IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 30, 2021, accepted June 29, 2021, date of publication July 30, 2021, date of current version August 10, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3101282

Adversarial Training Time Attack Against Discriminative and Generative Convolutional Models

SUBHAJIT CHAUDHURY, (Member, IEEE), HIYA ROY, (Member, IEEE), SOURAV MISHRA, (Student Member, IEEE), AND TOSHIHIKO YAMASAKI, (Member, IEEE)

Department of Information and Communication Engineering, The University of Tokyo, Bunkyo-ku, Tokyo 113-8656, Japan

Corresponding author: Subhajit Chaudhury (subhajit@hal.t.u-tokyo.ac.jp)

This work was supported by the Japan Society for the Promotion of Science (JSPS) through the Grants-in-Aid for Scientific Research (KAKENHI) under Grant 19K22863.

ABSTRACT In this paper, we show that adversarial training time attacks by a few pixel modifications can cause undesirable overfitting in neural networks for both discriminative and generative models. We propose an evolutionary algorithm to search for an optimal pixel attack using a novel cost function inspired by domain adaptation literature to design our training time attack. The proposed cost function explicitly maximizes the generalization gap and domain divergence between clean and corrupted images. Empirical evaluations demonstrate that our adversarial training attack can achieve significantly low testing accuracy (with high training accuracy) on multiple datasets by just perturbing a single pixel in the training images. Even under the use of popular regularization techniques, we identify a significant performance drop compared to clean data training. Our attack is more successful than previous pixel-based training time attacks on state-of-the-art Convolutional Neural Networks (CNNs) architectures, as evidenced by significantly lower testing accuracy. Interestingly, we find that the choice of optimization plays an essential role in robustness against our attack. We empirically observe that Stochastic Gradient Descent (SGD) is resilient to the proposed adversarial training attack, different from adaptive optimization techniques such as the popular Adam optimizer. We identify that such vulnerabilities are caused due to over-reliance on the cross-entropy (CE) loss on highly predictive features. Therefore, we propose a robust loss function that maximizes the mutual information between latent features and input images, in addition to optimizing the CE loss. Finally, we show that the discriminator in Generative Adversarial Networks (GANs) can also be attacked by our proposed training time attack resulting in poor generative performance. Our paper is one of the first works to design attacks for generative models.

INDEX TERMS Generalization in deep learning, data poisoning, adaptive optimization, training time attack, variational information bottleneck.

I. INTRODUCTION

Convolutional Neural Networks (CNNs) [1]–[3] are a powerful class of models that learn hierarchical feature representations and have shown notable empirical success in various application areas. Typically, in an over-parametrized setting with a highly non-convex loss surface, classical learning theory [4] predicts that these deep neural network models should have a high out-of-sample error because the solution is likely to get stuck at a local minimum. Nonetheless, deep neural networks appear to generalize well, even in small data regimes. Numerous previous works have shown that current deep learning models are not robust against adversarial attacks [5]–[8]. However, due to these models' notable empirical success in various application areas such as computer vision [1], [9], natural language processing [10]–[12], and other real-world domains, deep learning is poised to lead us to the next industrial revolution. The increasing use of such machine learning models in security-critical applications and the unpredictable behavior of these models under tiny well-crafted perturbations demands a better understanding of neural network robustness to ensure safe and practical implementations.

Traditional methods in adversarial attacks [5]–[7], [15]–[21] fool trained neural networks using *adversarial* query images. These attacks add small perturbations to

The associate editor coordinating the review of this manuscript and approving it for publication was Kok-Lim Alvin Yau.

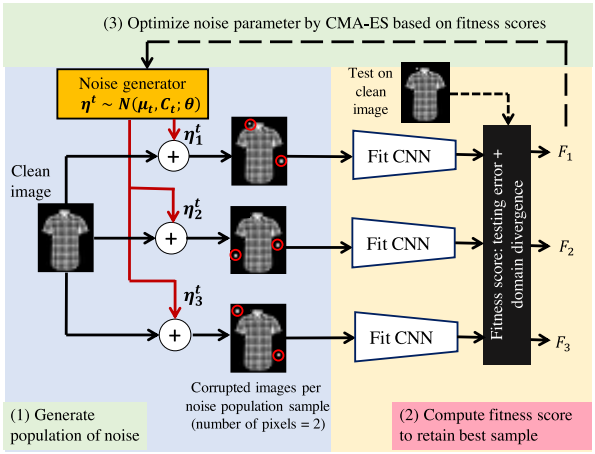


FIGURE 1. Overview of our proposed evolutionary algorithm for training time attack optimization. We sample from a noise generator ($N_P = 2$ case shown) to perturb a few pixels in training images and fit CNN models on noisy data. Evaluation is performed on clean data from the training set to find fitness scores (high generalization loss). Noise generator parameters are updated by evolutionary strategy [13].

the query images resulting in the classification function crossing the true class's decision boundary causing incorrect classification. However, such perturbations are imperceptible by humans, making them difficult to detect. Such perturbation-based methods fall under the category of evasion attacks that fools the model during inference.

We propose a novel evolutionary strategy-based algorithm, called *EvoShift*, for optimizing pixel attacks that are added to the training images. Figure 1 shows an overview of our method. The difference between our method and evasion attacks is shown in Figure 2. Our contributions in this paper can be summarized as

