

Dynamic and Diverse Transformations for Defending Against Adversarial Examples

Yongkang Chen
National Key Laboratory
of Science and Technology
on Information System Security
Beijing, China
yk.chen365@outlook.com

Ming Zhang*
National Key Laboratory
of Science and Technology
on Information System Security
Beijing, China
zm_stiss@163.com

Jin Li
National Key Laboratory
of Science and Technology
on Information System Security
Beijing, China
tianyi198012@163.com

Xiaohui Kuang
National Key Laboratory
of Science and Technology
on Information System Security
Beijing, China
xiaohui_kuang@163.com

Xuhong Zhang
School of Software Technology
Zhejiang University
Zhejiang, China
zhangxuhong@zju.edu.cn

Han Zhang
INSC&BNRist
Tsinghua University
Beijing, China
zhhan@tsinghua.edu.cn

Abstract—It is demonstrated that deep neural networks can be easily fooled by adversarial examples. To improve the robustness of neural networks against adversarial attacks, substantial research on adversarial defenses is being carried out, of which input transformation is a typical category of defenses. However, because the transformation also has an impact on the accuracy of clean examples, the existing transformation-based defenses usually adopt minor transformations such as shift and scaling, which limits the defense effect of the transformation to some extent. To this end, we propose a method by using dynamic and diverse transformations for defending against adversarial attacks. Firstly, we constructed a transformation pool that contains both minor and major transformations (e.g., flip, rotate). Secondly, we retrained the model with the data transformed by major transformations to ensure that the performance of model itself is not affected. Finally, we dynamically select transformations to preprocess the input of the model to defend against adversarial examples. We conducted extensive experiments on MNIST and CIFAR-10 datasets and compared our method with the state-of-the-art adversarial training and transformation-based defenses. The experimental results show that our proposed method outperforms the existing methods, improving the robustness of the model against adversarial examples greatly while maintaining high accuracy on clean examples. Our code is available at <https://github.com/byerose/DynamicDiverseTransformations>.

Index Terms—neural networks, adversarial examples, dynamic and diverse transformations, transformation-based defenses

I. INTRODUCTION

Deep neural networks (DNN) have emerged as a critical tool for a variety of tasks including image recognition [1], natural language processing [2] and speech recognition [3]. Nevertheless, the DNN model is easily manipulated [4], [5] by adversarial examples, which are difficult to distinguish from natural examples and can result in incorrect predictions

* Ming Zhang is corresponding author.

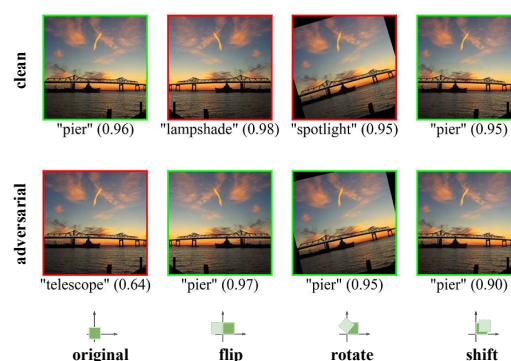


Fig. 1. Classification results of one clean examples and one corresponding adversarial example after applying different transformations. The images with green boxes are correctly classified, and the images with red boxes are incorrectly classified.

being made by the networks. In this regard, neural networks may pose a potential risk to intelligent systems, which have been widely used in real-world applications. An autonomous vehicle, for instance, may misuse traffic signs if its perception system is manipulated [6], [7] by adversarial examples, resulting in a negative effect on its physical system based on DNN. As a consequence of this phenomenon, more and more researchers are attempting to understand DNN's adversarial vulnerability [8], [9] and developing methods [10], [11], [12], [13] to defend against it.

Existing methods for improving the robustness of neural networks are typically used during training or inference. During training, defense methods attempt to harden the model itself in order to improve adversarial robustness. Adversarial training

[14], [15] is a typical strategy that uses a combination of adversarial and clean examples [16] to train neural networks. Adversarial training has been demonstrated to be effective, particularly on MNIST and CIFAR-10 [17]. However, adversarial training might also reduce performance on clean examples. In addition to adversarial training, data augmentation during training [18], [19] is also useful to improve the adversarial robustness by increasing diversity of the training data.

For defenses in inference, the basic goal is to eliminate the adversarial perturbations such that the input examples are no longer adversarial. Among them, transformation-based defenses [20], [21], [22], [23] work by preprocessing images using transformations before calling the model to classify images. For example, JPEG Compression [24] and Image Resizing [25], have both been shown to be effective for resisting adversarial examples. Compared with the defense methods based on training, these defense methods are computationally effective and easy to integrate with other defenses. However, the traditional transformation-based defenses usually use some simple transformations that are not strong enough and cannot build a universal defense against adversarial attacks. Moreover, the attacker can reverse the transformation if the transformation operation is differentiable.

