traditional Adversarial Training with PGD and the Ensemble Adversarial Training in [3]) on separate pre-trained VGG-11 models from [2]. These models, along with the untouched pre-trained VGG-11 served as our baseline comparison models in the experiments. All additional training was performed for 256 epochs to maintain consistency.

Finally, we implemented both white and black box adversarial attack on each of the four models at varying values of ε. For the white box attacks, we created adversarial examples on the pre-trained VGG-11 architecture with both single-step PGD and FGSM. For the black box attacks, we implemented PGD and FGSM transfer attacks using GoogleNet and ResNet-18 as our holdout models.

The holdout and pre-trained models we used for our experiments are summarized in Table 1 below:

Trained Model	**Pre-Trained Models**	**Holdout Models**
VGG-11	DenseNet-169	GoogleNet
	MobileNet v2	ResNet-18
	Inception v3	

Table 1: **Models used for Ensemble Adversarial Training on CIFAR-10**

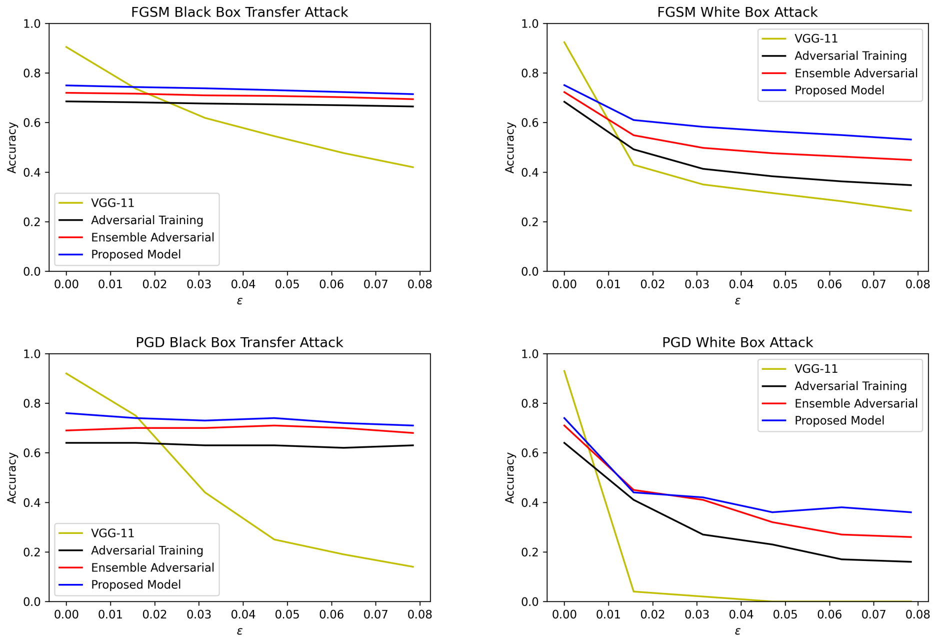

Figure 1: **Results from White Box and Black Box Attacks.** For various values of $\varepsilon \in (\frac{0}{255}, \frac{20}{255})$, we plot the performance of four different pre-trained VGG-11 models: one with no tuning, VGG-11 fine-tuned with traditional AT, VGG-11 fine-tuned with regular EAT, and VGG-11 trained with our proposed Inclusion-Exclusion Enhanced Ensemble Adversarial Training method.

Below are the reported accuracies for the various white and black box attacks against the four compared models:

ε	**VGG-11**	**AT VGG-11**	**EAT VGG-11**	**Proposed Method**
0	0.93	0.64	0.71	0.74
$\frac{4}{255}$	0.04	0.41	0.45	0.44
$\frac{8}{255}$	0.02	0.27	0.41	0.42
$\frac{12}{255}$	0.00	0.23	0.32	0.36
$\frac{16}{255}$	0.00	0.17	0.27	0.38
$\frac{20}{255}$	0.00	0.16	0.26	0.36

Table 2: **PGD White Box Attack Performance Accuracies**

ε	**VGG-11**	**AT VGG-11**	**EAT VGG-11**	**Proposed Method**
0	0.9239	0.6837	0.7227	0.7508
$\frac{4}{255}$	0.4294	0.4917	0.5486	0.61
$\frac{8}{255}$	0.35	0.4131	0.4977	0.5826
$\frac{12}{255}$	0.3154	0.3831	0.4761	0.5645
$\frac{16}{255}$	0.2824	0.3625	0.4627	0.5493
$\frac{20}{255}$	0.2441	0.3473	0.4488	0.5314

Table 3: **FGSM White Box Attack Performance Accuracies**

ε	**VGG-11**	**AT VGG-11**	**EAT VGG-11**	**Proposed Method**
0	0.92	0.64	0.69	0.76
$\frac{4}{255}$	0.75	0.64	0.7	0.74
$\frac{8}{255}$	0.44	0.63	0.7	0.73
$\frac{12}{255}$	0.25	0.63	0.71	0.74
$\frac{16}{255}$	0.19	0.62	0.7	0.72
$\frac{20}{255}$	0.14	0.63	0.68	0.71

