TREND: Transferability-Based Robust ENsemble Design

Deepak Ravikumar ‡, Sangamesh Kodge ‡, Isha Garg ‡, and Kaushik Roy ‡, Fellow, IEEE

Abstract—Deep learning models hold state-of-the-art performance in many fields, but their vulnerability to adversarial examples poses a threat to their ubiquitous deployment in practical settings. Additionally, adversarial inputs generated on one classifier been shown to transfer to other classifiers trained on similar data, which makes the attacks possible even if model parameters have not yet been systematically studied, leading to a gap in our understanding of robustness of neural networks to adversarial inputs. In this work, we study the effect of network architecture, optimizer, input, weight, and activation quantization on transferability of adversarial samples. We also study the transferability of different attacks. Our experiments reveal that transferability is significantly hampered by input quantization and architectural mismatch between source and target, and the choice of optimizer turns out to be critical. We observe that transferability is architecture-dependent for both weight and activation quantized models. To quantify transferability, we use simple metric and demonstrate the utility of the metric in designing a methodology to build ensembles with improved adversarial robustness. When attacking ensembles we observe that “gradient domination” by a single ensemble member model hampers existing attacks. To combat this we propose a new state-of-the-art ensemble attack. We compare the proposed attack with existing attack techniques to show its effectiveness. Finally, we show that an ensemble consisting of carefully chosen diverse networks achieves better adversarial robustness than would otherwise be possible with a single network. The source code for this work has been made available at https://github.com/purdue-nrl/TREND.

Impact Statement—Deep neural networks (DNNs) have been successful in solving a wide variety of tasks and are believed to have a potential to revolutionize human life. However, adversarial attacks expose the brittle nature of the learned solution; one can easily generate images that fool DNNs with changes imperceptible to humans. Such adversarial attacks pose a great challenge to the deployment of DNNs in safety critical applications. A few settings where adversarial attacks can inflict considerable damage on human life are as follows. 1) Detection of bots and misinformation on social media. 2) Deployment in self driving cars. 3) Deployment in medical imaging and diagnostics. In most real world scenarios, the adversary does not have access to the model parameters. Yet, the adversary is still able to successfully craft attacks due to the transferable nature of these attacks. In this work, we systematically study transferability of these adversarial samples across models and preprocessing techniques, and propose a design methodology to build ensembles of DNNs that perform better under adversarial attacks. Our methodology offers an alternative way of building more robust model for realworld deployment.

Index Terms—Adversarial machine learning, deep learning, image classification, neural networks.

I. INTRODUCTION

DEEP learning has become state-of-the-art for many machine learning tasks over the past few years. Deep neural networks (DNNs) have achieved human level performance in image recognition [1], [2]. They have also been used for speech recognition [3] and natural language processing [4]. However, recent research [5]–[7] has shown the existence of small perturbations which when added to the input can cause DNNs to misclassify the input. These adversarial perturbations are imperceptible to the human eye and are crafted with the specific intent of fooling DNN into misclassifying the images. The existence of adversarial inputs has led to a considerable knowledge gap in the explainability of deep neural nets, which limits their use in safety critical applications such as malware detection [8] or autonomous driving systems [9].

The generation of adversarial inputs usually requires access to the target network (the network to be attacked) in the form of network weights or logsits. However, in most practical scenarios, the adversary does not have access to the internal parameters, but is able to observe the output of the network for a given input [10]. In such cases, previous works [11]–[13] have shown that the adversary can train a substitute network by generating synthetic data using the outputs or the logits of the target network. Such query-based attacks, often labeled as black-box attacks, are successful because of the transferability of adversarial perturbations: attacks crafted to fool one network often fool another network trained on similar data.

In this article, with the motive of understanding the transferability of adversarial images between networks, we systematically study how transferability is affected by choice of network architecture, network initialization, optimizer, input, weight, and activation quantization made by the defender and the attack methodology of the adversary. To account for these factors, we use a metric of transferability defined as the ratio of transferred

2691-4581 © 2022 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.
images to the generated images. Further we propose a new state-of-the-art ensemble attack observing that “gradient domination” by a single ensemble member model hampers existing attacks. Additionally, we show that understanding transferability aids in building robust ensembles of DNNs.

The key contributions of this work are summarized as follows:

1) We explore how the choice of initialization, model architecture, optimizer, quantization of input, weight, and activation by the defender and attack methodology by the adversary affect the transferability of adversarial images from one DNN to another. We also make empirical suggestions on the most effective model to use, both from the point of view of an adversary and defender. These experiments were performed on various small and large datasets, attacked using projected gradient descent (PGD) [14], Carlini Wagner $L_2$ attack [15], and DeepFool [16].

2) We devise a new state-of-the-art method for adversarially attacking an ensemble of DNNs. We identify the limitations of existing techniques and show that the proposed average gradient direction (A-GD) attack achieves SoTA attack performance by comparing it with existing attack techniques.

3) We outline a methodology, transferability-based robust ensemble design (TREND), which uses a set of diverse models having low transferability to build an ensemble with higher robustness than would otherwise be possible with a single DNN.

II. RELATED WORK

Attack Methodologies: Many methods have been proposed for generating adversarial inputs. Some of the popular attacks in literature are fast gradient sign method (FGSM) [6], a single step attack; basic iterative method (BIM) [7], a multistep iterative attack; Carlini Wagner attack (CW) [15], an optimization-based attack; and PGD [14], an iterative version of FGSM with a random start point within an $\epsilon$ bound around the clean image. There are numerous other attacks like Jacobian-based saliency map attack (JSMA) [17], DeepFool [16], and elastic-net attacks (EADAttack) [18]. In this work, we study transferability under PGD [14], Carlini Wagner $L_2$ [15] attack, and DeepFool [16]. These attacks were chosen to encompass iterative, optimization, and decision boundary attacks, respectively, and have been shown [19], [20] to circumvent a diverse array of defenses. Recent work [49] proposes modifications to loss function to generate adversarial perturbations which generalize across several models trained for image captioning. We utilize back-propagation through differential approximation (BPDA) [20] style gradient back-propagation to allow gradients to propagate through any nondifferentiable functions such as input or activation quantization, wherever needed.

Transferability: The ability of adversarial perturbations generated on one model to successfully fool other models was first observed in [5] and subsequently in [6]. It has been shown that transferability is not unique to deep learning, but exists across various machine learning classifiers [21]. The authors of [21] show transferability across DNNs, k-nearest neighbors, decision trees, support vector machines, and logistic regression. Such transfer attacks can be implemented successfully in both targeted or nontargeted scenarios [22]. Moreover, the transferability of adversarial examples is hypothesized to occur due to the alignment of the decision boundaries across various models [22]. The adversarial examples are shown to span a contiguous subspace of high dimensionality ($\approx 25$) [23]. The authors find that a significant factor of this subspace is shared between two models, which they attribute to the closeness of the decision boundaries learnt by the models. However, Wu et al. [24] argue that the transfer is asymmetric and hence, cannot be explained completely by closely aligned decision boundaries.