- To obtain our training time attack, we solve a joint min max optimization with the outer maximization designed to find the pixel noise and inner minimization designed to train the neural network on the noisy images. We impose a constraint on the optimization such that the CE loss on the noisy images is low, and the loss on the clean images is high. Figure 1 shows the various components of our proposed algorithm for finding optimal training pixel attacks. Such a formulation results in CNN trained on the noisy images having a very high error on clean test images, thus exposing serious vulnerabilities in CNNs that are detrimental to robust learning.
- Interestingly, we find that optimization choice plays a vital role in generalization robustness. We show empirical evidence that SGD is resilient to our training time attacks, different from adaptive optimization techniques (Adam). Although adaptive optimization methods are a popular choice for practitioners and researchers, we show that they can easily overfit in the presence of our training time attacks. We believe that this is an important finding for the machine learning research community.
- We also apply regularization methods to counteract our proposed adversarial training time attack and find that well-known regularization methods such as dropout and weight decay are ineffective against our attack. We find that random-crop data augmentation is moderately effective for a few pixel attacks.
- As a defense against our attack, we propose a robust loss function for CNN classification training that is resilient against our training attacks using an information maximization framework. This result suggests that the traditional cross-entropy minimization framework for CNN training might cause non-robust feature learning, which might be mitigated by our proposed information-theoretic loss function.
- We introduce the concept of vulnerability in GANs under the proposed *EvoShift* attacks, causing poor image generation quality due to overfitting in the GAN discriminator. Figure 3 shows that GANs trained under our proposed attack fail to obtain a detailed reconstruction of the object in the image, thus exposing weakness in image generation using GANs. This paper is the first work showing vulnerabilities in GANs under training time attacks to the best of our knowledge.

Our method has some similarities to data poisoning attacks [22]–[24], where the adversary injects a few malicious samples in the training data to cause incorrect classification (typically targeted) during inference. However, our method is technically different from poisoning attacks. Poisoning methods only target a few images in the test set to attack by changing the decision boundary in a limited local region. In contrast, our method's main objective is to induce overfitting in neural networks. The proposed attack tries to change training images so that the induced decision function shows a significant departure from the true decision boundary. Our attack finds the best location of pixel disturbance for each class in a multi-label dataset that maximally increases overfitting. Using our method, we expose serious vulnerabilities in neural networks that can overfit even single-pixel disturbance, which is an undesirable feature for robust machine learning.

This work is an improved and extended version of our previous work [25], which finds few-pixel training perturbations designed to analyze the generalization of CNNs under adversarial training perturbations. Additionally, in this paper, we proposed the concept of adversarial training time attack for GANs (Section V-D) and introduced a robust loss function for CNN learning (Section VII). We also performed additional experiments for partial dataset attack (Section VIII-C), training accuracy in the presence of regularization (Section VIII-C), and transfer of our attack to ImageNet dataset using the Spatial Value Function (SVF) method (Section V-E).

II. RELATED WORKS

A. ADVERSARIAL ATTACKS

This recent line of work [5]–[8] demonstrated that it is possible to fool trained neural networks using *adversarial* query images that are imperceptible from normal unperturbed

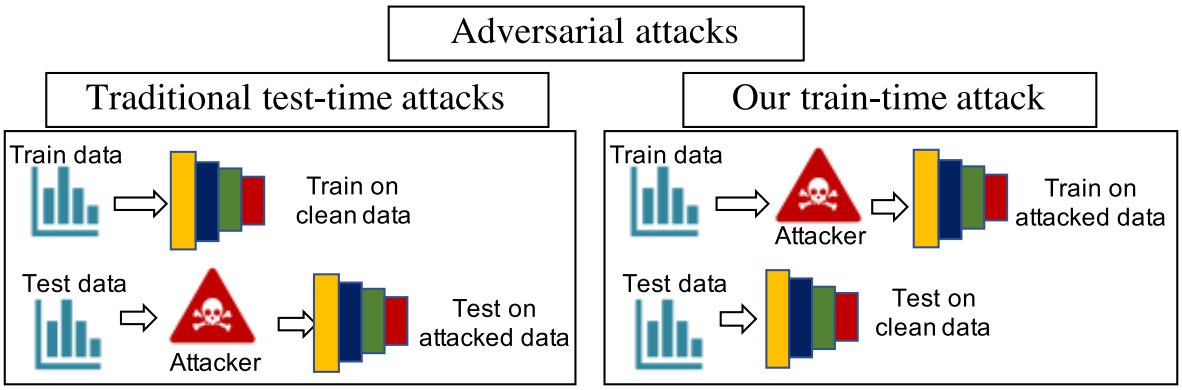
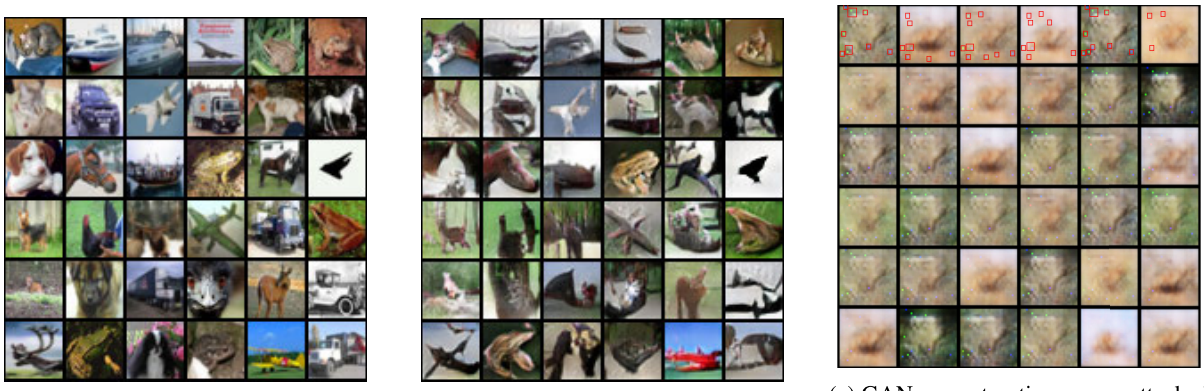


FIGURE 2. Difference of our proposed method from traditional adversarial attacks. Common adversarial attacks [5]–[7] are test-time attacks where the model is trained on clean images and tested on attacked images. Our setting is where the model training is performed on the attacked training images, and testing is performed on clean samples. We use an evolutionary algorithm for finding training time attacks that maximally show low testing error on clean test samples.



(a) Samples from true image distribution. (b) GAN reconstruction on true images. (c) GAN reconstruction on our attacked data.