To boost the transformation-based defenses, we focus on an important phenomenon: some transformations have high accuracy for both adversarial and clean examples, whereas some are effective for adversarial but fail on clean examples. This intuition is illustrated in Fig.1, shift can remove the confrontation of the adversarial examples, as can flip and rotation. The former has no effect on the classification of clean examples, which is called minor transformation, whereas the latter causes an incorrect classification of clean examples, which is called major transformation. Given the drop in classification accuracy caused by major transformations, traditional transformation-based defenses frequently use minor transformations [26] to remove the fooling ability of adversarial examples. Motivated by this observation, we propose a innovative transformation-based defense with dynamicity and diversity that can defend against adversarial attacks while maintaining the accuracy of clean examples. We specifically merge defense methods of training and inference. During training, major transformations are used to retrain the model so that it can adapt to major transformations. During inference, we dynamically choose a combination of transformations that includes both major and minor transformations, acting jointly on the input examples to defend against adversarial examples. Dynamic and diverse combinations of transformations bring obfuscated gradients and make it the attacker more challenging to exploit the transformation.

Unlike adversarial training, which reduces classification accuracy on clean examples, our method retains high accuracy on clean examples by retraining with major transformations. And we empirically show that our method offers a universal and effective defense in both gray-box and black-box scenarios. Achieved by optimizing the number of transformations used in

inference, our proposed method can provide a superior defense against adversarial attacks compared with previous methods.

The contributions of this paper can be summarized as follows:

- We systematically evaluated the influence of transformations on the classification accuracy and further divided them into major and minor transformations. We use these two categories of transformations simultaneously to design our defense method. As far as we know, we are the first to use the major transformations to defend against adversarial examples.
- We proposed a method by using dynamic and diverse transformations for defending against adversarial attacks.
- We conducted extensive experiments on MNIST and CIFAR-10 to verify the effectiveness of our method. The results show that our proposed method is obviously superior to the existing adversarial training and transformation-based defenses.

II. RELATED WORK

A. Adversarial Attacks

Szegedy et al. [14] proposed the initial work that adversarial examples would lead to misclassification of the deep neural networks. The adversarial examples can be generated by adding noise according to the adversarial gradient, such methods include the fast gradient sign method (FGSM) [27], the projected gradient descent (I-FGSM) method [17], and the Jacobian-based saliency map attack (JSMA) method [28]. The optimization-based attacks, such as Carlini and Wagner (CW) method [11], consider adversarial attacks as an optimization problem with the optimization objective of minimizing adversarial noise. Inspired by the linear classification model, DeepFool [29] aims to find the smallest perturbation that leads to misclassification by projecting the image to the nearest separation hyperplane. The attacks mentioned above compute the adversarial perturbations that can fool the target model on specific images. Moosavi-Dezfooli et al. [30] focus on computing image-agnostic perturbations, which can fool the model on any image with high confidence.

In another scenario, the attacker does not have access to the target model's parameters and gradients, and the methods among them can be divided into transfer-based and query-based black-box attacks. Transfer-based attacks generate adversarial examples using a source model and then transfer them to a target model. Dong et al. [31] proposed the Translation-Invariant iterative FGSM (TI²-FGSM). In this method, the untranslated gradient maps are convolved with a predefined kernel to reduce the computational complexity. Xie et al. [32] proposed the diverse inputs iterative FGSM (DI²-FGSM), which improve the transferability of adversarial examples by creating diverse input patterns. For query-based attacks, the essence of the problem is to determine the attack direction using gradient estimates or random searches based on the exact feedback of subsequent inquiries. The boundary attack [33] (BA) modified the intermediate examples on the