Ensembles: Ensemble methods leverage the averaging effects of multiple models to make a final prediction. Model averaging methods include bagging [25] and boosting [26], [27]. We consider one of the simplest approaches to make decisions: the majority voting strategy. We choose the majority voting ensemble method because it is more challenging from an adversarial point of view. This is because majority voting obfuscates the gradient compared to other ensemble methods (such as average and weighted average). Ensembles are used as a tool to reduce variance in classifiers [28]. However, we utilize them to gain adversarial robustness without significantly degrading the performance on clean images. Recent research [29]–[31] has focused on developing defense strategies using ensembles. However, it has been shown [32] that ensembles are not immune to adversarial attacks. In this article, we take a deeper look at the link between transferability and the adversarial accuracy of ensembles, first suggested in [32]. More recently the authors of [50] propose adversarially training an ensemble of models as a system for adversarial defense. Further, Alex et al. [51] proposed diversity promoting learning algorithm to train deep ensemble models. We show that careful selection of individual DNNs that make up the ensemble can improve the overall robustness of the ensemble.

III. METHODOLOGY

A. Selecting the Right Model

Recent research [29]–[31] suggests that ensembles provide robustness against adversarial attacks. The robustness of the ensemble, or the lack thereof, is hypothesized to arise from the property of transferability [32]. Since images are transferable to different models, an image meant to fool one network will fool a majority of the networks in the ensemble and, therefore, the whole ensemble. We delve deeper into this hypothesis by constructing ensembles with models that have varying transferability between them, as captured by the transferability metric (see Section IV). We expect ensembles with models that have a high transferability metric (averaged by exchanging the source and target) between them to be less robust, as an image that fools one model will transfer well and fool the ensemble. We assemble ensembles by choosing pairs of models with low transferability metric among them, resulting in an ensemble with improved...
robustness. In the subsequent sections, we go into the details of how to effectively attack and design ensembles.

### B. Attacking an Ensemble

1) **Existing Ensemble Attacks:** Optimization-based attacks like PGD use the gradient of the loss function with respect to the input to decide the direction of change. However, the gradient for an ensemble is undefined. The most common approach is to average the gradient from each model [19], [32], [46]. We refer to this as the Direction of Average Gradient (D-AG) attack and has been shown to be highly effective [19]. Another approach is to average the gradients from the models that voted for the final predicted class as in EMPIR [31]. These attacks iteratively generate the adversarial images similar to PGD described by

\[
x^0 = \text{clamp}(x_{\text{nat}} + \alpha \cdot k)
\]

\[
x^{t+1} = \text{clamp}(x^t + \alpha \cdot D^t)
\]

where \(x_{\text{nat}}\) is a natural image, \(k \in \mathbb{R}^d\) and is sampled from \(\text{unif}(-1, 1)\), \(d\) is the input image dimension, \(x^t\) is the adversarial image at \(t\) iteration, \(\alpha\) is the \(L_\infty\) bound for the attack, \(D^t \in \{-1, 1\}^d\) is the gradient direction for the ensemble at \(t\) iteration and the clamp function clamps its input to the image bounds \([0,1]\). The D-AG attack commonly used in literature calculates the gradient direction \(D_{\text{DAG}}^t\) given by

\[
G_i^t = \nabla_{x^t} L(\theta_i, x^t, y)
\]

\[
D_{\text{DAG}}^t = \text{sgn} \left( \frac{1}{N} \sum_{i=1}^{N} G_i^t \right)
\]

where \(N\) is the number of models in the ensemble, \(\text{sgn}\) is the sign function and \(\nabla_{x^t} L(\theta_i, x^t, y)\) is the gradient of the \(i\)th member of the ensemble whose parameters are \(\theta_i\).

2) **Proposed Attacks:** The challenge when attacking ensembles is that gradient from one of the models in the ensemble tends to dominate the average gradient, especially when the attack strength is low a phenomenon we dub “gradient domination.” This results in an adversarial direction that is unable to fool multiple models simultaneously, reducing the attack’s effectiveness. Hence, we devise attacks that account for this phenomenon and show that we achieve SoTA attack success rates. Observing that the PGD attack uses the gradient direction rather than the gradient (magnitude and direction) is key to our attacks. This provides flexibility in identifying the gradient direction for the ensemble and allows our attack methods to counter the “gradient domination” phenomenon. The first method, which we call unanimous gradient direction (U-GD) attack chooses only those gradient directions where all the individual models’ gradient directions align. The second method, A-GD, calculates the gradient direction for the ensemble by calculating the sign of average gradient direction.

**Unanimous Gradient Direction:** The U-GD attack calculates the gradient direction \(D_{\text{UGD}}^t\) for the ensemble as given by

\[
S_i^t = \text{sgn}(G_i^t)
\]

\[
A^t = \frac{1}{N} \left( \sum_{i=1}^{N} S_i^t \right)
\]

\[
M^t = \lfloor |A^t| \rfloor \cdot \text{sgn}(A^t)
\]

\[
D_{\text{UGD}}^t = A^t \cdot M^t
\]

where \(A^t \cdot M^t\) represents the elementwise product of the average gradient direction \(A^t\) with the mask \(M^t\), \(\lfloor |A^t| \rfloor\) represents the floor of the absolute value of \(A^t\), \(N\) is the number of models in the ensemble, and \(\text{sgn}\) is the sign function. Equation (3) translates to choosing only those gradient directions where all the individual model’s gradient directions are in agreement. However, as the number of models in the ensemble increase the effectiveness of this approach decreases (see Figs. 8 and 9). This is due to the strict requirement that all models must agree on the gradient direction. To overcome this we propose an A-GD.

**Average Gradient Direction:** In this approach, the gradient direction for the ensemble is calculated by averaging the gradient direction from each model

\[
D_{\text{AGD}}^t = \text{sgn}(A^t)
\]

where \(A^t\) is given by (3) and \(\text{sgn}\) is the sign function. The effectiveness of the A-GD attack is because of its ability to counter the “gradient domination” problem by using the gradient direction rather than both the magnitude and direction. This is illustrated with the help of Fig. 1.