FIGURE 3. Our proposed attack method in GANs: (a) true images on CIFAR10 dataset, (b) images generated by GAN [14] trained on the true data distribution, (c) images generated by training on our proposed attacked data by just changing only 10-pixel values. Generated images faithfully recreate the pixel nuisance features (highlighted in the first row with red markers) while ignoring the semantic features such as object shape and color.

images. Su *et al.* [26] showed that it is possible to craft adversarial test images by single-pixel perturbations in training images. These attacks fall under the category of *evasive* attacks that exploit the weakness in trained models by attacking query images. Instead of attacking query data during inference, our method corrupts the training data to maximize the generalization error.

B. DATA POISONING

In data poisoning, the attacker injects malicious samples in the training data to control the model behavior during test time. Such an attack was first introduced in Support Vector Machines (SVM) for binary classification problems in [22]. Recently, there have been some works in the field of neural networks [24] as well. Koh and Liang [27] used influence functions to synthesize adversarial training examples that can flip the predicted labels of a set of testing images. Shafahi *et al.* [23] used a forward-backward-splitting iterative procedure [28] to create targeted data poisoning attacks that performed better than previous methods. As we mentioned earlier, different from previous works, our method presents a general gradient-free strategy for crafting adversarial training perturbations, which is agnostic to the underlying learning algorithm, with precise control on noise parameters. Jacobsen *et al.* [29] studied the effect of single-pixel perturbations on MNIST training images on test performance. They showed that adding one pixel to training images that encodes the class label, and then testing on the clean test set, can yield a high generalization gap. Tanay *et al.* [30] showed that neural network models could be made almost arbitrarily sensitive to a single pixel while maintaining identical test performance between models. Different from poisoning methods, our method's main objective is to induce overfitting in neural networks using the proposed gradient-free optimization.

C. NEURAL NETWORK GENERALIZATION

Numerous previous works [31]–[34] studied the generalization properties of neural networks under such a high complexity parameter space. Zhang *et al.* [35] showed that neural networks could fit random noise. The idea of pixel perturbation has also been explored in [36] to measure the testing accuracy of images. Different from previous works,

our method analyzes the robustness of neural networks under optimally crafted perturbations in training images, similar to Wilson *et al.* [37], which presented a manually crafted artificial example. However, our method is a generalization to such problems that can generate optimal training perturbations for an arbitrarily sized dataset using evolutionary algorithms.

D. INFORMATION THEORY-BASED METHODS

We also review some generative learning methods for adversarial defense. Alemi *et al.* [38] showed that learning with Variational Information Bottleneck (VIB) is robust to standard perturbation-based adversarial example. Song *et al.* [39] proposed a generative model called PixelDefend to detect adversarial samples and moving them back to the training data distribution. Meng and Chen [40] used autoencoders to detect adversarial inputs by using the reconstruction threshold and proposed a mechanism to defend against a gray box attack. With the recent interest in the information-theoretic view because of the information bottleneck [41], [42], the estimation of mutual information [43], [44] has attracted a lot of attention.

E. ADVERSARIAL DEFENSE METHODS

Adversarial defense methods such as defensive distillation [45], feature squeezing [46], MagNet [40], and certified adversarial robustness [47] are designed for defense against test-time adversarial attacks. Since our proposed method is tackling adversarial training time attacks, such defense methods are not applicable for test-time defense. Furthermore, generative model-based defensive methods also exist in the literature. [48] uses GANs for generating adversarial samples and train the classifier with such samples for adversarial robustness. Similarly, [39], [49] use generative models to clean the attacked images by projecting the attacked image on the learned manifold of the generative models. Our contribution is not related to adversarial defense. We show fundamental weakness in GANs. More specifically, we show that GANs trained on our proposed adversarial training attacks fail to generate semantic features from images and only generate attack pixel features that are strong features for the GAN discriminator. This is a novel kind of attack against GANs which has not been explored in previous works.

III. PROBLEM FORMULATION

We consider a multi-class classification task with the input space $X \in \mathbb{R}^N$ and the label space $Y = \{1, \ldots, N_c\}$. The true data distribution is given as, $S = \{\boldsymbol{x}_i, y_i\}_{i=1}^{n} \sim \mathcal{D}_S$, where n is the total number of images, $\boldsymbol{x}_i$ is an instance of the image from the dataset, and y_i is the corresponding image label. Our goal is to design a pixel-perturbation attack such that the classifier trained on the perturbed training data yields a high empirical risk (or test error) on the true uncorrupted samples. Only the training set is used for obtaining such pixelwise attack. Considering that for each sample in S, we can draw class-wise input perturbations, $\boldsymbol{\Delta} = \{\boldsymbol{\delta}_i\}_{i=1}^{N_c} \sim N(\boldsymbol{\mu}, \Sigma)$, parameterized by the mean $\boldsymbol{\mu}$ and the covariance matrix Σ. N_c is the total number of classes. The class-wise noise is added to training images as $\boldsymbol{x}_i^p = \boldsymbol{x}_i + \boldsymbol{\delta}_{y_i}$. The joint distribution of the perturbed data, which is constructed by assigning labels of the true samples to the corresponding perturbed samples, is given as $P = \{\boldsymbol{x}_i^p, y_i^p\}_{i=1}^{n} \sim \mathcal{D}_{adv}$.

Let us define a classifier function $h : X \rightarrow Y$ from a hypothesis space $\mathcal{H}$. The corresponding empirical risk on samples drawn from a distribution $\mathcal{D}$ is defined as, $R_{\mathcal{D}}(h) \stackrel{\text{def}}{=} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\big(I[h(\boldsymbol{x}) \neq y]\big)$, which signifies the error on the samples drawn from $\mathcal{D}$. Our objective is to find the optimal perturbation parameter that increases the empirical risk on the clean samples while minimizing it on the corrupted samples, thus compromising generalization in neural networks. This is given as

$$\begin{aligned} \boldsymbol{\theta}^* = \max_{\boldsymbol{\theta}} \Big(R_{\mathcal{D}_S}(h^*) - R_{\mathcal{D}_{adv}}(h^*) \Big) \\ s.t.\ h^* = \arg\min_{h\in\mathcal{H}} R_{\mathcal{D}_{adv}}(h). \end{aligned} \tag{1}$$

The above objective finds an optimal perturbation parameter that increases the empirical risk on the clean samples while minimizing it on the corrupted samples, thus compromising generalization in neural networks.

