2022 26th International Conference on Pattern Recognition (ICPR)
August 21-25, 2022, Montréal, Québec, Canada

Introducing Diversity In Feature Scatter Adversarial Training Via Synthesis

Satyadwyoom Kumar
Dept. Electronics and Communication
Netaji Subhas Institute of Technology
New-Delhi, India
satyadwyoom.ec18@nsut.ac.in

Apurva Narayan
Dept. Computer Science
The University of British Columbia
Kelowna, BC, Canada
apurva.narayan@ubc.ca

2022 26th International Conference on Pattern Recognition (ICPR) | 978-1-6654-9062-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICPR56361.2022.9956053

***Abstract*—In an attempt to understand how deep learning models interpret inputs, it has been found that they change their prediction when a carefully optimized imperceptible noise termed adversarial perturbation is added to the input data. Many researchers are focusing on developing methods to counter such effects, but such methods do not generalize well to adversarial test data. Recently, Feature-Scatter adversarial training has come up to solve such a problem, but this method uses the traditional adversarial training framework as its basis that cannot generate diverse perturbations.**

In this paper, we propose an approach that combines both the Feature-Scatter adversarial training and the generator-based adversarial training framework to optimally explore the adversarial data manifold achieving better robust generalization. We perform extensive experimentation across a wide variety of datasets such as Cifar10, Cifar100, and SVHN. Our framework significantly outperforms the state-of-the-art methods against both strong white-box attacks and black-box attacks.

I. Introduction

With the availability of large quantity of data, deep neural networks have achieved exceptional performance across multiple domains, solving wide variety of complex tasks related to automation, visual perception, and prediction. Therefore deep-learning methodologies must be both robust and precise such that they do not change their prediction under adversarial noises and correctly classify clean, adversarial inputs. However, it is observed that the model performance drastically decreases in the presence of adversarial perturbation.

In the pioneering work of [1], the author found that neural networks tends to change their prediction when a visually imperceptible, small in magnitude, and carefully optimized noise is added to the input. Motivated by the work of [1], there is an effort to model adversaries that help understand [2] vulnerabilities of deep neural networks. These developments across the AI community to generate perturbations in different settings have prompted researchers to identify factors contributing to adversarial perturbations and subsequently propose defense techniques that produce deep learning models robust to such adversarial perturbations. Out of many proposed defenses [3], [4], [5], [6], it has been found that the min-max optimization framework that incorporates adversarial examples into the model training process effectively reduces the model's vulnerability to adversarial perturbations and does not provide any false sense of robustness [7]. However, there is a disadvantage of decrease in the model's ability to generalize well against unseen adversarial data.

Robust generalization relates to achieving good performance on unseen adversarial data by an adversarially trained model [8] and is a point of higher weightage in the developement of robust deep learning. Recently, it has been found that this robust generalization of models is directly proportional to the amount of data available during training [8]. One such method that tries to solve the issue of robust generalisation is [9], where the author proposed an approach by plugging a probabilistic distance measure in a gradient-based adversary [6] instead of the traditional cross-entropy loss. This approach produces perturbations by using the relationship between data samples to improve the generalization of adversarially trained models without requiring a large amount of unlabeled training data which may not be possible for domains where data is scarce.

Approaches such as [10], [11] are proposed as an alternative to the gradient-based adversary of [6]. These approaches are based on the learning to learn framework, a generative framework proposed by [12] wherein a convolutional neural network is modeled as an adversary that learns to fool a classifier. Since most of these methodologies follow a "single-step" approach, they suffer from learning a sub-optimal adversary [11] and prone to looking in some fixed direction on the data manifold while searching for adversarial perturbations. [11] proposes a solution to this problem by introducing diversity into the perturbation-producing framework, but the proposed methods is computationally expensive and similar to [6] that lacks the ability of robust generalization.

Motivated by the idea proposed by [9] we develop a novel framework based on single step generation strategy that learns the relationship between different data-sample and effectively explores the adversarial space. Our contributions can be summarised as follows:

- We present a generative adversary that produces strong adversarial examples when compared to Feature-Scatter framework's [9] adversary.
- The proposed framework maximizes the distance between probability distributions while introducing diversity in generated perturbations at each time step.

- We incorporate our adversary in the model training process which produces a classifier which shows high robust-generalisation against strong adversarial attacks.
- We perform extensive experiments across multiple datasets such as Cifar10, Cifar100, and SVHN comparing our methods with chosen baselines.
- Our framework is generalizable across multiple datasets and has surpassed previous approaches in terms of accuracy when tested against a variety of strong black-box and white-box attacks.