The first row of Fig. 1 shows an image from CIFAR-10 and the corresponding adversarial gradients visualized as an RGB image for FP, Q1, and Q2 ResNet18 models. The second row visualizes the individual gradient directions S1 [i.e., \(S_i^t\) from (3)], S2, and S3, respectively, as an RGB image. The third row’s first image visualizes the A-GD’s gradient direction, and the rest of the row shows the difference between A-GD’s direction and individual gradient direction. The final row does the same.
but for D-AG. The first image of the final row visualizes the average gradient from the three models, the second visualizes the gradient direction for this average. From the visualizations we can clearly see the similarity between D-AG and S1. Further we see the “D-AG - S1” is nearly all zero implying the D-AG is dominated by S1, i.e., model 1’s gradient direction. Comparing this to A-GD, we observe that “A-GD - S1” is not all zero implying that the A-GD’s adversarial direction is not dominated by gradient directions from a single model but is more equally shared among the member models making the A-GD attack more effective.

C. Robust Ensemble Design

In this subsection, we utilize the transferability metric introduced in Section IV to put forth a methodology TREND, to build an ensemble with improved robustness. The ensembles predicts a class using majority voting. In case of conflict (i.e., no majority vote), one of the models is chosen at random and its output is considered as the ensemble’s prediction. The hypothesis that high transferability between models in an ensemble results in reduced adversarial robustness was first suggested in [32]. We leverage this idea and build an ensemble with improved robustness by choosing models with low transferability. To identify these models, we consider a list of all pairs of individual models under consideration and calculate the transferability metric for each pair. To account for asymmetry, we average the two numbers obtained by interchanging source and target. From this list, we choose a desired number of models with the lowest transferability metric.

IV. EXPERIMENTAL SETUP

In this section, we perform experiments to study how network initialization, network architecture, input, weight, and activation quantization affect transferability. We study the effect of these factors independently on CIFAR-10, CIFAR-100 [33], and ImageNet [34] datasets. All the models in the article were trained using either the stochastic gradient descent (SGD) or Adam optimizer. The specific optimizer used is delineated in each case. Initial learning rate was set to $10^{-2}$ and it was scaled down by a factor of 10 at 60% and 80% completion using a learning rate scheduler. The models trained using SGD used a momentum of 0.9 and weight decay of $5 \times 10^{-4}$. The models were trained for 250 epochs on the ImageNet dataset and 400 epochs on CIFAR-10 and CIFAR100 datasets. At the end of each epoch the model was evaluated on the validation set and the model weights that achieved the best validation accuracy was saved. The model weights that achieved the best validation accuracy was used to evaluate the network performance on the test set and its accuracy was reported. Table I shows the training, validation, and test set sizes for each dataset used.

| Dataset   | Train set size | Validation set size | Test set size |
|-----------|----------------|---------------------|---------------|
| CIFAR-10  | 45000 (90%)    | 5000 (10%)          | 10000         |
| CIFAR-100 | 45000 (90%)    | 9000 (10%)          | 10000         |
| ImageNet  | 124937 (97.5%) | 32029 (2.5%)        | 50000         |

Baseline Accuracies: The baseline accuracies of the quantized models on different datasets with base architecture ResNet18 and VGG11 are shown in Tables II and III, respectively. The baselines accuracies for different architectures are presented in Table IV. Table IV lists the performance of pretrained models provided by PyTorch, while Table II lists performance

| Quantization | CIFAR-10 | CIFAR-100 | ImageNet |
|--------------|----------|-----------|----------|
| FP           | 93.33 %  | 72.55 %   | 55.73 %  |
| Q8           | 94.03 %  | 72.38 %   | 55.63 %  |
| Q6           | 93.93 %  | 72.58 %   | 55.39 %  |
| Q4           | 93.42 %  | 71.10 %   | 55.09 %  |
| Q2           | 88.22 %  | 62.66 %   | 50.12 %  |
| Q1           | 78.25 %  | 48.17 %   | 40.27 %  |
| HT           | 86.99 %  | 58.95 %   | 46.95 %  |
| W16          | 93.36 %  | 72.00 %   | -        |
| W8           | 93.30 %  | 72.71 %   | -        |
| W4           | 93.10 %  | 72.79 %   | -        |
| W2           | 92.80 %  | 71.04 %   | -        |
| W1           | 91.49 %  | 68.27 %   | -        |
| A16          | 91.15 %  | 65.92 %   | -        |
| A8           | 90.86 %  | 65.36 %   | -        |
| A4           | 90.96 %  | 65.67 %   | -        |
| A2           | 90.69 %  | 64.66 %   | -        |
| A1           | 88.70 %  | 60.04 %   | -        |

| Quantization | CIFAR-10 | CIFAR-100 | ImageNet |
|--------------|----------|-----------|----------|
| FP           | 88.58 %  | 55.50 %   |          |
| Q8           | 87.74 %  | 56.58 %   |          |
| Q6           | 87.51 %  | 55.24 %   |          |
| Q4           | 87.10 %  | 54.50 %   |          |
| Q2           | 82.58 %  | 50.44 %   |          |
| Q1           | 74.10 %  | 37.88 %   |          |
| HT           | 80.74 %  | 46.27 %   |          |
| W16          | 88.16 %  | 56.58 %   |          |
| W8           | 88.16 %  | 55.24 %   |          |
| W4           | 87.75 %  | 54.50 %   |          |
| W2           | 86.82 %  | 50.44 %   |          |
| A16          | 88.45 %  | 57.33 %   |          |
| A8           | 88.15 %  | 57.63 %   |          |
| A4           | 87.63 %  | 41.12 %   |          |
| A2           | 87.00 %  | 57.71 %   |          |
| A1           | 74.74 %  | 32.76 %   |          |

| Arch.    | CIFAR-10 | CIFAR-100 | ImageNet |
|----------|----------|-----------|----------|
| RN18     | 93.33 %  | 72.55 %   | 69.76 %  |
| RN34     | 94.69 %  | 75.85 %   | 73.31 %  |
| RN50     | 94.95 %  | 77.45 %   | 73.31 %  |
| RN101    | 95.00 %  | 76.98 %   | 77.37 %  |
| VGG11    | 88.58 %  | 78.55 %   | 69.02 %  |
| VGG19    | 89.61 %  | 69.74 %   | 72.38 %  |
| VGG11BN  | 89.94 %  | 63.68 %   | 72.38 %  |
| VGG19BN  | 91.34 %  | 66.04 %   | 72.38 %  |
| DN121    | -        | -         | 74.43 %  |
| WRN50_2  | -        | -         | 78.47 %  |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
Fig. 2. Average number (averaged over 5 seeds for CIFAR, single seed for ImageNet) of adversarial images transferred from source to target for various architectures under PGD and CW $L_2$ attacks. (a) CIFAR-10, architecture analysis under PGD attack. (b) ImageNet, architecture analysis under PGD attack. (c) CIFAR-10, architecture analysis under CW $L_2$ attack ($\kappa = 15$). (d) ImageNet, architecture analysis under CW $L_2$ attack ($\kappa = 30$).