IV. THEORY ON MAXIMUM DOMAIN DIVERGENCE BASED PERTURBATION OPTIMIZATION

This section outlines the theory behind the domain divergence-based perturbation optimization, which lays the foundation for our evolutionary strategy-based perturbation optimization.

A. DOMAIN DIVERGENCE

Given a source domain ($\mathcal{D}_S$) and a target domain ($\mathcal{D}_T$), the notion of domain divergence refers to how samples in each of the domains differ from the other. For the conventional risk minimization regime, the ''domain gap'' can be measured by the difference between empirical risk in the source and target domains. Ben-David *et al.* [50], [51] formally defined this notion as the proxy $\mathcal{A}$-distance, which was used by domain adaptation methods [52], [53] for reducing the source and target domain errors in an adversarial setting.

Let us consider a domain $\mathcal{X}$ and a collection of subsets of $\mathcal{X}$, given as $\mathcal{A}$. Given two domain distributions $\mathcal{D}_S$ and $\mathcal{D}_T$ over $\mathcal{X}$, and a hypothesis class $\mathcal{H}$, the $\mathcal{A}$-divergence between the domains is given as

$$d_{\mathcal{A}}(S, T) \stackrel{\text{def}}{=} 2 \sup_{A\in\mathcal{A}} \Big| \Pr_{\mathcal{D}_S}[A] - \Pr_{\mathcal{D}_T}[A] \Big|, \tag{2}$$

where the hypothesis class $\mathcal{H}$ is a class of functions representing binary classifiers and is symmetric as defined in [50]. The above distance is the $\mathcal{H}$-divergence $d_{\mathcal{H}}(.,.)$ when we compute the distance of the class of subsets with characteristics functions in the hypothesis space $\mathcal{H}$.

Ben-David *et al.* [50], Ganin *et al.* [52] showed that although it is generally difficult to compute the $\mathcal{H}$-divergence

for the hypothesis space of linear classifiers, it can be approximately computed using the empirical $\mathcal{H}$-divergence from samples $\boldsymbol{x}_i^s \sim \tilde{D}_S$ and $\boldsymbol{x}_i^t \sim \tilde{D}_T$, and is defined as

$$\hat{d}_{\mathcal{H}}(S,T) \stackrel{\text{def}}{=} 2\Bigg(1 - \min_{h\in\mathcal{H}}\Bigg[\frac{1}{n}\sum_{i=1}^{n} I[h(\boldsymbol{x}_i^s)=0] + \frac{1}{n'}\sum_{i=n+1}^{N} I[h(\boldsymbol{x}_i^t)=1]\Bigg]\Bigg), \quad (3)$$

where n samples from the source domain and n' samples from the target domain are drawn. Given samples from the two domains, the above empirical distance can be computed as the proxy $\mathcal{A}$-distance by learning a classifier $h \in \mathcal{H}$, which optimally learns to discriminate between the source and target samples. The proxy $\mathcal{A}$-distance is defined as, $\hat{d}_{\mathcal{A}} = 2(1-2\epsilon)$ according to [50], where ϵ is the discriminator error.

B. BOUND ON TARGET RISK

We are interested in finding a bound of the empirical target risk obtained by learning a source samples classifier. Shai Ben *et al.* (and later used by Ganin *et al.* [50]–[52]) showed that the bound on the target risk could be computed in terms of the proxy $\mathcal{A}$-distance defined above, as follows.

Theorem 1: Considering $\mathcal{H}$ be a hypothesis class of Vapnik–Chervonenkis (VC) dimension d, for n samples $S \sim (\tilde{D}_S)^n$ and $T\widetilde{(\hat{D}_T)}^n$, then with the probability $1-\delta$ over the choice of samples, for every $h \in \mathcal{H}$:

$$\hat{R}_T(h) \leq \hat{R}_S(h) + \sqrt{\frac{4}{n}\left(d\log\frac{2en}{d} + \log\frac{4}{\delta}\right)} + \hat{d}_{\mathcal{H}}(S,T) + 4\sqrt{\frac{1}{n}\left(d\log\frac{2n}{d} + \log\frac{4}{\delta}\right)} + \beta, \quad (4)$$

with $\beta \geq \inf_{h^*\in\mathcal{H}}\left[R_S(h^*) + R_T(h^*)\right]$ and $\hat{R}_S(h) = \frac{1}{n}\sum_{i=1}^{m} I\left[h(\boldsymbol{x}_i^s) \neq y_i^s\right]$.

Given a fixed hypothesis space, we observe that increasing the $\mathcal{H}$-divergence between the two domains would make the above bound loose. It is to be noted that high domain divergence increases the range of values for the target risk, increasing the likelihood of overfitting.

The above analysis is relevant in our setup since we are interested in finding perturbations that, although constrained to a few pixel changes, increase the generalization gap (analogous to $\mathcal{H}$-divergence) between clean and perturbed training images. Although the above analysis is shown for a binary classification system, it is relevant to multi-class classification systems. We use this insight in our formulation to craft a fitness score to increase the domain divergence between the true and perturbed distributions.

V. OUR PROPOSED PIXEL-BASED PERTURBATION

Based on the domain divergence theory, we outline our proposed noise optimization strategy in this section. First, we explain how we parametrize the noise, then we describe our cost function, and finally, we present our proposed algorithm for the optimal perturbation generation.