II. Related Work

A. Gradient based methods

In this subsection, we discuss various approaches that use a hand-crafted optimization procedures utilizing backward propagated gradients to optimize adversarial perturbations. We start with [13] who present a single-step method known as the fast gradient sign method (FGSM) for quickly generating adversarial perturbations. The author [13] found that adversarial perturbations transfer to different architectures as these architectures learn a similar loss function that results in similar learnt parameters across these architectures. Subsequently, [6] propose an iterative version of FGSM attack for different $norms^{(l_1,l_2,l_\infty)}$ to produce much stronger adversarial examples. They formalize the problem of adversarial training as a min-max optimization problem where maximization relates to adversarial attack and minimization relates to model training to produce models that are robust to adversarial attacks.

Recently, [9] propose an approach to solving problems such as overfitting [14] and bias towards decision boundary existing in the adversarial training framework of [6]. The main idea behind their approach is that they view adversarial training from the data manifold perspective rather than the classification loss. Hence, maximize the learnt data manifold represented by an optimal transport distance between the clean and perturbed data. Such an approach results in both clean and robust generalization far greater than past works such as [6].

some other works discussing different properties of adversarial perturbations [2], [15], [16], [17], [18].

B. Generator based methods

In this subsection, we discuss some of the recent approaches that use a generator/convolutional neural network to either detect whether the input image is adversarial or generate individual sample-based, universal adversarial perturbations for image classifiers. We start with some works like [19], [20] who motivated by the idea that the adversarial data distribution is indistinguishable from the true data distribution, present a generator based framework. In their framework, the generator trains to generate adversarial data, and the discriminator network trains to recognize whether the incoming data is adversarial or not. Thus, performing adversarial and clean data classification.

[12] propose the learning to learn(L2L) framework, where a convolutional neural network uses the backpropagated gradient information to generate adversarial perturbations shifting from the hand-crafted optimization algorithms such as [6], [13]. Another work by [21] proposes a similar GAN [13] based approach where a generator learns to generate adversarial perturbation using only the input image to fool a classification model. In their work, the author proposes to train a classifier instead of a discriminator that learns from the generated perturbations to improve its robustness.

[11] observed that models trained with (L2L) [12] lack robustness against iterative attacks [6]. Hence, they propose a generator network that learns to generate and refine perturbations for an image through multiple iterations while simultaneously introducing diversity in the generated perturbation at each time step. However, similar to approaches discussed in the above subsection generator based methods also suffer from a similar problem of robust generalization and hence an important problem to explore.

III. Background

A. Optimal Transport (OT) Distance

The optimal transport (OT) distance comes from the optimal transport theory that deals with optimally transporting a probability distribution to another probability distribution. Thus, providing an alternative metric to train generative models [22]. The optimal transport distance between two given probability distributions *p* and *q* can be defined as:

$$D(p,q) = inf_{\gamma \in \Pi(p,q)} E_{x,y\sim\gamma} c(x,y) \tag{1}$$

Where x and y are two images, $\Pi(p,q)$ is the set of all joint distributions $\gamma(x,y)$ with $p(x)$ and $q(y)$ as marginals and $c(x,y)$ is the cost function which could use a euclidean or non-euclidean measure. [22] were the first to present the approach of coupling both the traditional GAN framework and optimal transport distance which they called *Earth-Mover distance*. Where $p(x)$ and $q(y)$ can be interpreted as piles of earth and $D(p,q)$ is the amount of mass that γ has to transport to convert all the points of distribution $q(y)$ to the points of distributions $p(x)$.

B. Feature Scatter Adversarial Training

Feature Scattering Adversarial Training technique [9] deals with utilizing the optimal transport distance explained above to generate adversarial data. Then maximizing the distance between the classification probabilities of clean data distribution represented by *u* and perturbed data distributions *v*, similar to *p* and *q* in equation 1. This generated perturbation δ is utilized in the traditional training setup. This is mathematically represented using the following equation:

$$\begin{aligned} &\text{minimize } \frac{1}{n}\Sigma_1^n L(f_\phi(x' = x+\delta), y) \\ &\text{s.t. } \delta \text{ maximize } D(u,v) \end{aligned} \tag{2}$$

Here n is the total number of data samples, f is the classification model parameterized by ϕ and δ is the adversarial perturbation that is added to the clean data x to generate the adversarial data $x^{'}$. δ maximizes the optimal transport distance