Transferability: We study transferability between models by generating adversarial images on the source model and evaluating them on the target model. The number of adversarial images that transfer from the source to the target model is affected by the baseline accuracies of both the source and the target models. A source with high classification accuracy will generate more adversarial images compared to a source with low classification accuracy. Similarly, the target model accuracy also has the same effect. More adversarial images transfer to target models with higher accuracy when compared to models with lower accuracy. Different datasets have different testset sizes, hence comparing transferability across different datasets also presents a challenge.

To fairly evaluate transferability we need to use a metric that accounts for these factors. We define the transferability metric $TM$ as

$$TM = \frac{f_{st}}{f_{ss}}$$

where $f_{st}$ is the number of adversarial images transferred from source to target, and $f_{ss}$ is the number of adversarial images generated on the source model.
where $f_{ss}$ is the number of adversarial images generated by attacking the source model and $f_{st}$ is the number of images that transfer from the source to the target model. To account for differences in accuracies, we choose a subset of the testset that was correctly classified by both the source and the target model. The transferability metric $TM$ is a number between $[0, 1]$ and represents a quantitative measure of the transferability between a given pair of models. The metric does not account for attack strength variation, however, we found the $TM$ at a single attack strength is representative in most cases. Appendix D fits a curve to model transferability as a function of attack strength and shows that such a function is monotonically increasing implying $TM$ at a single attack strength is sufficient. The adversarial images were generated using PGD for 40 iterations with an $\epsilon$ of $8/255$ and step size of 0.01, Carlini Wagner $L_2$ attack for 100 iterations and DeepFool attack.

### Confusion Matrices (CM):
We present the results as a confusion matrix which represents the transferability metric ($TM$) obtained after performing transfer attacks between different pairs of models, see Fig. 2(a). These numbers were obtained by averaging over multiple runs across different seeds. The rows and columns of confusion matrix represent the effect of changing the target model and the source model, respectively. The deeper the color of the cells in the CM, the higher the transferability between the corresponding source and target model. The transferability metric ($TM$) can also be viewed as the factor with which the adversarial accuracy of the black box model is expected to decrease. An adversary performing black box attacks has control over only the source model and would want to choose a model with the highest row average, in order to successfully attack a range of target models. Similarly, a defender controls only the target model and would want to choose the model with the lowest column average, in order to have the lowest transferability across any chosen source model. The averages are, therefore, shown alongside the corresponding rows and columns. The confusion matrices for CIFAR-10, CIFAR-100, and ImageNet are shown in Blue, Green, and Red, respectively.

## V. Transferability Analysis

### A. Attack Methodology
In this work we study transferability under PGD [14], Carlini Wagner $L_2$ [15], and DeepFool [16] attacks. From Table V, we observe that PGD generates adversarial images that transfer more readily compared to DeepFool or CW $L_2$ attack. We investigated the reason behind such poor transferability of CW $L_2$ ($\kappa = 0$) and DeepFool. We observed that CW and Deepfool attacks when used without a constraint on the adversarial sample confidence tend to identify low confidence examples with the perturbations tailored for a specific network and hence the transferability of such adversarial images is low. By increasing the confidence of the adversarial images generated by CW (achieved by tweaking the $\kappa$ parameter), we observed increased transferability (see Table V). Our observations are corroborated by the authors of the CW attack in their article [15]. However, increasing transferability comes at the cost of increased perturbation distance. This can be observed from Table V, which shows the increase in $L_2$ and $L_\infty$ norms from $\kappa = 0$ and $\kappa = 30$. Increasing the transferability of DeepFool by increasing the confidence of the generated adversarial images counters the objective of the attack and hence we do not recommend DeepFool for transfer attacks and exclude it from further analysis in the article. In the next subsection we study the effect of optimizer and network initialization on transferability.
TABLE VI
AVERAGE NUMBER (MEAN ± STD. DEV) OF ADVERSARIAL IMAGES GENERATED AND TRANSFERRED FROM SOURCE TO TARGET ON CIFAR-10 (CF10) AND CIFAR-100 (CF100) DATASETS FOR DIFFERENTLY SEEDED MODELS TRAINED USING ADAM AND SGD OPTIMIZER UNDER PGD, CW L2, AND DEEPPROOF ATTACK

| Dataset | Arch. | Attack | SGD as source to Adam as target | Adam as source to SGD as target |
|---------|-------|--------|---------------------------------|---------------------------------|
|         |       |        | Generated  | Transferred | Trans. (%) | Generated  | Transferred | Trans. (%) |
| CF10    | ResNet18 | PGD    | 9072 ± 14 | 8312 ± 69 | 91.63 ± 0.74 | 9068 ± 14 | 8646 ± 76 | 95.34 ± 0.79 |
|         |        | CW (κ = 30) | 8997 ± 23 | 7945 ± 133 | 88.31 ± 1.51 | 9072 ± 14 | 1349 ± 178 | 14.88 ± 1.96 |
|         | VGG11  | PGD    | 7700 ± 110 | 5967 ± 105 | 77.50 ± 1.11 | 2829 ± 654 | 1795 ± 299 | 64.46 ± 4.28 |
|         |        | CW (κ = 30) | 8340 ± 33 | 7190 ± 126 | 86.21 ± 1.46 | 8343 ± 33 | 453 ± 41 | 5.42 ± 0.48 |
|         | VGG11BN| PGD    | 8410 ± 24 | 5222 ± 185 | 62.09 ± 2.21 | 8417 ± 21 | 4242 ± 213 | 50.42 ± 2.49 |
|         |        | CW (κ = 30) | 8408 ± 21 | 4781 ± 268 | 56.87 ± 3.20 | 8418 ± 21 | 599 ± 56 | 7.12 ± 0.67 |
| CF100   | ResNet18 | PGD    | 6423 ± 31 | 5553 ± 121 | 86.45 ± 1.73 | 6420 ± 31 | 5742 ± 85 | 89.44 ± 1.20 |
|         |        | CW (κ = 30) | 6130 ± 62 | 4866 ± 190 | 79.39 ± 3.08 | 6423 ± 31 | 1785 ± 152 | 27.23 ± 0.33 |
|         | VGG11  | PGD    | 4284 ± 46 | 2778 ± 62 | 64.68 ± 1.61 | 2352 ± 179 | 1026 ± 53 | 43.72 ± 1.69 |
|         |        | CW (κ = 30) | 4729 ± 40 | 2018 ± 119 | 42.69 ± 2.74 | 4727 ± 40 | 231 ± 18 | 4.90 ± 0.37 |
|         | VGG11BN| PGD    | 5423 ± 28 | 3563 ± 76 | 65.71 ± 1.30 | 5114 ± 41 | 3064 ± 83 | 59.91 ± 1.60 |
|         |        | CW (κ = 30) | 5529 ± 28 | 2219 ± 92 | 40.14 ± 1.61 | 5557 ± 28 | 796 ± 62 | 14.33 ± 1.10 |

B. Training

1) Optimizer: From Table V we observe that the choice of optimizer does have a significant effect on transferability. Table V suggests that ResNet18 networks trained with SGD seem to be less transferable compared to ResNet18 networks trained with Adam optimizer. However, the opposite is true for VGG11 and VGG11BN. Hence we are unable to make a recommendation on the choice of optimizer. Further, Table VI shows the transferability between models trained using different optimizers. Interestingly, we observe an asymmetry, adversarial images generated for SGD trained models transfer well to Adam trained models, however, the reverse is not true. This empirical result suggests that improved ensemble diversity can be achieved by including models trained using Adam and SGD.