A. PARAMETERIZING PIXEL ATTACK

We design our adversarial training time attack in the form of few-pixel perturbations for each class. Let us assume there is a total of N_c classes in the dataset. We perturb images of the same class label, with N_p pixel perturbations, which is represented as $\boldsymbol{\Delta} = \{(x_0^0, y_0^0, v_0^0), \ldots (x_{N_p}^0, y_{N_p}^0, v_{N_p}^0), \ldots (x_0^{N_c}, y_0^{N_c}, v_0^{N_c}), \ldots (x_{N_p}^{N_c}, y_{N_p}^{N_c}, v_{N_p}^{N_c})\}$, consisting of $(N_p \times N_c)$ pairs of pixel noise (x, y, v) where (x, y) represents the spatial location of the pixel noise and v represents the intensity of pixel disturbance.

Given a noise sample $\boldsymbol{\Delta}$, all images I_j in class k will have their pixel value represented by (x_i^k, y_i^k) assigned the intensity value v_i^k, such that, $I_j[x_i^k, y_i^k] = v_i^k$, for i=1 to N_p. These pixel perturbations act as the distractor features in our training images. The motivation behind the class-wise pixel attack encoding is to force the neural network to use the noisy pixel as the discriminative feature for that class while ignoring the semantic features such as image appearance and color information. Our goal is to find the distribution of spatial locations where such pixel distractions are the most effective for overfitting the CE loss.

During optimization, we draw N_p pixel perturbations for each class, from a multi-variate normal distribution, $\boldsymbol{\Delta} = \{\boldsymbol{\delta}_i\}_{i=1}^{N_C} \sim N(\boldsymbol{\mu}, \Sigma)$, parameterized by $(\boldsymbol{\mu}, \Sigma)$ which are the mean vector and the covariance matrix respectively. $\boldsymbol{\Delta}$ represents the $3 \times N_C \times N_p$ dimensional parameterization vector comprising of all class-wise perturbations. For color images, $\boldsymbol{\Delta}$ has $5 \times N_C \times N_p$ dimensions because the pixel value $v = (v_R, v_G, v_B)$ consists of three values for each channel. The optimization of these pixel noise is explained in the following section.

B. COST FUNCTION

Our neural network model F_θ is trained on the adversarially attacked training data by optimizing the CE loss, using the traditional training objective, $\mathbb{E}_{(x,y)\sim\mathcal{D}_{adv}}\left[\mathcal{L}_{CE}(F_\theta(x), y)\right]$. We propose a perturbation objective that will be successful if the model F_θ has a low CE loss on the perturbed training images and a high loss on the clean images. We combine the above condition in the form of a single equation, given as

$$\max_{\boldsymbol{\Delta}, s=0} \min_{\theta, s=1} \mathbb{E}_{(x,y)\ \mathcal{D}}\left[\mathcal{L}_{CE}(x + s\boldsymbol{\Delta}, y; \theta)\right]. \quad (5)$$

In the above equation, s acts as a switch to turn on/off the training data's perturbation. The minimization concerning the neural network parameters θ is performed in the presence of the perturbation, such that it overfits the noise. The maximization is performed with $s = 0$ such that the CE loss on clean samples should be high, thus creating an adversarial attack that maximizes the generalization gap encouraging overfitting.

According to Equation 5, it is difficult to optimize the noise parameter $\boldsymbol{\Delta}$ using standard gradient-based methods, because the gradient concerning $\boldsymbol{\Delta}$ is 0, due to multiplication with s. Therefore, we resort to gradient-free Covariance Matrix Adaptation Evolutionary Strategy (CMA-ES)

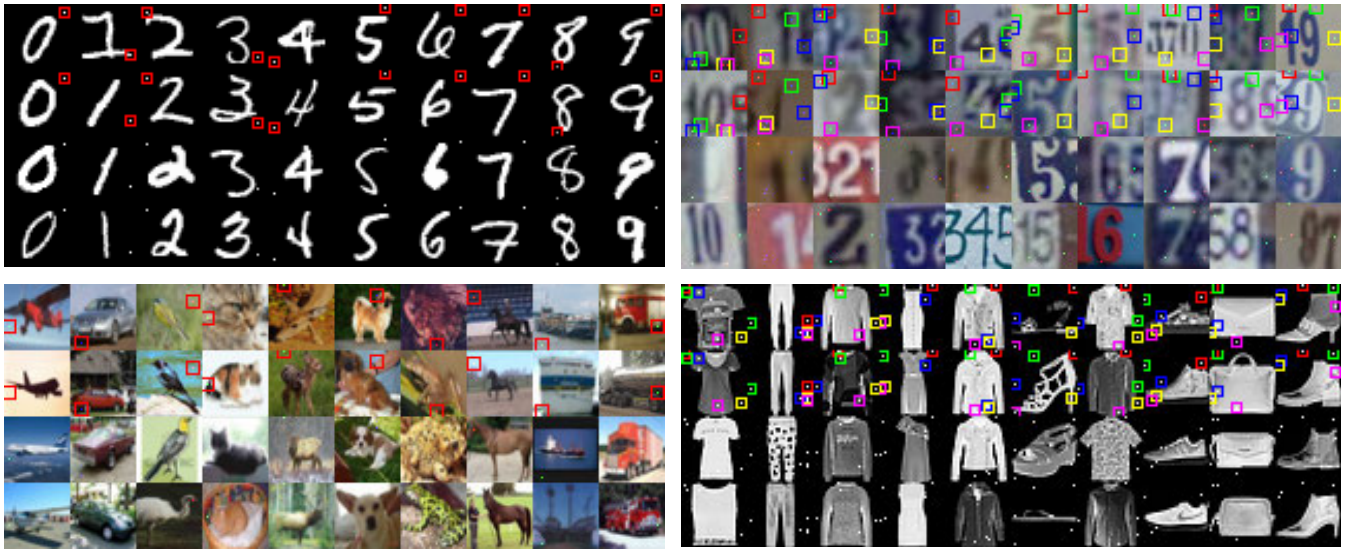


FIGURE 4. Perturbed image sample for single-pixel attack by our proposed EvoShift algorithm, showing class-wise pixel perturbations for (a) MNIST (top-left), (b) SVHN (Top-right), (c) CIFAR10 (Bottom-left), and (d) Fashion-MNIST (Bottom right). The top two image rows in each dataset samples are highlighted to help the reader spot the classwise pixel perturbations.

based optimization for finding the optimal perturbations. We describe the cost function for our CMA-ES optimization consisting of the following components as follows.