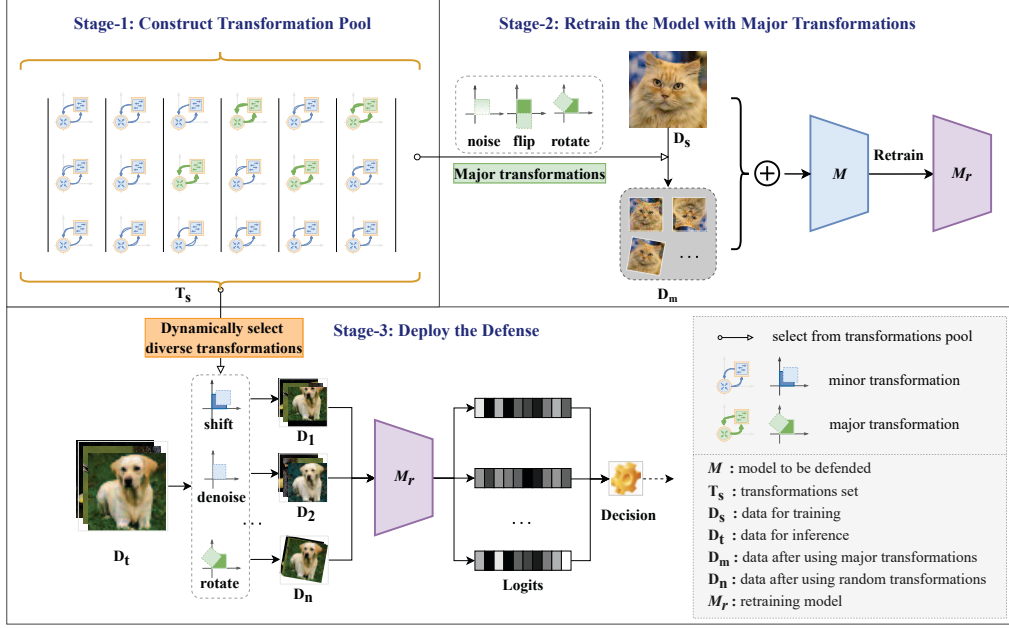


Fig. 2. Our proposed defense method consists of three stages. In stage-1, the transformation pool is constructed. In stage-2, the model is retrained with examples transformed by major transformations. In stage-3, the retrained model is used to defense against adversarial examples by using dynamic and diverse transformations.

decision boundary so that the perturbation norm steadily drops by random walk while maintaining the incorrect classification. The simple black-box attack [34] (SimBA) randomly selects a vector from a predefined orthonormal basis and either adds or subtracts it from the target image.

B. Adversarial Defenses

Many defense methods have been proposed to make the models more robust. The transformation-based defense methods aims to restore the adversarial examples to the clean examples. Das et al. [24] proposed a JPEG compression (JPEG) method for effective preprocessing defense. The high-frequency signal components are removed using a color and frequency domain transform in the image's square blocks, which are insensitive to the human visual system, while preserving the low-frequency signal components. This is similar to selective image blurring, which helps to remove the adversarial examples' perturbations. Xie et al. [25] proposed using randomization to mitigate the adversarial attacks. Randomization operations include random resizing, which adjusts the image to a random size, and random padding, which randomly fills zeros around the image. The method is quick to develop, needs little processing, and integrates well with other defenses. Kou et al. [35] proposed to train a separate distribution classifier to differentiate the softmax outputs of transformed inputs for boosting transformation-based defenses.

Adversarial training is currently the most effective strategy to improve adversarial robustness, which defends against adversarial attacks by using generated adversarial examples

in training [27] and clean examples. PGD-based adversarial training [17] is practically thought to be relatively effective among them. Some other methods [36], [37] combine adversarial training with external data sources. Zhang et al. proposed TRADES [38] to simultaneously control the boundary error and natural error to improve the trade-off by decomposing the adversarial prediction error into these two components. That is, it moves the classifier's decision boundary away from the example by minimizing the difference between the prediction of natural and adversarial examples. The method won first place in the NeurIPS 2018 Adversarial Vision Challenge.

III. METHODOLOGY

In the methodology section, we first outline our method and then describe the design of each module in stages. This method adopts the process of data augmentation during training and random transformations with dynamicity and diversity during inference, as shown in Fig. 2 and explained in the following:

a) Stage-1: Construct a transformation pool T_s and group all the transformations according to their different attributes.

b) Stage-2: All transformations are divided into major and minor transformations based on their influence on the classification accuracy of clean examples. Then retrain the model with the major transformations that generate some sets of input examples D_m .

c) Stage-3: Deploy transformation pool and retraining model in inference. This stage consists of two parts. In the front part, dynamically select transformations to preprocess

TABLE I
THE TRANSFORMATION POOL. THE TRANSFORMATION POOL IS
COMPOSED BY NINE GROUPS: EACH GROUP CONSISTS OF DIFFERENT
TYPES OF TRANSFORMATIONS.

Groups	Transformation Details
Adjustment	<i>brightness, sharpness, contrast, average blur, motion blur</i>
Affine	<i>vertical compress, both compress, vertical stretch, both stretch</i>
Compression	<i>jpeg quality (10, 20, 40, 60, 80)</i>
Denoise	<i>tv_chambolle, tv_bregman, bilateral, wavelet, nl_means</i>
Filter	<i>sobel, gaussian, rank, median, minimum, maximum, percentile</i>
Geometric	<i>swirl, scale, perspective, piecewise, elastic</i>
Morphology	<i>erosion, dilation, opening, closing</i>
Rotate	<i>rotate (5, 10, 15), horizontal flip</i>
Shift	<i>left, right, up, down, top_left, top_right, bottom_left, bottom_right</i>

the input of the model. For the back part, design a decision strategy to make the best prediction.