2) Network Initialization: The gradient descent algorithm is known to be sensitive to initialization [35]. Different parameter initializations lead training to converge to different solutions [36]. We investigate the effect of initialization by training ten models with different random initial seeds. To study the effect of initialization we consider ResNet18 [37], VGG11, and the batch normalized version of VGG11 (VGG11 BN) [38] architectures.

Table V shows the baseline accuracies and the number of adversarial images transferred from the source to the target model under PGD, CW L2, and DeepFool attack averaged over ten differently seeded models on CIFAR-10 and CIFAR-100. For example, Table V shows ResNet18 trained using SGD on CIFAR-10 as transferring 8166 ± 104 adversarial images. This number represents the average and standard deviation over the various seeds (i.e., Seed 1 to 10) as sources and targets. Each seed was chosen as a source and the transferred number of images were used to obtain a 90 datapoint average (ten seeds, nine targets per seed since the source seed was excluded from being the target). From Table V we observe very low deviation in the transferred adversarial images suggesting that the number of images transferred does not radically change across different seeds trained using the same optimizer. However, the choice of optimizer is quite significant. This suggests that initialization using the Kaiming initialization [39] strategy may not play a significant role in transferability but the choice of optimizer does.

Table V also seems to imply that architecture is an important consideration for transferability. We investigate this in the next subsection.

C. Architecture

We study the effect of architecture on transferability by analyzing cross model transfer of adversarial images between ResNet18 (RN18), ResNet34 (RN34), ResNet101 (RN101), VGG11, VGG19, VGG11BN, VGG19BN, DenseNet121 (DN121) [40], and WideResNet50_2 (WRN50_2) [41]. Fig. 2 shows the average (over five seeds) number of adversarial images transferred from source to target under PGD attack for various architectures on CIFAR-10 and ImageNet (see Appendix A for CIFAR-100 and CW L2 results).

Fig. 2 shows the transferability of PGD and CW L2 attacks across various architectures on CIFAR-10 and ImageNet dataset. Fig. 2(a) and (c) can be interpreted by analyzing the four quadrants, with each quadrant representing a family of source or target model architectures (ResNet or VGG variants). The top-right quadrant of Fig. 2(a) and (c) is lighter than the bottom-left quadrant. This implies that adversarial images generated on VGG are more transferable to ResNets than the other way around. The results for CIFAR-100 follow the same trend and are shown in Appendix A. Surprisingly, we find that the matrices are considerably asymmetric. These findings reveal that transferability is not commutative. Another empirical observation is that the left half of Fig. 2(a) and (c) is the darkest, implying that ResNets are more susceptible to transfer attacks. This observation also corroborates by [42] whose authors attributed this to the skip connections of ResNets.
and leveraged this understanding to build better transfer attacks. These trends also hold for ImageNet to certain extent. However, we see a significant drop in transferability across the board when compared to CIFAR-10 and CIFAR100; this is especially true with CW $L_2$ attack [see Fig. 2(d)]. From Fig. 2, we observe that the column averages for ResNet18, ResNet34, VGG11, and VGG19 are among the highest. The high column average for VGG networks can be attributed to intrafamily transferability, with just two numbers boosting up the column average. Hence, VGG models are better source models for the adversary because the row averages for VGG networks are consistently high across various datasets and provides the highest chances for successful black-box attacks. Petrov and Hospedales [43] also made similar observations with respect to VGG networks. ResNets are easier targets and should, therefore, be avoided by defenders, as they consistently show very high column averages across various datasets.

D. Quantization

Recent research [30], [31], [44] suggested that quantization has potential to provide robustness against adversarial images. We expect these trends to be applicable to transferability as well. Therefore, we study how input, weight, and activation quantization affect transferability. For all experiments henceforth, our base model is ResNet18. This is because we hope to show improved robustness for the challenging case of easy adversarial targets (i.e., ResNets). Thus, the less challenging case of harder targets (i.e., inherently more robust to transfer attacks) would also benefit from the proposed methodology. Further, the results for VGG11 as the base model are presented in Appendices B and C.

![Image](https://example.com/image.png)

**Fig. 3.** Average (over five seeds for CIFAR, single seed for ImageNet) number of adversarial images transferred from source to target for input quantized models. (a) CIFAR-10, input quantization analysis under PGD attack. (b) ImageNet, input quantization analysis under PGD attack. (c) CIFAR-10, input quantization analysis under CW $L_2$ attack ($\kappa = 30$). (d) ImageNet, input quantization analysis under Carlini Wagner $L_2$ attack ($\kappa = 30$).

![Image](https://example.com/image.png)

**Fig. 4.** Decision boundaries around image (a) of a full precision and 1-b quantized input model. (a) Image from CIFAR-10. (b) Decision boundary for FP. (c) Decision boundary for Q1.
1) Input Quantization: Input quantization, as the name suggests, quantizes the input to the network. We analyze various input bit widths ranging from 8-b down to 1-b per channel, per pixel for both the source and the target models. We quantize a minibatch of images using the following formula:

\[ b = \frac{i_{\text{max}} - i_{\text{min}}}{2^n} \]

\[ I_{\text{quant}} = \left\lfloor \frac{I - i_{\text{min}}}{b} \right\rfloor b + \left( \frac{1}{2} b + i_{\text{min}} \right) \]

where \( I_{\text{quant}} \) is the quantized minibatch of images, \( b \) is the bin width, \( n \) is the bit width used for input quantization, \( i_{\text{min}} \) is the minimum, and \( i_{\text{max}} \) is the maximum value of the minibatch \( I \). This scheme is similar to the one suggested by [44], the difference being, we normalize the input before quantization. The inputs were normalized using the Z-score method. We also consider the nonlinear quantization scheme of halftoning described in [45]. The quantized models are represented by “Q” bit-width, halftone by “HT” and “FP” refers to the full precision network. We use BPDA [20]-based gradient backpropagation through the quantization scheme. From Fig. 3 (see Appendix B for CIFAR-100 results), we see that low bit width input quantized models (HT, Q1, and Q2) have very low transferability under both PGD and CW \( L_2 \) attacks. Additionally, transferability of input quantized models is highly asymmetric. It is far easier to transfer from Q2 to various models than it is to transfer to Q2. This is also true for Q4 and HT, though to a lesser extent. To further understand the effect of input quantization, we visualize the decision boundaries of the network. The basis (i.e., \( x \) and \( y \) axes) of the visualization shown in Fig. 4 were chosen to be the normalized adversarial gradient vector obtained from PGD (\( x \)-axis) and a random vector orthogonal to the former. Using these two vectors as the basis, the input space was traversed and the corresponding classes represented with different colors. The centers of Figs. 4(b) and Figs. 4(c) represent Figs. 4(a) in the input space. We observe that input quantization increases the distance to the decision boundary in most directions; however, transfer attacks still successfully find adversarial examples.