Semantic mismatch cost: Using the above argument, our fitness score should encourage a high cross-entropy loss on the images from the true data distribution while showing a low loss on the adversarially attacked training samples. The above condition emulates overfitting in the model. This is formulated as the difference of loss terms between these scenarios which we designate as the semantic mismatch cost (S_m) as follows:

$$S_m = \frac{1}{N} \sum_{(x,y)\sim\mathcal{D}} \Big[\mathcal{L}_{CE}(x, y; \theta) - \mathcal{L}_{CE}(x + \mathbf{\Delta}, y; \theta) \Big], \quad (6)$$

where the above score is maximized by the CMA-ES for a fixed θ trained on $(x + \mathbf{\Delta}, y)$ samples. The first term encourages a high loss on samples drawn from the true distribution, while the second term promotes a low loss on the perturbed image. This score measures the generalization gap between the samples drawn from true distribution and perturbed distribution which differ by only a few pixels.

Domain divergence: Given a source domain $\mathcal{D}_S$ and a target domain $\mathcal{D}_T$, the notion of domain divergence refers to how samples in each of the domains differ from the other, as explained in Section IV. In our settings, we want to train the model to have a low empirical risk on samples from $\mathcal{D}_{adv}$ and a high risk on the true distribution $\mathcal{D}$. This can be viewed as increasing domain divergence between these distributions. Ben-David *et al.* [50] and Ganin *et al.* [52] showed that approximate domain divergence could be computed by learning a binary classifier $h \in \mathcal{H}$, which optimally learns to discriminate between the source and target samples. In our case, we train a discriminator between uniformly sampled images from the true (label 1) and attacked (label 0) distributions. The domain divergence score S_d is computed as

$$S_d = 2\Big(1 - \big[\mathbb{E}_{x\in\mathcal{D}}\mathbb{I}[h(\boldsymbol{x})=0] + \mathbb{E}_{x\in\mathcal{D}_{adv}}\mathbb{I}[h(\boldsymbol{x})=1]\big]\Big), \quad (7)$$

where $\mathbb{I}(.)$ is the indicator function used for computing the error in discriminator. Intuitively, for attacked training samples in a population (for CMA-ES) that are dissimilar from the true samples, the discriminator can learn good separable features, thus having high domain divergence. We empirically find that by adding the domain divergence score, the stability of convergence for the CMA-ES algorithm can be improved; however, the final fitness score achieved is comparable to when it is not used.

C. OUR PROPOSED EvoShift ALGORITHM

Based on the above-discussed theory and cost function analysis, we present the details of the evolutionary strategy-based adversarial attack algorithm (EvoShift) as explained in Algorithm 1. We start the first generation of CMA-ES from initial perturbation parameter $\boldsymbol{\theta}_0 = (\boldsymbol{\mu}_0, \boldsymbol{C}_0)$. For each generation t, we sample multiple pixel perturbation parameters $\{\Delta_j\}_j$ and obtain the optimal neural network weights $\boldsymbol{\theta}^*$, by training a CNN from scratch on each such perturbation sample by minimizing the CE loss. After each generation, the sampling parameters are updated by the CMA-ES algorithm to retain the attacks corresponding to the top-performing costs .[1] The attack corresponding to the best performing cost across all generations is returned. We find that models trained on such samples show poor generalization, thus uncovering significant vulnerabilities in CNNs. It is

[1]More details on the CMA-ES algorithm can be found in the original paper [13].

Algorithm 1 EvoShift

Require: Training data $(x, y) \sim \mathcal{D}$, ES params $\boldsymbol{m}_0$, Σ_0, σ_0

1: **for** t from 0 to N_{gen} **do**
2: Sample a population of noise: $\{\boldsymbol{\Delta_j}\}_{j=1}^{\lambda} \sim N(\boldsymbol{\mu_t}, \Sigma_t)$, where λ is population size in a generation.
3: Fit j^{th} models: $\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}}[\mathcal{L}_{CE}(x + \boldsymbol{\Delta_j}, y; \theta)]$
4: Get the j^{th} semantic score as $\mathcal{S}_m^j = \frac{1}{N}\sum_{(x,y)\sim\mathcal{D}}\left[F_\theta(x)_y + \sum_{j\neq y}(1 - F_\theta(x)_j)\right]$
5: Train discriminator to classify between the true samples x and attacked samples $x + \delta_j$, and assign domain divergence cost as $\mathcal{S}_d^j = -2(1 - 2\epsilon)$
6: Compute total cost for the j^{th} sample $\mathcal{C}_j = \mathcal{C}_m^j + \mathcal{C}_d^j$.
7: Update ES parameters based on the fitness score $m_{t+1}, \Sigma_{t+1}, \sigma_{t+1} = \text{CMA-ES}(m_t, \boldsymbol{\Sigma}_t, \sigma, \{\mathcal{C}_j\}_j)$.
8: Store the solution with best fitness score in $\boldsymbol{\Delta}^*$
9: **end for**
10: **return** best solution: δ^* as output

to be noted that only samples from the training set were used to optimize the attack in the above algorithm. No test set samples were seen during attack optimization or model training. Figure 4 shows our proposed attacked training data on various datasets.