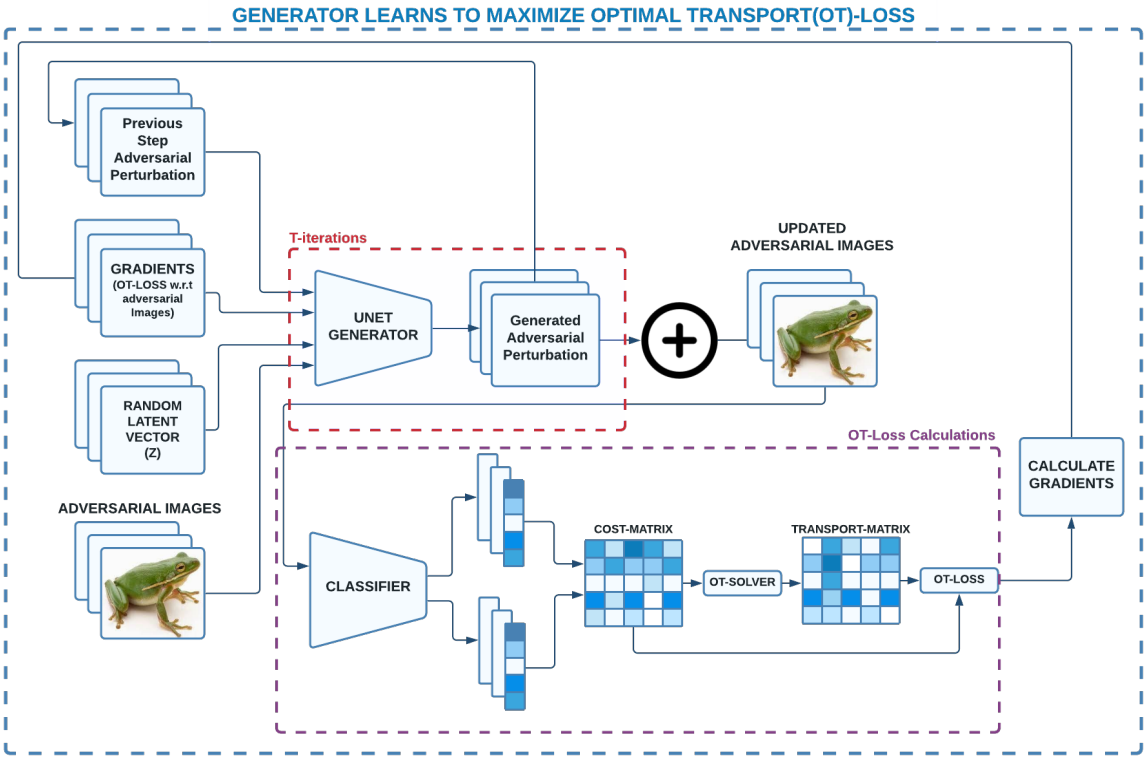

Fig. 1: FS-GAN: Generator training setup, where the generator utilises the data orientation captured through OT-loss to produce a diverse set of adversarial perturbations.

(calculated using sinkhorn solver) denoted by $D(u, v)$. $D(u, v)$ is defined by the following equation:

$$D(u,v) = inf_{\gamma \in \Pi(u,v)} \Sigma_{i=1}^{n} \Sigma_{j=1}^{n} T_{ij} \cdot c(x, x') \tag{3}$$

$$c(x_i, x_j^{'}) = 1 - \frac{f_\phi(x_i) \cdot f_\phi(x_j^{'})}{||f_\phi(x_i)||_2 ||f_\phi(x_j^{'})||_2} \tag{4}$$

Here T denotes the transport matrix that has the same meaning as γ defined in subsection "Optimal Transport (OT) Distance" and $c(x_i, x_j^{'})$ denotes the cost matrix which signifies how expensive it is to transport i^{th} sample of clean data x to the j^{th} sample of perturbed data $x^{'}$. This cost matrix is calculated using the cosine-similarity error between the clean classification probabilities u and the adversarial classification probabilities v. Therefore, such a training uses the inter-dependence between data-samples to improve the robustness of the classification model without requiring any ground truth label. In the next section we discuss how we incorporate OT-distance and feature-scattering to our proposed framework to produce better robust models.