This leads us to conclude that low bit-width input quantization significantly reduces the success of transfer attacks. For instance, quantizing the inputs of CIFAR-10 from FP to Q1 improves adversarial accuracy by \( \sim \) 9% between Q1 and FP. Input quantized models with bit width greater than 2 make better source models for adversaries. This is because these models have high row averages (see Fig. 3), which results in the highest chance for

Fig. 5. Average number (averaged over five seeds) of adversarial images transferred from source to target on CIFAR-10 dataset on weight and activation quantized models. (a) Weight quantized models under PGD attack. (b) Activation quantized models under PGD attack. (c) Weight quantized models under CW \( L_2 \) attack (\( \kappa = 30 \)). (d) Activation quantized models under CW \( L_2 \) (\( \kappa = 15 \)) attack.
a successful black-box transfer attack. Input quantized models with bit widths 1, 2 or HT are more robust to transfer attacks, and hence make better target models for the defenders. This is clear from the low column averages (see Fig. 3) for these models across various datasets, provide the best chance of defense.

2) Weight and Activation Quantization: In this subsection, we study how quantizing the network parameters and activations affects transferability. Fig. 5 shows the transferability among ResNet18 models with different weight and activation bit precisions for CIFAR-10 (see Appendix C for CIFAR-100 results). The weight quantized models are represented by “w bit-width,” activation quantized models by “a bit-width” and “FP” represents 32-bit full precision model. Results for VGG weight and activation quantized models are available in Appendix C. We find that the trends for activation and weight quantization are highly dataset and architecture dependent. It is difficult to make generic recommendations for the adversary or the defender. We show the confusion matrices for different datasets and architectures in Appendix C, and the relevant ones can be consulted when making a decision.

VI. RESULTS

In this section we present the performance results of the proposed attacks and TREND-based ensembles.

A. Proposed Attacks

We evaluate the performance of our attacks on deep ensembles which are traditionally [47], [48] built with models that are trained independently and the diversity among members arises from the randomness of the initialization and of the training procedure. Fig. 6 compares the success rate of the proposed attack methods against D-AG [19], [32], [46] and EMPIR style attack [31]. Fig. 7 compares the performance of various attacks across different attack strengths. We see that the proposed A-GD attack significantly outperforms other methods at lower attack strengths. Other attacks get close to A-GD performance around attack strength of about 0.03 and higher. The A-GD attack is no more compute intense than D-AG but is significantly more effective (up-to 1.53× improvement in attack success rate). Having observed that A-GD attack achieves state-of-the-art performance, we use this attack for testing our ensembles in the subsequent sections and we recommend the use of A-GD as the benchmark for testing all future DNN ensembles.

B. Robust Ensemble Design

Ensembles built using the proposed method were evaluated using the attacks described in subsection III-B. Fig. 8 shows various ensembles and the corresponding clean accuracies and accuracies under attack. Clean accuracy refers to the accuracy of the ensemble, under no attack conditions. From Fig. 8 we see that the ensemble of FP-Q1-Q2, FP-Q1-Q2-HT, and FP-Q1-Q2-FP-V-A consistently outperform other ensembles with respect to adversarial robustness. This trend was expected from the transferability metric for Q1, Q2, and HT input quantized models, which shows that Q1, Q2, and HT have the lowest average transferability metric. The addition of FP model boosts the baseline accuracy. The trend holds for different datasets, as shown in Appendix E. Appendix E also details various ensemble combinations and their adversarial accuracies under attack, and the trends expected from the transferability metric.
Fig. 8. Accuracy of various ensembles (attack strength $\epsilon = 0.01$) under D-AG, U-GD, A-GD, and EMPIR style attack. From the plots, we see that models with low transferability form more robust ensembles. Note FP-V-A refers to a full precision VGG11 adam trained model. All other models were SGD trained ResNet18. Note across the various ensembles our A-GD attack is the most successful.

Fig. 9. Number of models in a TREND built ensemble versus adversarial robustness (% accuracy, higher is more robust) at $L_\infty$ bound of 0.01 on CIFAR-10 under different attacks. The robust ensembles TREND methodology choose for were 2-{HT ResNet18, Q2 ResNet18}, 3-{FP ResNet18, Q1 ResNet18, Q2 ResNet18}, 4-{FP ResNet18, Q1 ResNet18, Q2 ResNet18, FP VGG11 adam trained}, 5-{FP ResNet18, Q1 ResNet18, Q2 ResNet18, FP VGG11 adam trained, FP VGG11BN adam trained}. The hatched plot represents the corresponding FP ensemble performance for reference.

We also observe that adding more low TM models to the ensembles boosts adversarial robustness, while maintaining or improving clean accuracy. Fig. 9 illustrates this, it visualizes the performance of TREND built ensembles (solid plot) with FP ensembles (hatched plot) as reference. It shows that increasing the number of models in a TREND ensemble boosts adversarial robustness under the strongest attack, while improving clean accuracy. An ensemble, FP-Q1-Q2, designed using TREND is seen to be $1.30 \times$, $1.36 \times$, and $8.43 \times$ more robust than individual Q1, Q2, and FP models correspondingly at $\epsilon$ of 0.01. Further, the FP-Q1-Q2 model is $3.17 \times$ more robust than FP-FP-FP ensemble at $\epsilon$ of 0.01 with $\sim 2.1\%$ drop in clean accuracy.