D. EXTENSION TO ATTACKS ON GENERATIVE ADVERSARIAL NETWORKS

The above attack analysis is targeted toward multi-class classification systems. However, this can also be used for attacking generative models that consist of a discriminative subsystem, namely GANs [54]. GANs are generative models that learn the data distribution of the training data, which can then be used for creating novel samples from it. It consists of a generator network (G) and a discriminator network (D) and a learning objective function consisting of a min-max optimization as shown below

$$\min_G \max_{D\in(0,1)} \mathbb{E}_{x\sim p_{data}} \log(D(x)) + \mathbb{E}_{z\sim p_z} \log(1 - D(G(z))), \quad (8)$$

where the discriminator classifies between samples from the generator and the true data distribution, providing a gradient signal for the generator to produce samples similar to the data distribution. Typically, the discriminator is trained using a CE loss to classify between the true image samples and generated samples, although some recent works use a different loss for discriminator learning [55], [56]. In this paper, we specifically use the implementation of [14], which uses a CE loss for discriminator training, which is a binary classifier in an encoder-decoder setting. We show that our proposed training time attack can also cause overfitting on the GAN's discriminator resulting in poor reconstruction of images, as shown in Figure 3 due to suppression of the semantic features.

E. EXTENSION TO ATTACKS ON LARGER DATASETS

In the above sections, we discussed the proposed attack being optimized for a particular dataset. Computing the optimal pixel attack requires training multiple CNNs on a small subset of the dataset for each generation, which can be computationally expensive for limited resources. Therefore we propose a method to transfer the learned pixel location from a smaller dataset to a larger dataset without using the expensive gradient-free attack optimization. We call this method Spatial Value Function (SVF) sampling. The idea is to sample pixels close to the attack locations of the source dataset and apply it to the target dataset. We compute the SVF by convolution with a Gaussian Point Spread Function (PSF) at the pixel perturbation location(s) for the source dataset as follows:

$$\text{SVF}_{t+1}(x, y) = \sum_j \sum_{(x_k,y_k)\in\delta_j} \text{PSF}(x - x_k, y - y_k) \dots \quad (9)$$

where the index j iterates over all classes and k iterates over the pixel corruptions in each class (depends on N_p). We extend the SVF to match the target image dimensions by performing bilinear interpolation. Finally, the pixel corruption location on the target dataset is obtained by importance sampling based on the reshaped SVF on the target dataset.

We assume that the source dataset and the target dataset have a spatially similar primary object location. We consider CIFAR10 as the source dataset. For the target dataset, we choose a subset of 64 × 64 ImageNet dataset with ten classes. For these two datasets, it is reasonable to assume that the object would be primarily located at the image center. We show that such transferred attacks to the ImageNet data successfully reduces testing accuracy in Section VIII-K.

In case the number of classes in the target dataset is more than that of the source dataset, we can sub-sample within the spatial value function to sample the required number of pixel attacks. For example, if the target dataset has C_t classes whereas the source has C_s classes, where $C_t > C_s$, we can create C_t small density distribution on the SVF using techniques such as Expectation Maximization [57]. Following that, sampling from each such small distribution would give us the pixel locations for the attack pixels. However, to ensure non-overlapping distributions, we also have to maximize the entropy of the overall system. This method would give well-separated distributions if C_t and C_s are not separated by large orders of magnitude, which otherwise would lead to overlapping pixel attacks for multiple classes. In such cases, intermediate fine-tuning using the proposed evolutionary strategy might be required to ensure good separability of attack pixels.

VI. EXPLAINING POOR GENERALIZATION TO PROPOSED PIXEL ATTACK

In this section, we identify that the drastic drop of generalization performance in CNNs for classification models [1], [2] and GANs [54] under our proposed training time attack, which is typically due to the over-reliance of the CE loss on the added noise in the image. Traditional training with the CE loss results in unconstrained mutual information maximization between the learned features and the target labels, exhibiting over-dependence on attack pixel features, even if

they are not semantically meaningful. We propose a robust feature learning scheme that preserves the semantic information by maximizing the mutual information between the latent features and input images to mitigate this problem.

Considering x as the input image and z as the latent features (logits), which is computed from the deep model as $z = F_\theta(x)$, the goal of the CE loss is to maximize the mutual information between the features and the target labels y. Such formulations are typical in image classification networks and GANs for discriminator learning.

Given a data distribution $(x, y) \sim \mathcal{D}$, the goal of the classification network is to maximize the mutual information $I(y; z)$ as stated in [29]. However, in the absence of priors, such methods can learn highly predictive features that do not align with the human perceptual system. Specifically, in our EvoShift algorithm, our noisy perturbations in the training images act as highly discriminative features, which leads to suppression of semantic features. From an information-theory point of view, the CE loss does not preserve the source distribution information. It only focuses on the high predictive features that lead to vulnerabilities in the presence of our proposed training time attack.

VII. ROBUST FEATURE LEARNING

In this section, we propose a novel loss function for training a CNN that is robust against overfitting to spurious pixel perturbations (similar to our proposed attack). We also specify metrics to measure the model's affinity to learn features based on noisy pixel artifacts (''nuisance'') or object properties such as shape and color (''semantics'').

A. ROBUST TRAINING WITH MUTUAL INFORMATION CONSTRAINTS

In the presence of our proposed perturbed training samples, neural networks overfit the spurious features leading to over-reliance on such spurious features. This results in an invariant response in the presence of such artifacts, which is termed as semantic invariance by [29]. The high semantic invariances induced in the neural network models can be attributed to the standard CE loss function's insufficiency, which favors choosing simple predictive features for the label rather than complicated features that require multi-layered reasoning. Therefore, under our proposed EvoShift, the CE loss learns a decision function based on the spurious input perturbations.