IV. Methodology

A. Overview

Our proposed framework works on the principle of generative adversarial networks [23]. The gan framework [23] consists of two components, a generator, and a discriminator. The generator learns to generate a distribution such that the discriminator fails to discriminate between the generated distribution and the clean distribution, thus maximizing the discriminator loss. The discriminator on the other hand learns to distinguish between the generator generated distribution and the clean data distribution, thus minimizing the generator loss.

Similar to the GAN framework [6] proposed a multi-iterative attack that learns to maximize the classification loss for a given data through an adversarial perturbation. However, the author utilizes a hand-crafted algorithm for searching for an adversarial perturbation that requires tweaking multiple hyper-parameters to result in the most optimal results. Such a hand-crafted algorithm has several problems, where the first one being that the perturbations are optimized considering only the classification loss thus, uses no information about how the adversarial clean data is aligned in a latent space, and the second one is that the optimization procedure is biased towards the decision boundary to generate perturbations as noted by [9].

Though [11] have proposed to solve such problems. These approaches are expensive in terms of cost(iterations) since their generative adversary has to go through a large number of iterations to refine the perturbation to maximize classification loss. Moreover, similar to the work of [6], the adversarially trained models of [11] cannot generalize well to unseen adversarial data. To solve this problem of generalization and improve upon the robustness of the classification model, we

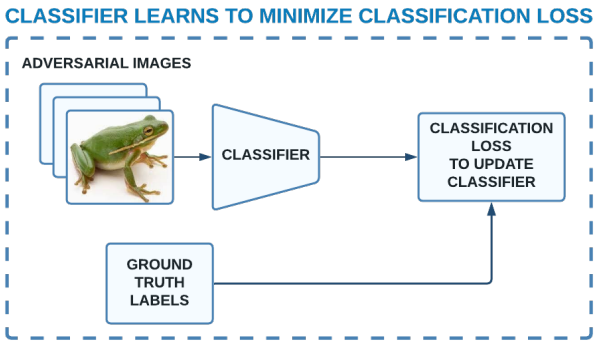

Fig. 2: Classification model Training

propose a novel GAN framework that can learn the orientation of clean and perturbed data through OT-loss. We use the proposed GAN framework to solve a min-max optimization problem, where the GAN framework generator learns to maximize the OT-loss rather than the cross-entropy loss.

For each generated perturbation, the generator uses the inter-sample relationship between data samples obtained through OT-Loss. The generator training setup is depicted in Figure 1. The main advantage of our generative framework over Feature-Scatter adversarial training [9] is that our method does not generate adversarial perturbations from scratch for each data batch as is done in the hand-crafted algorithm of [9]. Moreover, during the learning phase, our model also learns to generate diverse perturbations through mode-seeking loss usage [24]. Thus, producing more varied adversarial samples that when used in training a classifier allows the classification model to perform well on many more sample points available in an adversarial data space [8]. This usage of OT-loss allows our method to not have an excessive bias towards the decision boundary as seen by [9]. The classification model then learns to minimize the label smoothened cross-entropy loss regularised by the OT-loss over the generated adversarial data as depicted in Figure 2.

B. Algorithm

To train our proposed adversarial training framework we use Algorithm 1 which begins by first initializing the generator model $G_\theta(\cdot)$ with parameter θ, a classification model $F_\phi(\cdot)$ with parameter ϕ, the perturbation bound ϵ, the total number of attack iterations t (set to 1) and the total number of training epochs T, classifier learning rate α_1, generator learning rate α_2, L_{ms} coefficient λ, and label smoothing parameter s_p.

We then sample a randomly chosen data-batch $x_i, y_i{}_1^n \in S$ and initialize the perturbation δ with random uniform noise clipped between the perturbation bound ϵ. These perturbations are added to the clean samples x_i to initialize adversarial samples $x_i^{'}$. Then for each time-step till t, random latent vectors z is sampled from the normal distribution, and $D(\cdot,\cdot)$ (OT-Loss) is calculated between the output logit values of the clean and adversarial samples. Using latent vector z, adversarial samples $x_i^{'}$, perturbation δ, and OT-Loss gradients $\nabla_{x_i^{'}} L_{ot}$, the generator generates and updates perturbation δ

Algorithm 1: FS-GAN training

Initialise: Generator $G_\theta(\cdot)$, Classification model $F_\phi(\cdot)$, dataset S, training epochs T_e, batch size n, generator learning rate α_1, classifier learning rate α_2, perturbation bound ϵ, L_{ms} coefficient λ, label-smoothing parameter s_p, attack iteration t;

for *epoch = 1 to T* **do**

  for *random data batch* $\{x_i, y_i\}_1^n \in S$ **do**

    Sample: $\delta \in U(-\epsilon, +\epsilon)$;