VII. CONCLUSION

TREND is a methodology to systematically design an ensemble with improved adversarial robustness. In this article we analyze the effect of DNN architecture, input, weight and activation quantization on transferability. Our analysis suggests that the choice of optimizer is critical and our experiments show that the ResNet architecture is more susceptible to transfer attacks than the other architectures considered. Quantizing the inputs significantly reduces transferability when the inputs are quantized to low bit widths (one and two bits). Additionally, our experiments reveal that the effect of weight and activation quantization is highly dependent on the dataset. We also observe that transferability is asymmetric. If adversarial images transfer well from source to target, the vice versa need not necessarily be true. Additionally, wherever applicable, we offer guidelines for the construction of defense and attack models based on the transferability trends observed. We identify that current ensemble attacks are hampered by the “gradient domination” effect and propose two attack methods that overcome this to achieve SoTA attack performance. Our results clarify that the
adversarial robustness of an ensemble is indeed determined by how transferable an adversarial image is among the models in the ensemble. Using the transferability metric, we are able to construct ensembles with improved robustness. A TREND designed ensemble of FP-Q1-Q2 is seen to be 1.30, 1.36, and 8.43 more robust than individual Q1, Q2, and FP models correspondingly at $\epsilon$ of 0.01. Further, the FP-Q1-Q2 model is 3.17 more robust than FP–FP–FP ensemble at $\epsilon$ of 0.01 with $\sim 2.1\%$ drop in clean accuracy.

**APPENDIX A**

**ARCHITECTURE ANALYSIS**

Fig. 10(a) shows the number of adversarial images transferred from source to target for various architectures on CIFAR-100. Fig. 10(a) can be interpreted by analyzing the four quadrants, with each quadrant representing a family of source or target model architectures (ResNet or VGG variants). The top-right quadrant of Fig. 10(a) is lighter than the bottom-left quadrant. This implies that adversarial images generated on VGG are more transferable to ResNets than the other way around.

**APPENDIX B**

**INPUT QUANTIZATION**

Figs. 10(c), 11, and 12(a) and (b) show the variation in transferability due input quantization on CIFAR-10 and CIFAR-100 with ResNet and VGG as the base models.

**APPENDIX C**

**WEIGHT AND ACTIVATION QUANTIZATION**

Fig. 12 shows the confusion matrices for activation and weight quantized models on CIFAR-100. Fig. 12 and (d) use ResNet18 as their base models, while Fig. 12 and (f) use VGG11 as their base model.

**APPENDIX D**

**TRANSFERABILITY METRIC AND ATTACK STRENGTH**

In this section, we analyze the variation of transferability with respect to the attack strength $\epsilon$. We plot the number of images that transfer between different source-target models, referred to as $f_{st}$, for 20 different values of $\epsilon$ in Fig. 13. We observe that these curves cross one another and, therefore, the trends of transferability across models cannot be generalized from the observations at a single $\epsilon$. However, we are interested in plotting this curve with the fewest possible measurements, such that it would have good predictive power over a range of $\epsilon$ values. To do this, we note that each plot visually resembles the CDF of an exponential distribution. The same is true for the number of images generated on a model, denoted by $f_{ss}$. Hence, we characterize both functions in the following form:

$$f_{st}(\epsilon) = a(1 - e^{be^\epsilon})$$

$$f_{ss}(\epsilon) = a'(1 - e^{b'e^\epsilon})$$

Fig. 10. Average number (averaged over 5 seeds) of adversarial images transferred from source to target on CIFAR-100 dataset under PGD attack averaged over five differently seeded models for each architecture (a) and input quantization (b). (a) Architecture analysis, under PGD attack, 40 iterations. (b) Architecture analysis, under Carlini Wagner $L_2$ ($\kappa=30$) attack. (c) Input quantization analysis under PGD attack, ResNet18 base model.

Fig. 11. Input quantization analysis under Carlini Wagner $L_2$ attack ($\kappa=30$), ResNet18 base model.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
where $a$, $a'$, $b$, and $b'$ are the parameters for fitting the data, obtained experimentally. We find the parameters of the equation using a few datapoints. Fig. 13 shows the empirical curve obtained using 20 points, and the predicted curve, fit using four datapoints. We observe that they align well and we get <5% root mean square error (RMSE) for the fit.

Armed with these observations, model transferability as a function of attack strength $\epsilon$. Equation (5) captures transferability as a ratio of the number of images that transfer to the target to the number of images that were generated at the source. Equivalently our model of transferability normalizes $f_{st}$ by $f_{ss}$ and gives us the transferability metric, $TM$.

$$\begin{align*}
TM(\epsilon) &= \frac{f_{st}(\epsilon)}{f_{ss}(\epsilon)}.
\end{align*}$$

The transferability metric is a number between $[0, 1]$ and represents a quantitative measure of the transferability between a given pair of models. The constants in both equations capture the effects of dataset, architecture, input quantization, etc., and can be estimated with a few datapoints. For example, we use four datapoints to estimate the constants for various quantizations for CIFAR-10.

Table VII shows the standard error (RMSE) for the fit used to calculate the transferability metric. The average difference
between the predicted and actual values is 0.055 or 5.5% for CIFAR-10 dataset on ResNet18 base model.

APPENDIX E
RESULTS FOR DIFFERENT ENSEMBLES

We observe that Q1, Q2, and HT are robust models for CIFAR-10, CIFAR-100, and ImageNet datasets. The result for ensemble on CIFAR-10 and CIFAR-100 under different attacks are shown in Fig. 14.

REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. Hinton, “Imagenet classification with deep convolutional neural networks,” in Proc. Adv. Neural Inf. Process. Syst., vol. 25, 2012, pp. 1–9.
[2] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.
[3] G. Hinton et al., “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” IEEE Signal Process. Mag., vol. 29, no. 6, pp. 82–97, Nov. 2012.
[4] D. Andor et al., “Globally normalized transition-based neural networks,” in Proc. 54th Annu. Meeting Assoc. Comput. Linguistics, 2016, pp. 2442–2452.
[5] J. Bruna et al., “Intriguing properties of neural networks,” in Proc. Int. Conf. Learn. Representation, 2014.
[6] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in Proc. Int. Conf. Learn. Representation, 2015.
[7] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” in Proc. Int. Conf. Learn. Representation Workshop Track, 2017, pp. 1–14.
[8] G. E. Dahl, J. W. Stokes, L. Deng, and D. Yu, “Large-scale malware classification using random projections and neural networks,” in Proc. IEEE Int. Conf. Acoust., Speech and Signal Process., 2013, pp. 3422–3426.
[9] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, “Practical black-box attacks against machine learning,” in Proc. ACM Asia Conf. Comput. Commun. Secur., 2017, pp. 506–519.
[10] A. N. Bhagoji, W. He, B. Li, and D. Song, “Practical black-box attacks on deep neural networks using efficient query mechanisms,” in Proc. Eur. Conf. Comput. Vis., 2018, pp. 158–174.
[11] Y. Guo, Z. Yan, and C. Zhang, “Subspace attack: Exploiting promising subspaces for query-efficient black-box attacks,” in Proc. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 3820–3829.
[12] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, “Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models,” in Proc. 10th ACM Workshop Artif. Intell. Secur., 2017, pp. 15–26.
[13] N. Narodytska and S. P. Kasiviswanathan, “Simple black-box adversarial attacks on deep neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops, 2017, pp. 6–14.
[14] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, “Towards deep learning models resistant to adversarial attacks,” in Proc. Int. Conf. Learn. Representation, 2017.
[15] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in Proc. IEEE Symp. Secur. Privacy, 2017, pp. 39–57.
[16] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “Deepfool: A simple and accurate method to fool deep neural networks,” in Proc. Conf. Comput. Vis. Pattern Recognit., 2016, pp. 1–9.
[17] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in Proc. IEEE Eur. Symp. Secur. Privacy, 2016, pp. 372–387.
[18] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh, “Ead: Elastic-net attacks to deep neural networks via adversarial examples,” in Proc. Assoc. Adv. Intell. Intell., 2018, pp. 10–17.
[19] F. Tramer, N. Carlini, W. Brendel, and A. Madry, “On adaptive attacks to adversarial example defenses,” in Proc. 34th Conf. Neural Inf. Process. Syst., 2020, pp. 1–13.
[20] A. Athalye, N. Carlini, and D. A. Wagner, “Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples,” in Proc. Int. Conf. Learn. Representation, 2018, pp. 274–283.
[21] N. Papernot, P. D. McDaniel, and I. J. Goodfellow, “Transferability in machine learning: From phenomena to black-box attacks using adversarial samples,” Comput. Res. Repository, 2016, arXiv:1605.07277.
[22] Y. Liu, X. Chen, C. Liu, and D. X. Song, “Delving into transferable adversarial examples and black-box attacks,” in Proc. Int. Conf. Learn. Representation, 2017, pp. 1–14.