A. Transformation Pool

In stage-1, we construct a transformation pool consisting of 9 transformation groups with 47 transformations, as shown in TABLE I. Each transformation group contains a subset of transformations that are conceptually similar but differ in type. We evaluate each transformation individually on two datasets to guarantee that all the transformations work properly and can eliminate the confrontation of adversarial examples. Moreover, we also evaluate the influence of transformations on the classification accuracy of clean examples. For those whose test accuracy drops greatly, that is major transformations, are used to retrain the model.

B. Major and Minor Transformations

In stage-2, to further boost defense against adversarial examples, we use a process of retraining the model and explain it as the following. The model is trained using major transformations. The traditional transformation-based defenses rarely use major transformations due to the impact of major transformations on classification accuracy.

a) Minor Transformations: Research has shown that transformations are helpful for removing confrontation of adversarial examples, but the effect of transformations on clean examples has been overlooked. Accordingly, we examine a variety of image transformation methods currently used in the computer vision field as shown in Fig.3. Many transformations can withstand adversarial attacks while not affecting clean

examples significantly, which makes them highly useful for transformation-based defenses. We refer to these types of transformations as minor transformations since they are often an unobserved process. It can be seen in stage-3 of Fig. 2 that minor transformations are used directly for defense in inference.

b) Major Transformations: In addition to minor transformations, we found a few more transformations that are not originally suitable for defense inference. However, these transformations have a significant effect on the accuracy of clean examples even if they are successful at defending against adversarial attacks. We therefore use this type of transformation to retrain the model. This hardens the model and makes it more resilient to major transformations, as shown in Fig. 3. For example, the use of "shift up" reduces classification accuracy, but the performance improves after the model has been retrained. This type of transformation results in obvious changes to the image, which is why it is called a major transformation.

c) Retraining: In our proposed method, we adopt a similar retraining method on two datasets and models. Given a set of major transformations in the previous section, randomly select a transformation applied for each batch of input images, so that the model is familiar with the transformations used during inference. For MNIST, we train for an additional 100 epochs. For CIFAR-10, we train for 200 epochs. All training processes use SGD optimizer.

C. Dynamicity and Diversity

In stage 3, transformation of the images can limit the intensity of adversarial perturbations before the images are fed into the model. When several transformations are applied to the input, the transformations can work together to eliminate the adversarial examples' confrontation. It is worth mentioning that following retraining, major and minor transformations can be compatible in the combination of transformations. Diverse transformation combinations are crucial to boosting defense. Increasing the number of transformations in front of the model's input improves the defense performance of the adversarial examples. And the more transformations utilized in the retraining process, the more robust the model becomes. In this regard, our goal is to broaden the range of transformation combinations as much as possible in order to maximize defense efficiency.

Dynamity plays an important role in improving adversarial defense performance by preventing attackers from exploiting transformation. The attacker's access to obtain the model's gradient is an crucial factor in the attack's success. We use a variety of randomization operations in our method to avoid this problem, which includes randomized transformation, such as ratio of "shift up" randomly selecting from $U[0, 0.1]$, and dynamic transformation combination, in which the transformations applied to different inputs are uncertain. In our proposed method, major and minor transformations work along with dynamicity, and gradients of the model are obfuscated and this prevents the transformation process from being exploited.