    $x_i^{'} \leftarrow x_i + \delta$;

    for i *= 1 to* t **do**

      $z \leftarrow N(0,1)$;

      $L_{ot} \leftarrow D(F_\phi(x_i), F_\phi(x_i^{'}))$;

      $\delta \leftarrow G_\theta(z, \delta, x_i^{'}, \nabla_{x_i^{'}} L_{ot})$;

      $x_i^{'} \leftarrow x_i + \delta$;

      $L_{ot} \leftarrow D(F_\phi(x_i), F_\phi(x_i^{'}))$;

      $L = L_{ot} + \lambda \cdot L_{ms}$;

      $\theta \leftarrow \theta + \alpha_1 \cdot \frac{1}{n} \sum \{\nabla_\theta L\}$;

    $L_{cls} \leftarrow (L_{soft_{x}ent}(F_\phi(x_i^{'}), y_i, s_p))$ + L_{ot};

    $\phi \leftarrow \phi$ - $\alpha_2 \cdot \frac{1}{n} \sum \{\nabla_\phi L_{cls}\}$;

within the perturbation bound ϵ. At the same time we calculate the diversity loss L_{ms} (ms-loss) proposed by [24] as follows:

$$L_{ms} = \frac{d_\delta(\delta_1, \delta_2)}{d_z(z_1, z_2)} \tag{5}$$

Here two random latent vector z_1, z_2 (i.i.d samples of z) are used to generate perturbations δ_1, δ_2 using $G_\theta(\cdot)$ using same values of adversarial sample $x_i^{'}$, gradients $\nabla_{x_i^{'}} L_{ot}$ and perturbation δ. The equation 5 contains the mean absolute distance between the generated perturbations δ_1, δ_2 and the mean absolute distance between the random latent vectors z_1 and z_2. This loss calculates the distance between the generated perturbations w.r.t. the distance between the random latent vectors. As shown by [11] the usage of diversity helps the generator produce a diverse set of perturbations for each time-step t as depicted in Figure 3 containing a 2D t-SNE plot.

The updated perturbation δ is added to the previous adversarial samples $x_i^{'}$. After which the generator is updated with learning rate α_1 to maximize the OT-Loss/sinkhorn distance between the updated adversarial and clean samples and the ms-loss L_{ms}. This process is repeated for the attack iterations t. After the attack iterations t are completed the refined perturbation is added to the clean image x_i and passed onto the classification model $F_\phi(\cdot)$. The effectiveness of our approach is visualized in Figure 4, as seen our generative adversary produces more adversarial samples out of a fixed data-batch when compared to feature-scatter [9]. The classification model $F_\phi(\cdot)$ learns with learning rate α_2 to minimize the sum of soft-crossentropy loss and the adversarial OT-loss, where the soft-crossentropy loss is calculated between the output logits values

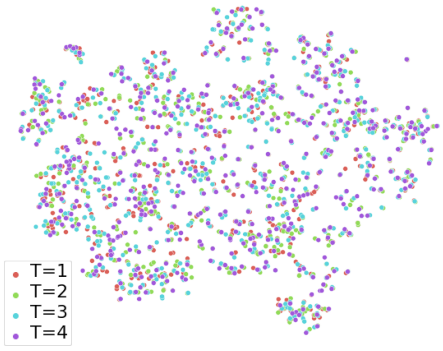

Fig. 3: A 2D t-SNE plot produced using the logit values obtained from a Wide-ResNet model when our generated adversarial examples are passed as input. This plot depicts the diversity in generated perturbation for different passes with fixed generator learnt weights and data-batch.

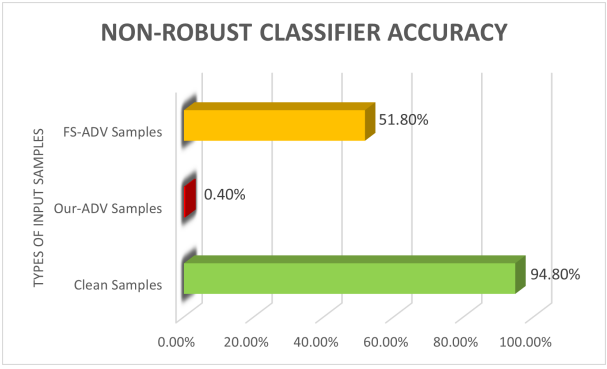

Fig. 4: Clean and adversarial samples accuracies for our and Feature-Scatter approach on a non-robust classifier for a fixed data batch.