![Fig. 14. Results for ensembles under different attacks for CIFAR-100. (a) CIFAR-100 gradient average. (b) CIFAR-100 unanimous gradient direction.](image-url)
[23] F. Tramèr, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, “The space of transferable adversarial examples,” Comput. Res. Repository, 2017, arXiv:1704.03453.

[24] L. Wu et al., “Understanding and enhancing the transferability of adversarial examples,” 2018, arXiv:1802.09707.

[25] L. Breiman, “Bagging predictors,” Mach. Learn., vol. 24, no. 2, pp. 123–140, Aug. 1996.

[26] Y. Freund and R. E. Schapire, “Game theory, on-line prediction and boosting,” in Proc. 9th Annu. Conf. Comput. Learn. Theory, 1996, pp. 325–332.

[27] Y. Freund and R. E. Schapire, “Experiments with a new boosting algorithm,” in Proc. 13th Int. Conf. Int. Conf. Mach. Learn., 1996, pp. 148–156.

[28] U. Naftaly, N. Intrator, and D. Horn, “Optimal ensemble averaging of neural networks,” Network: Comput. Neural Syst., vol. 8, pp. 283–296, 1999.

[29] M. Abbasi and C. Gagné, “Robustness to adversarial examples through an ensemble of specialists,” 2017, arXiv:1702.06856.

[30] W. Xu, D. Evans, and Y. Qi, “Feature squeezing: Detecting adversarial examples in deep neural networks,” in Proc. Netw. Distrib. Syst. Secur., 2020, pp. 1–12.

[31] S. Sen, B. Ravindran, and A. Raghunathan, “Empir: Ensembles of mixed precision deep networks for increased robustness against adversarial attacks,” in Proc. Int. Conf. Learn. Representations, 2020, pp. 1–12.

[32] W. He, J. Wei, X. Chen, N. Carlini, and D. Song, “Adversarial example defense: Ensembles of weak defenses are not strong,” in Proc. 11th USENIX Workshop Offensive Technol., W. Enck and C. Mulliner, Eds., 2017, pp. 1–11.

[33] A. Krizhevsky et al., “Learning multiple layers of features from tiny images,” Univ. Toronto, Toronto, ON, Canada, Tech. Rep. TR-2009, 2009.

[34] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. Conf. Comput. Vis. Pattern Recognit., 2009, pp. 248–255.

[35] J. F. Kolen and J. B. Pollack, “Back propagation is sensitive to initial conditions,” in Proc. Adv. Neural Inf. Process. Syst., vol. 3, 1991, pp. 860–867.

[36] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, “On the importance of initialization and momentum in deep learning,” in Proc. Int. Conf. Mach. Learn., 2013, pp. 1139–1147.

[37] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[38] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” in Proc. Int. Conf. Learn. Representations, 2015.

[39] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 1026–1034.

[40] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in Proc. Conf. Comput. Vis. Pattern Recognit., 2017, pp. 2261–2269.

[41] S. Zagoruyko and N. Komodakis, “Wide residual networks,” in Proc. Brit. Mach. Vis. Conf., 2016.

[42] D. Wu, Y. Wang, S.-T. Xia, J. Bailey, and X. Ma, “Skip connections matter: On the transferability of adversarial examples generated with ResNets,” in Proc. Int. Conf. Learn. Representations, 2020.

[43] D. Petrov and T. Hospedales, “Measuring the transferability of adversarial examples,” 2019, arXiv:1907.06291.

[44] P. Panda, I. Chakraborty, and K. Roy, “Discretization based solutions for secure machine learning against adversarial attacks,” IEEE Access, vol. 7, pp. 70 157–70 168, 2019.

[45] S. Kodge, H. Chaudhary, and M. Sharad, “Low power image acquisition scheme using on-pixel event driven halftoning,” in Proc. IEEE Comput. Soc. Annu. Symp., 2017, pp. 260–265.

[46] T. Strauss, M. Hanselmann, A. Junginger, and H. Ulmer, “Ensemble methods as a defense to adversarial perturbations against deep neural networks,” 2017, arXiv:1709.03423.

[47] T. Dietterich, “Ensemble methods in machine learning,” in Proc. First Int. Workshop Multi. Classifier Syst., 2000, pp. 1–15.

[48] B. Lakshminarayanan, A. Pritzel, and C. Blundell, “Simple and scalable predictive uncertainty estimation using deep ensembles,” in Proc. Adv. Neural Inf. Process. Syst., 2017, pp. 6402–6410.

[49] W. Hanjie, L. Yongtuo, and C. Hongmin, “Learning transferable perturbations for image captioning,” in Proc. ACM Trans. Multimedia Comput., Commun., Appl., 2022, pp. 1–18.

[50] C. Hao et al., “More or less (mol): Defending against multiple perturbation attacks on deep neural networks through model ensemble and compression,” in Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. Workshops, 2022, pp. 645–655.

[51] B. Alex, K. Dimche, and B. Damian, “Saliency diversified deep ensemble for robustness to adversaries,” in Proc. AAAI-22 Workshop Adversarial Mach. Learn. Beyond, 2022.