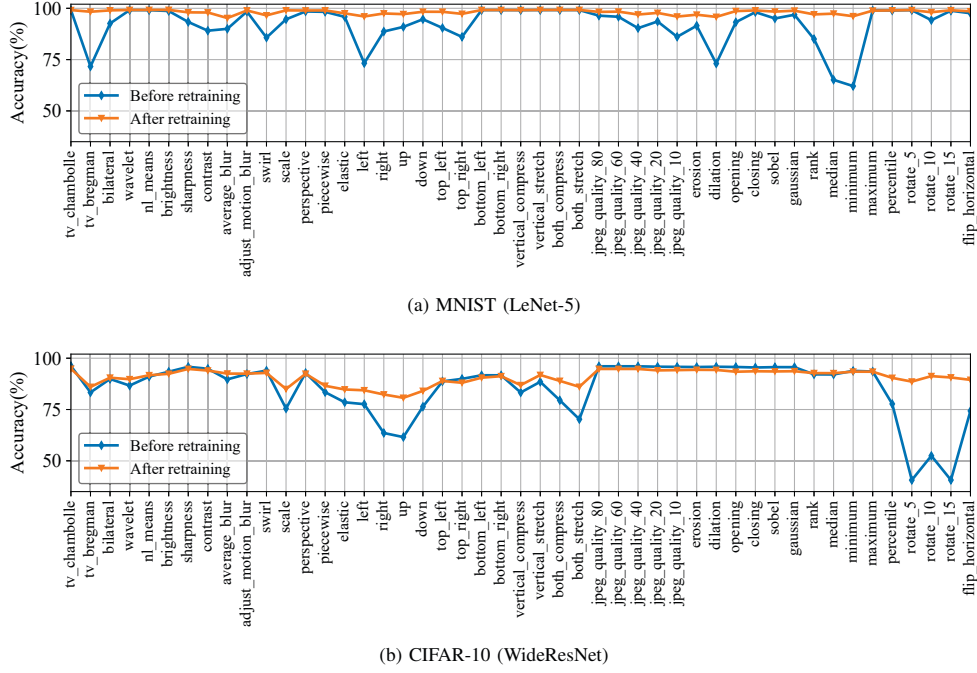


Fig. 3. Classification accuracies of examples after using 47 different transformations on MNIST and CIFAR-10 models.

TABLE II
ACCURACIES (%) OF ADVERSARIAL EXAMPLES BY USING DIFFERENT
DECISION STRATEGIES. THE ADVERSARIAL EXAMPLES ARE CRAFTED
BY USING FGSM.

Decision Strategy	Accuracy(%)	
	MNIST	CIFAR-10
Random	92.95	72.94
Majority-Voting	96.01	77.18
Average-Weights	96.01	77.22
AWMV	96.03	78.21

Although diversity and dynamicity are different, they are intertwined and act in concert on defense against adversarial examples. Based on the analysis, our method performs random transformations defense with both dynamicity and diversity in inference to counter adversarial perturbations. At each iteration in inference, some transformation groups are randomly selected from the transformation pool, and some transformations are randomly selected from each group. After selecting the transformation applied to an image, randomize the transformation parameters and then the transformations takes effect on the image.

D. Decision Strategy

In Stage 3, more than one transformation applied on input examples brings the logits of multiple branches of data flow. These branches will then be used to make decision to reach

final prediction y_{out} . We implement and evaluate four decision strategies of multiple branches, which are explained as follows:

- **Random Strategy** randomly selects one logits from all transformation branches to reach the final output.
- **Average-Weights Strategy** gives a weight to each logits of the transformation branch, and the final output is the largest value in the weighted average of all transformation branches.
- **Majority-Voting Strategy** selects a candidate with largest logits from each transformation branch, and of all the candidates, the one who receives the most votes is the prediction result.
- **Average-Weights and Majority-Voting (AWMV) Strategy** combines average-weights and majority-voting strategy. Each logits of the transformation branch gets a weight. The candidates are chosen from the output with the weighted average, and the candidate with the most votes is returned. The decision strategy can be formulated as follows:

$$y_{out} = \underset{g}{\operatorname{argmax}} \left(\frac{1}{g * k} \sum_{i=0}^g \sum_{j=0}^k f(t_{ij}(x)) \right) \quad (1)$$

where g denotes number of transformation groups and k denotes number of transformations. t_{ij} denotes transformation of the ij -th selection.

Experiments as shown in TABLE II, reveal that AWMV is clearly superior to other decision strategies, thus we employ AWMV to make final decision in our proposed method.

IV. EXPERIMENTS

A. Experimental Setup

a) *Datasets and models*: To evaluate our defense method, we use the MNIST and CIFAR-10 datasets. These datasets' images are categorized into 10 classes. For the MNIST, we use LeNet-5 [39] architecture that has 98.7% test accuracy. For the CIFAR-10, we use WideResNet [40] architecture that has 95.6% test accuracy. On MNIST, the AlexNet [41] architecture is utilized as a substitute model for transfer attacks, while on CIFAR-10, the ResNet-50 [1] architecture is utilized. WideResNet's depth and widen factors are set to 28×20 .

b) *Attacks and compared defenses*: For fair comparison, all experiments are run on two NVIDIA RTX 2080TI GPUs, and all methods are implemented using PyTorch. We consider several mainstreamed white-box attack methods, including FGSM [27], I-FGSM [17], CW [11], DeepFool [29], JSMA [28]. FGSM and I-FGSM use default perturbation budget $\epsilon = 8/255, 16/255, 32/255$ for MNIST and $\epsilon = 4/255, 8/255, 16/255$ for CIFAR-10 respectively. Furthermore, we also consider black-box attack methods, such as BA [33], SimBA [34], TI²-FGSM [31], DI²-FGSM [32]. Experiments present the evaluation performance against the untargeted attacks in L_∞ or L_2 norm conditions. We use the implementations of TI²-FGSM and DI²-FGSM provided by the Torchattacks [42] library. For other attack methods, we use the implementations from Adversarial Robustness Toolbox [43] library. We compare our methods with two transformation-based defenses including JPEG [24] and RRP [25], as well as an adversarial training method TRADES [36]. All the implementations of defense methods come from the original authors.