$F_\phi(x_i + \delta)$ and the true class y_i using the label-smoothing parameter s_p.

V. Experimentation

A. Datasets

We test our methodology across real-world datasets consistent with adversarial attack and defense literature, like Cifar10 [25] containing (32 x 32) images of 10-classes with 50000 training and 10000 testing samples, Cifar100 [25] is another dataset similar to cifar10 containing 100 classes with 500 training samples per-class and 10000 testing samples, SVHN [26] a digit recognition datasets containing 10 classes with 73257 training and 26032 testing (32,32) images.

B. Training Setup

We train both our and the baseline (Baseline approaches were tested using the approach and the codes provided by their respective authors [9], [11], [12]) approaches from scratch. We use the UNet [27] architecture as our generator model and the Wide-ResNet (WRN-28-10) as the classifier and train it using each approach for 100 epochs for Cifar10, SVHN to facilitate better comparison against different approaches. For Cifar100 we train our and Feature-Scatter approach for 200 epochs, due to the lower number of samples per class. To train our generator we use a SGD optimizer with a learning rate initially set to 7.8e-3 for Cifar10, SVHN, and 7.8e-5 for Cifar100. This learning rate is reduced by a factor of 10 at the 60^{th}, and 90^{th} epoch. Similarly, the classifier is optimized using the SGD optimizer with a learning rate initialized with 0.1 for Cifar10, Cifar100 and 0.01 for SVHN. This learning rate is decreased by a factor of 10 at 60^{th} and 90^{th} epoch. We use the sinkhorn optimizer [28] for OT-loss with a regularisation of 0.01. The label smoothing parameter, the ms-loss coefficient are set to 0.5 and 0.05 respectively and the random latent vector z dimension is set to 10. Attack iterations t is set to 1 for Cifar10, SVHN and Cifar100. For performing computation we use a single NVIDIA Tesla V100 GPU.

C. Evaluation

In this subsection, we discuss the different attacks against which we test the baselines and proposed approach. We utilise these attacks since these attacks have been used by various works to benchmark their defence methodologies. These attacks are used to bring out vulnerability of a defence method in different threat models.

1) WhiteBox attacks:: Such attacks have complete access to the defence methodology. We measure the classification accuracies for each approach on l_∞ norm-based adversarial samples generated from attacks such as PGD [6], C&W [29], and FGSM [13]. For each attack $\epsilon = \frac{8}{255}$ and we report results for multiple iterations of 10, 20, 100 for PGD(step size = $\frac{2}{255}$, random restarts = 10) and C&W attack.

2) BlackBox attacks:: Such attacks have no information about classifier weights or the defence method used. We test against a (PGD, C&W, FGSM)-transfer attacks with the same hyper-parameter as described for white-box attacks using a copy of the victim model. We also test against the SPSA [30] attack with a batch size of 128 and the no. of attack iterations set to 1000 and a l_∞ norm of ϵ=$\frac{8}{255}$. SPSA attack becomes intuitive for testing model robustness since using OT-Loss based adversarial training, the model may learn to maximize the $cost^{iterations}$ of the finding a strong perturbation.

D. Results

We discuss the results of various white-box and black-box attacks performed on our methodology and compare them with our baselines Feature-Scatter(FS) [9], L2L-DA [11], L2L [12], PGD [6] with 10-update steps (PGD10). Here, L2L-DA and L2L are approaches that use a generator network for constructing adversarial perturbations. Different attacks and their hyper-parameters are explained above.

CIFAR10: On Cifar10 our proposed methodology achieves an increment in robust accuracy of 15.82% on 10-step PGD trained model, 5.68% on FS, 22.68% on multi-iterative generator based approach (L2L-DA), and 38.85% on single-step generator based approach (L2L) against a 100-step white box PGD attack. Similarly, it can be seen from Table II that our approach outperforms FS by 4.91%, generator-based approaches such as L2L-DA, L2L by 25.49%, 40.7% and

TABLE I: Accuracy comparison of the proposed approach with the given baselines on CIFAR10 dataset under a variety of black-box attacks.