Table 4: **PGD Black Box Transfer Attack Performance Accuracies**

ε	**VGG-11**	**AT VGG-11**	**EAT VGG-11**	**Proposed Method**
0	0.9047	0.6852	0.7198	0.7497
$\frac{4}{255}$	0.7367	0.6816	0.7164	0.7431
$\frac{8}{255}$	0.6186	0.6766	0.7096	0.738
$\frac{12}{255}$	0.545	0.6732	0.7072	0.7307
$\frac{16}{255}$	0.4768	0.6696	0.7022	0.7228
$\frac{20}{255}$	0.4197	0.6647	0.6942	0.7145

Table 5: **FGSM Black Box Transfer Attack Performance Accuracies**

Method of Training	**Training Time**
Traditional AT (with PGD)	6.44 hours
EAT	2.48 hours
Inclusion-Exclusion Enhanced EAT	1.83 hours

Table 6: **Training Times for Various Methods**

5 Conclusion

Based on above shown results, we observe that the proposed model has improved adversarial defenses when compared to both traditional adversarial training and Ensemble Adversarial Training. Our technique, Inclusion-Exclusion Enhanced Ensemble Adversarial Training, outperforms the other training processes against both black and white box attacks of various types because our technique mitigates existing issues with EAT regarding the lack of diversity in adversarial perturbations. Our inclusion-exclusion method promotes diversity and thereby allows for greater coverage of the adversarial space.

Across all four plots displayed in Figure 1, we may observe that our proposed model results in a lower accuracy on natural images, which are indicated by $\varepsilon = 0$, than the non-adversarially trained model. Indeed, on these clean images, the pre-trained VGG11 model performs best out of all four models. However, our model outperformed traditional AT as well as EAT on clean images. This is of notable benefit, as strong performance on non-perturbed images is necessary for real-life applications of adversarial training.

It has been previously shown that EAT can yield improved robustness to transferred black box attacks from holdout models. The plots in Figure 1 indicate that our experiments corroborate this result. We were pleasantly surprised to find that our method yielded additional robustness to white-box attacks beyond traditional EAT. However, in general, the Attack Success Rate (ASR) of white box attacks increased more dramatically than the ASR of black box attacks as the perturbation size increased. Nevertheless, this looks to be a promising result and warrants further experiments for verification. We also note that for all models and for both black box and white box attacks, the PGD attack has a higher ASR than does the FGSM attack. This is likely due to PGD being a more sophisticated attack method than the single-step FGSM.

In regards to training time, we expected EAT to take less time than traditional AT because of the decoupling of image creation from training time. This was indeed the case, as EAT took approximately 2.48 hours and regular AT took approximately 6.44 hours. We found that our technique, Inclusion-Exclusion Enhanced EAT took a mere 1.83 hours, boasting improved training time over both traditional and Ensemble AT. However, it should be noted that we do not include the time consumed for generating the 350,000 adversarial examples in these metrics, as the adversarial image generation and the training itself are totally decoupled in our method. The total time for generating our images was approximately 10.2 hours. Since the adversarial examples can be saved, these images can easily be retrieved for future experiments using different holdout models, accentuating the scalability of our method on additional models.

6 Future Directions

In the future, we would like to see if increasing the number of models used results in increased robustness. Since each additional model approximately doubles the number of necessary training images and training time, we would need to consider computational resource availability. However, due to the decoupling benefit of our method, this could also be seen as an advantage: at the cost of only one additional model, we double the number of diverse adversarial perturbation training data. Moreover, once the training images are created, the training time doubling is comparatively trivial.

Additionally, we would like to examine the consistency of our results against a wider variety of hyper-parameters, such as different perturbation (ε) sizes and α, β hyperparameters for the creation of the adversarial examples. We would also like to examine the resilience of our method through ablation testing.

Finally, we would like to extend our Inclusion-Exclusion Enhanced Ensemble Adversarial Training technique to other notable image classification datasets such as MNIST, CIFAR-100, and ImageNet to see how results compare. One important consideration is the availability of pre-trained models specific to the other benchmark datasets. Performing the initial training on clean images would be yet another computationally expensive task to assign to our already limited resources. [3] has shown results for Ensemble Adversarial Training on MNIST, which is then used as a baseline for results on ImageNet. We would like to see if our technique shows an improved training time and increased robustness to black-box and white-box attacks on these datasets as well.

References

[1] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. 2021.

[2] Huy Phan. Pytorch models trained on cifar-10 dataset, 2021.

[3] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[4] Jingkang Wang, Tianyun Zhang, Sijia Liu, Pin-Yu Chen, Jiacen Xu, Makan Fardad, and BoLi. Adversarial attack generation empowered by min-max optimization. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:16020–16033, 2021.

[5] 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. 2020.