B. Setting the Hyperparameters

To ensure that our defense is as effective as possible, we must first define the hyperparameters in the defense method. The number of transformation groups g , the number of transformations k picked by each group, and the parameter θ of each transformation are all hyperparameters. For 47 transformations, there will be a large search space if all hyperparameters are optimized. Furthermore, optimizing θ has an influence on the transformation process' randomness. As a result, we optimize the parameters k .

We traverse the number of transformations to find the optimal hyperparameter. We use the accuracy of model classification as objective functions after defense deployment. k is the hyperparameter to be optimized, with k 's search space being $[1, 6]$. Because there are many random processes in the optimization process, each data point is averaged from ten experimental results. In this experiment, we use FGSM as our attack method. The maximum perturbation of each pixel is set to $\epsilon = 16/255$ on MNIST and $\epsilon = 8/255$ on CIFAR-10. Experiments show that increasing the values of the hyperparameter improves the deployed defensive model's classification accuracy, as shown in Fig 4. For MNIST, the clean accuracy and adversarial accuracy tend to be stable when

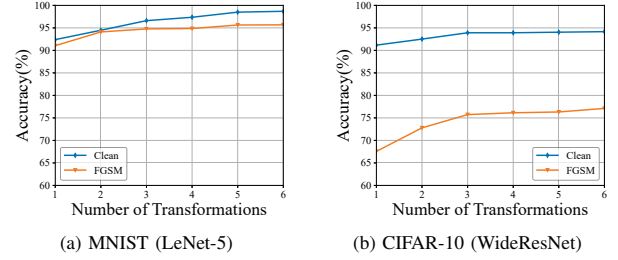


Fig. 4. Classification accuracies (%) of adversarial examples on MNIST and CIFAR-10 models when using different number of transformations. The adversarial examples are crafted by FGSM.

$k = 5$. For CIFAR-10, this happens when $k = 3$. To trade the number of transformations off against preprocessing costs, we set $k = 5$ on MNIST and $k = 3$ on CIFAR-10. The following experiments are all based on the above hyperparameter.

C. Defense Results: Gray-box Scenario

We build grey-box scenarios to evaluate our defense method, on the MNIST and CIFAR-10 dataset. The scenario implies that the attacker is unaware of the defense method and has access to only the model's parameters and architecture. Despite the fact that this is a grey-box scenario, we generate adversarial examples using the white-box attack methods, based on the model which is trained with clean examples. Compared with three other defense methods, we test these adversarial examples. TABLE III compares our proposed method with JPEG [24], RRP [25] and TRADES [36]. When identifying with clean examples, we first observe that the proposed method can maintain classification accuracy at the same level as the original model. In particular, TRADES introduce 9% drop on clean examples on CIFAR-10. We also see that the proposed method significantly helps to improve defense performance against untargeted attacks and outperform all other methods. Traditional JPEG and RRP defense is not always effective. For Deepfool and CW, which can achieve beyond 40% success rate, JPEG and RRP are respectively each only effective for one of them. When combating with FGSM and I-FGSM, which generate adversarial examples with lower perturbation intensity, our proposed methods outperform the other three methods by around 8% for CIFAR-10.

D. Defense Results: Black-box Scenario

This scenario assumes that the attacker only knows inputs and outputs of the service but has no knowledge of both the model and the defense methods. In our evaluation, transfer-based and query-based attacks are used as tools for black-box scenarios. Specifically, we use BA [33] and SimBA [34] as query-based attacks. For query-based attacks, we consider limiting the maximum number of queries to 1000. Adversarial examples with transferability are generated from AlexNet on MNIST and ResNet50 on CIFAR-10. Small perturbations crafted with traditional white-box attack methods have no

TABLE III
CLASSIFICATION ACCURACIES (%) OF CLEAN AND ADVERSARIAL EXAMPLES WHEN USING DIFFERENT DEFENSE METHODS. THE ADVERSARIAL EXAMPLES ARE CRAFTED IN THE GRAY-BOX SCENARIO, I.E., THE ATTACKER HAS THE KNOWLEDGE OF THE MODEL BUT HAS NO KNOWLEDGE OF THE DEFENSE METHODS.