Methodology	Clean Accuracy	PGD10	PGD20	PGD100	FGSM	SPSA
Ours (FS-GAN)	92.99%	**92.97%**	**93.08%**	**93.08%**	**93.13%**	**85.43%**
Feature-Scatter	95.04%	81.60%	81.00%	80.95%	82.35%	79.41%
Clean Model	94.59%	0.09%	0.08%	0.08%	24.34%	7.11%

TABLE II: Accuracy comparison of the proposed approach with the given baselines for CIFAR10 dataset under a variety of white-box attacks.

Methodology	Clean Accuracy	PGD10	PGD20	PGD100	FGSM	CW20	CW100
Ours (FS-GAN)	93.730%	**70.07%**	**66.44%**	**61.34%**	79.58%	**70.49%**	**64.97%**
Feature-Scatter	**94.90%**	69.61%	63.52%	55.66%	**84.46%**	62.40%	60.06%
L2L-DA	80.43%	40.2%	39.13%	38.66%	46.42%	56.86%	39.48%
L2L	86.43%	30.27%	25.74%	22.49%	63.54%	49.90%	24.27%
PGD10	84.82%	47.37%	45.98%	45.52%	55.51%	45.70%	47.42%

TABLE III: Accuracy comparison of the proposed approach with the given baselines for SVHN dataset under a variety of white-box attacks

Methodology	Clean Accuracy	PGD10	PGD20	PGD100	FGSM	CW20	CW100
Ours (FS-GAN)	96.37%	**70.69%**	**64.98%**	**56.78%**	82.26%	**66.94%**	**55.30%**
Feature-Scatter	**96.46%**	70.01%	61.76%	51.27%	**85.79%**	61.4%	50.38%
L2L-DA	91.50%	50.54%	48.60%	47.86%	60.73%	66.06%	44.81%
L2L	94.07%	27.92%	19.17%	12.03%	83.30%	59.63%	18.11%
PGD10	92.78%	48.77%	47.91%	46.89%	68.27%	48.70%	47.28%

TABLE IV: Accuracy comparison of the proposed approach with the given baselines for CIFAR100 dataset under a variety of white-box attacks

Methodology	Clean Accuracy	PGD10	PGD20	PGD100	FGSM	CW20	CW100
Ours (FS-GAN)	74.28%	**45.65%**	**45.05%**	**44.38%**	51.09%	**39.01%**	**29.43%**
Feature-Scatter	**75.66%**	44.25%	43.20%	41.32%	**58.00%**	34.00%	27.05%
L2L-DA	53.89%	20.20%	19.56%	19.28%	23.14%	29.48%	19.13%
L2L	61.63%	19.60%	17.74%	16.44%	35.99%	32.65%	17.92%
PGD10	60.40%	24.82%	23.88%	23.53%	29.14%	23.20%	23.00%

10-step PGD trained model by 17.75% respectively against 100-step white-box C&W attack. Also as it can be seen from Table I that our approach is completely robust against a wide variety of strong black-box attacks and improves upon FS by a large margin on the robust accuracy. These results also validate that our approach produces a wide variety of diverse perturbations that prevents any black-box attack from affecting model performance.

SVHN: Following the trend seen on Cifar10, our proposed methodology on SVHN achieves an increment in robust accuracy of 9.89% on 10-step PGD trained model, 5.51% on FS, 8.92% on multi-iterative generator based approach (L2L-DA), and 44.75% on single-step generator based approach (L2L) against a 100-step white box PGD attack. According to Table III our approach outperforms FS by 4.92%, generator-based approaches such as L2L-DA, L2L by 10.49%, 37.19% and 10-step PGD trained model by 8.02% respectively against 100-step white-box C&W attack.

CIFAR100: Defending models trained on such datasets is much more challenging owing to a low number of just 500 samples per class available for training. Previously none of the generator-based approaches such as L2L-DA and L2L were tested on such datasets, so to facilitate better comparison we train these approaches on Cifar100 and report the results in Table IV. Our proposed approach is able to improve upon by 3.06% on FS, 25.1% on multi-iterative generator based approach (L2L-DA), 27.94% on single-step generator based approach (L2L), and 20.85% on a 10-step PGD trained model against a 100-step PGD. A similar trend can be seen for a 100-step C&W attack where a method surpasses FS by 2.38%, L2L-DA by 10.3%, L2L by 11.51% and 10-step PGD trained model by 6.43%. These results validate that our approach is generalizable across multiple dataset delivering similar performance.

VI. Conclusion

We present a novel defense framework for learning a robust classification model with good robust generalization capability even when there is a lack of a large number of labeled data samples when compared to past generator-based adversarial defence frameworks. Our framework utilizes a convolutional neural network-based generator that effectively learns the inter-sample relationship between data samples producing a diverse set of perturbations. Overall, the proposed approach fuses a generative adversary with the optimal transport distance measure that uses diversity-based regularisation to efficiently explore the adversarial data manifold. Through extensive experiments performed in this study, we demonstrate that the classifier trained using our approach has better performance against strong adversarial attacks. We highlight some drawbacks of previous works using a generator modeled as an adversary and the advantage our framework brings to the community.

REFERENCES

[1] 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.

[2] S. IlyasA *et al.*, "Adversarialexamples arenotbugs, theyarefeatures," *Procofthe33rdAnnual Confon NeuralInformationProcessingSystems (NeurIPS). Cambridge, MA: MITPress*, vol. 125136, 2019.

[3] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-gan: Protecting classifiers against adversarial attacks using generative models," *arXiv preprint arXiv:1805.06605*, 2018.

[4] J. Buckman, A. Roy, C. Raffel, and I. Goodfellow, "Thermometer encoding: One hot way to resist adversarial examples," in *International Conference on Learning Representations*, 2018.

[5] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "Pixeldefend: Leveraging generative models to understand and defend against adversarial examples," *arXiv preprint arXiv:1710.10766*, 2017.

[6] 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.

[7] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International Conference on Machine Learning*. PMLR, 2018, pp. 274–283.

[8] L. Chen, Y. Min, M. Zhang, and A. Karbasi, "More data can expand the generalization gap between adversarially robust and standard models," in *International Conference on Machine Learning*. PMLR, 2020, pp. 1670–1680.

[9] H. Zhang and J. Wang, "Defense against adversarial attacks using feature scattering-based adversarial training," *Advances in Neural Information Processing Systems*, vol. 32, pp. 1831–1841, 2019.

[10] K. R. Mopuri, U. Ojha, U. Garg, and R. V. Babu, "Nag: Network for adversary generation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 742–751.

[11] Y. Jang, T. Zhao, S. Hong, and H. Lee, "Adversarial defense via learning to generate diverse attacks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2740–2749.

[12] H. Jiang, Z. Chen, Y. Shi, B. Dai, and T. Zhao, "Learning to defend by learning to attack," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2021, pp. 577–585.

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[14] L. Rice, E. Wong, and Z. Kolter, "Overfitting in adversarially robust deep learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 8093–8104.

[15] A. Shafahi, M. Najibi, A. Ghiasi, Z. Xu, J. Dickerson, C. Studer, L. S. Davis, G. Taylor, and T. Goldstein, "Adversarial training for free!" *arXiv preprint arXiv:1904.12843*, 2019.

[16] A. Shafahi, M. Najibi, Z. Xu, J. Dickerson, L. S. Davis, and T. Goldstein, "Universal adversarial training," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 5636–5643.

[17] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, and A. Madry, "Do adversarially robust imagenet models transfer better?" *arXiv preprint arXiv:2007.08489*, 2020.

[18] E. Wong, L. Rice, and J. Z. Kolter, "Fast is better than free: Revisiting adversarial training," *arXiv preprint arXiv:2001.03994*, 2020.

[19] A. Matyasko and L.-P. Chau, "Improved network robustness with adversary critic," *arXiv preprint arXiv:1810.12576*, 2018.

[20] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," *arXiv preprint arXiv:1801.02610*, 2018.

[21] H. Wang and C.-N. Yu, "A direct approach to robust deep learning using adversarial networks," *arXiv preprint arXiv:1905.09591*, 2019.

[22] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*. PMLR, 2017, pp. 214–223.

[23] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[24] Q. Mao, H.-Y. Lee, H.-Y. Tseng, S. Ma, and M.-H. Yang, "Mode seeking generative adversarial networks for diverse image synthesis," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1429–1437.

[25] A. Krizhevsky, G. Hinton *et al.*, "Learning multiple layers of features from tiny images," 2009.

[26] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," 2011.

[27] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[28] M. Cuturi, "Sinkhorn distances: Lightspeed computation of optimal transport," *Advances in neural information processing systems*, vol. 26, pp. 2292–2300, 2013.

[29] 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.

[30] J. Uesato, B. O'donoghue, P. Kohli, and A. Oord, "Adversarial risk and the dangers of evaluating against weak attacks," in *International Conference on Machine Learning*. PMLR, 2018, pp. 5025–5034.