Gray-box	Method	Clean	FGSM			I-FGSM			CW(L_2)	DeepFool	JSMA
			$\epsilon=8$	$\epsilon=16$	$\epsilon=32$	$\epsilon=8$	$\epsilon=16$	$\epsilon=32$			
MNIST (LeNet-5)	No defense	98.93	97.56	94.37	68.40	97.53	94.35	60.17	47.00	60.25	0.24
	JPEG	98.98	97.70	94.92	69.65	97.72	94.89	63.84	48.23	96.69	22.06
	RRP	97.22	95.12	89.86	50.72	94.79	89.73	49.29	86.15	68.02	76.07
	TRADES	97.65	97.19	96.56	94.74	97.25	96.63	95.30	92.55	95.59	89.06
	Our method	98.31	98.21	98.05	97.65	98.15	98.00	97.68	97.19	97.86	96.50
Gray-box	Method	Clean	FGSM			I-FGSM			CW(L_2)	DeepFool	JSMA
			$\epsilon=4$	$\epsilon=8$	$\epsilon=16$	$\epsilon=4$	$\epsilon=8$	$\epsilon=16$			
CIFAR-10 (WideResNet)	No defense	96.22	58.58	44.81	24.09	30.08	16.23	9.13	53.71	54.76	59.27
	JPEG	93.05	59.61	40.93	22.00	40.01	18.63	9.76	70.23	65.44	70.83
	RRP	94.41	70.08	60.12	42.79	71.74	59.84	36.46	79.29	77.69	76.38
	TRADES	87.80	76.17	72.87	59.61	75.99	74.16	66.27	76.85	77.09	75.86
	Our method	94.47	83.80	77.84	58.70	83.16	81.45	67.68	86.58	86.65	86.64

TABLE IV
CLASSIFICATION ACCURACIES (%) OF CLEAN AND ADVERSARIAL EXAMPLES WHEN USING DIFFERENT DEFENSE METHODS. THE ADVERSARIAL EXAMPLES ARE CRAFTED IN THE BLACK-BOX SCENARIO, I.E., THE ATTACKER ONLY KNOWS INPUTS AND OUTPUTS OF THE SERVICE BUT HAS NO KNOWLEDGE OF BOTH THE MODEL AND THE DEFENSE METHODS.

Black-box	Method	Clean	Query-Based		Transfer-Based	
			BA	SimBA	TI ² -FGSM	DI ² -FGSM
MNIST (LeNet-5)	No defense	98.93	32.26	46.60	70.53	60.85
	JPEG	98.98	96.16	75.88	72.57	63.77
	RRP	97.18	72.94	58.49	34.46	37.64
	TRADES	97.65	96.10	95.38	95.18	95.00
	Our method	98.32	97.98	97.56	97.39	97.26
CIFAR-10 (WideResNet)	No defense	96.22	63.73	71.79	54.86	43.67
	JPEG	93.05	76.05	78.90	53.87	44.87
	RRP	94.05	81.76	76.31	54.00	47.40
	TRADES	87.80	77.22	75.76	59.98	58.52
	Our method	94.64	87.32	85.16	63.96	59.30

satisfactory transferability, so we use previously proposed TI²-FGSM [31] and DI²-FGSM [32] as transfer-based attacks. Because original implementation of TI²-FGSM and DI²-FGSM specially designed on ImageNet, we modify the channels and kernel size to accommodate the adopted datasets.

The comparison with other defense methods is shown in Table IV. Our method achieves more effective defense in the black-box scenario. In particular, our methods significantly improve the classification accuracy against black-box attack methods in comparison to previous TRADES. When combating with BA and SimBA, which generate adversarial examples with limited queries, our proposed methods outperform the other three methods by around 10% for CIFAR-10. We also see that the traditional transformation-based defenses are extremely vulnerable in the face of transfer attacks. When defending against transfer-based attacks, RRP even lowers

adversarial accuracy compared to no defense.

V. CONCLUSION

In this paper, we propose a method by using dynamic and diverse transformations for defending against adversarial attacks. Firstly, we construct a transformation pool with diverse transformations. Secondly, all transformations are divided into two categories (minor or major) based on the classification accuracy on clean examples, and the major transformations are utilized to retrain the model. Finally, transformations are dynamically selected to preprocess the input and the final prediction of the model is decided by an average-weights and majority-voting strategy. Experiments demonstrate that our proposed method can improve defense efficiency while maintaining the classification accuracy on clean examples.

REFERENCES

- [1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [3] J. Park, Y. Boo, I. Choi, S. Shin, and W. Sung, "Fully neural network based speech recognition on mobile and embedded devices," *Advances in neural information processing systems*, vol. 31, 2018.
- [4] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *International Conference on Machine Learning*. PMLR, 2018, pp. 2137–2146.
- [5] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *Artificial intelligence safety and security*. Chapman and Hall/CRC, 2018, pp. 99–112.
- [6] E. Ohn-Bar and M. M. Trivedi, "Looking at humans in the age of self-driving and highly automated vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 90–104, 2016.
- [7] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao, "Adversarial sensor attack on lidar-based perception in autonomous driving," in *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2019, pp. 2267–2281.
- [8] A. Ross and F. Doshi-Velez, "Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.
- [9] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," *Advances in neural information processing systems*, vol. 32, 2019.
- [10] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," in *International Conference on Learning Representations*, 2018. [Online]. Available: <https://openreview.net/forum?id=rkZvSe-RZ>
- [11] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 39–57.
- [12] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *Proceedings of the 36th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, K. Chaudhuri and R. Salakhutdinov, Eds., vol. 97. PMLR, 09–15 Jun 2019, pp. 1310–1320. [Online]. Available: <https://proceedings.mlr.press/v97/cohen19c.html>
- [13] X. Zhang, J. Wang, T. Wang, R. Jiang, and L. Zhao, "Robust feature learning for adversarial defense via hierarchical feature alignment," *Information Sciences*, 2020.
- [14] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.
- [15] F. Croce, S. Goyal, T. Brunner, E. Shelhamer, M. Hein, and T. Cemgil, "Evaluating the adversarial robustness of adaptive test-time defenses," *arXiv preprint arXiv:2202.13711*, 2022.
- [16] F. Tramèr, A. Kurakin, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "Ensemble adversarial training: Attacks and defenses," *arXiv preprint arXiv:1705.07204*, 2017.
- [17] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.
- [18] S.-A. Rebuffi, S. Goyal, D. A. Calian, F. Stimberg, O. Wiles, and T. A. Mann, "Data augmentation can improve robustness," *Advances in Neural Information Processing Systems*, vol. 34, pp. 29 935–29 948, 2021.
- [19] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "Augmix: A simple data processing method to improve robustness and uncertainty," *arXiv preprint arXiv:1912.02781*, 2019.
- [20] A. Prakash, N. Moran, S. Garber, A. DiLillo, and J. Storer, "Deflecting adversarial attacks with pixel deflection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8571–8580.
- [21] Y. Zhang and P. Liang, "Defending against whitebox adversarial attacks via randomized discretization," in *The 22nd International Conference on Artificial Intelligence and Statistics*. PMLR, 2019, pp. 684–693.
- [22] J. C. Pérez, M. Alfara, G. Jeanneret, L. Rueda, A. Thabet, B. Ghanem, and P. Arbeláez, "Enhancing adversarial robustness via test-time transformation ensembling," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 81–91.
- [23] W. He, B. Li, and D. Song, "Decision boundary analysis of adversarial examples," in *International Conference on Learning Representations*, 2018.
- [24] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, L. Chen, M. E. Kounavis, and D. H. Chau, "Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression," *arXiv preprint arXiv:1705.02900*, 2017.
- [25] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," *arXiv preprint arXiv:1711.01991*, 2017.
- [26] C. Guo, M. Rana, M. Cisse, and L. Van Der Maaten, "Countering adversarial images using input transformations," *arXiv preprint arXiv:1711.00117*, 2017.
- [27] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.
- [28] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS&P)*. IEEE, 2016, pp. 372–387.
- [29] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2574–2582.
- [30] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1765–1773.
- [31] Y. Dong, T. Pang, H. Su, and J. Zhu, "Evading defenses to transferable adversarial examples by translation-invariant attacks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4312–4321.
- [32] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille, "Improving transferability of adversarial examples with input diversity," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2730–2739.
- [33] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," *arXiv preprint arXiv:1712.04248*, 2017.
- [34] C. Guo, J. Gardner, Y. You, A. G. Wilson, and K. Weinberger, "Simple black-box adversarial attacks," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2484–2493.
- [35] C. Kou, H. K. Lee, E.-C. Chang, and T. K. Ng, "Enhancing transformation-based defenses against adversarial attacks with a distribution classifier," in *International Conference on Learning Representations*, 2019.
- [36] Y. Carmon, A. Raghuathan, L. Schmidt, J. C. Duchi, and P. S. Liang, "Unlabeled data improves adversarial robustness," *Advances in Neural Information Processing Systems*, vol. 32, 2019.
- [37] D. Hendrycks, K. Lee, and M. Mazeika, "Using pre-training can improve model robustness and uncertainty," in *International Conference on Machine Learning*. PMLR, 2019, pp. 2712–2721.
- [38] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International conference on machine learning*. PMLR, 2019, pp. 7472–7482.
- [39] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [40] C. Xiao, J.-Y. Zhu, B. Li, W. He, M. Liu, and D. Song, "Spatially transformed adversarial examples," *arXiv preprint arXiv:1801.02612*, 2018.
- [41] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, vol. 25, 2012.
- [42] H. Kim, "Torchattacks: A pytorch repository for adversarial attacks," *arXiv preprint arXiv:2010.01950*, 2020.
- [43] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig et al., "Adversarial robustness toolbox v1. 0.0," *arXiv preprint arXiv:1807.01069*, 2018.