To alleviate this issue, Jacobsen *et al.* [29] designed a bijective neural network model that preserves the input information, thus capturing all variations in the input. However, standard neural network classifiers are not bijective by design. Therefore, there might be a loss of useful semantic information in such networks. Inspired by the concept of information preservation, we present a robust feature learning schematic that introduces an additional constraint in the objective function and the standard CE loss. In addition to maximizing the mutual information between the feature and the labels, learned features should maximize the mutual information $I(z; x)$ between the feature and image. This forces the latent representations to preserve class attributes such as shape and appearance, which are used to reconstruct the image features. Thus, the modified objective function can be formulated as

$$\max_{\theta}[-\mathcal{L}_{CE}(y, z; \theta) + I(z, x; \theta)] \quad s.t. \;\; I(z, x; \theta) < I_c, \tag{10}$$

where I_c is a bound on the mutual information without which we obtain the trivial solution $z = x$. The objective can be solved using the Lagrangian multiplier. Since it is intractable to find the marginal distribution $p(z)$ for the above objective, we minimize the upper bound of the regularized objective using an approximation of the marginal $r(z)$ following [38]. The robust objective can be written as

$$\begin{aligned} J(\boldsymbol{\theta}_E, \boldsymbol{\theta}_G, \boldsymbol{\theta}_C) = \mathbb{E}_{\boldsymbol{z}\sim E_{\theta_E}(\boldsymbol{z}|\boldsymbol{x})}\Bigg[& \overbrace{\sum_{i=1}^{C} -y_i D_{\theta_C}(\boldsymbol{z})_i}^{\text{Cross-Entropy Loss}} \\ & - \overbrace{\log q_{\theta_G}(\boldsymbol{x}|\boldsymbol{z})}^{\text{Decoder reconstruction loss}} \Bigg] \\ & + \beta \mathbb{E}_{\boldsymbol{x}\sim p(\boldsymbol{x})}\underbrace{[\mathrm{KL}[E_{\theta_E}(\boldsymbol{z}|\boldsymbol{x})||r(z)]]}_{\substack{\text{KL divergence between} \\ \text{encoder and marginal}}}. \end{aligned} \tag{11}$$

In the above loss function, we simultaneously optimize an encoder E_{θ_E}, a generator q_{θ_G}, and a classifier D_{θ_C}. This is implemented in practice by a Convolutional Variational Autoencoder [58] with the encoder and decoder networks. The classifier network D_{θ_C} is trained on the latent code from the bottleneck layer. During inference, the mean latent code is used as the feature for the classifier. We refer to this proposed loss function as ''vibCE'', due to its analytical similarity to the Variational Information Bottleneck (VIB) [38] based methods.

B. ROBUSTNESS EVALUATION METRICS

As explained in the previous section, there is a need to disentangle such factors of variations for systematic quantification of semantic features suppression by our proposed adversarial training time attack. Standard accuracy metrics cannot separate the effect of semantic and spurious features on classification performance. Therefore, we introduce two disentangled metrics, addressing the influence of semantic features and spurious task-irrelevant features on classification performance, which are described below.

Semantic Sensitivity (α_S): In the presence of predictive features due to our proposed training time pixel attack, the model learns spurious highly predictive features for encoding class information. However, a robust learning objective should learn to ignore such spurious features while focusing on semantic features such as image shape and appearance. This metric measures the contribution of task-relevant features toward classification performance by computing the test accuracy on clean data distribution in the absence of predictive nuisance features. A robust classifier

would produce test accuracy close to 1.0, even in the presence of our proposed pixel-wise attack. In contrast, compromised models will have test accuracy $\approx \frac{1}{N_c}$ (which is same as a random classifier). We define semantic sensitivity as

$$\alpha_S = \frac{\overbrace{\mathbb{E}_{(x,y)\in\mathcal{D}}\mathbb{I}[F_\theta^\delta(\boldsymbol{x})=y]}^{\text{test accuracy on clean data}} - \frac{1}{N_C}}{1.0 - \frac{1}{N_C}}, \tag{12}$$

where F_θ^δ refers to the classifier trained on attacked data. We normalize α_S to the range of [0, 1]. A robust classifier should have α_S close to 1.0.

Nuisance Sensitivity (α_N): This metric measures how easily a model can overfit in the presence of spurious features in the attacked training data. A robust model should produce the same network response irrespective of whether the true images are attacked or not. For example, if the pixel perturbation for class 9 is overlayed on images from an arbitrary class and the model predicts the label 9 for most images, it has high nuisance sensitivity. To measure this, we overlay each class-specific noise on all test images and measure the accuracy for that class over all classes. Thus, we define nuisance sensitivity (normalized between [0, 1]) as

$$\alpha_N = \frac{\overbrace{\mathbb{E}_{k\sim\text{Unif}(1,\ldots,N_C)}\mathbb{E}_{(x,y)\in\mathcal{D}}\mathbb{I}[F_\theta^\delta(\boldsymbol{x}+\boldsymbol{\delta_k})=k]}^{\text{test accuracy on noise overlayed images}} - \frac{1}{N_C}}{1.0 - \frac{1}{N_C}}, \tag{13}$$

where $\text{Unif}(1, \ldots, N_c)$ is the uniform sampling of class labels and $\mathbb{I}$ is the indicator function which counts the number of images that are classified as the attack class k. Robust classifiers should have α_N close to 0.0 because it should not respond to nuisance features that is added by our training time attacks.

VIII. EXPERIMENTAL RESULTS

In the experimental section, we perform extensive empirical analysis to address the following questions: (a) Can our proposed algorithm learn optimal noise configuration, which is better than randomly perturbed noise?, (b) Can our method outperform previous training time attack methods?, (c) How does our method affect partial attack on training data?, (d) Can popular regularization techniques defend against such attacks?, (e) Is our proposed attack effective in zero-shot transfer to other state-of-the-art CNN models?, (f) Does our attack perform zero-shot transfer to a new dataset?, (g) Are certain CNN optimization techniques robust against such attacks?, (h) Do input transformation-based methods provide defense against such attacks?, (i) Can our proposed attack for discriminative models also attack generative models such as GANs?, (j) Can our proposed robust feature learning defend against such attacks?, and (k) Can our proposed Spatial Value Function sampling successfully transfer attacks to larger datasets?

Datasets We test our algorithm on four datasets: MNIST, Fashion-MNIST, SVHN (cropped 32 × 32 images), and CIFAR10 images. Amongst them, two are grayscale and the other two are color image datasets which enable us to showcase the generality of our proposed method. We use four settings of number of pixel perturbation, $N_p = \{1, 2, 5, 10\}$. The perturbed MNIST images for $N_p = 1$ are shown in Figure 5(a). The optimal perturbations obtained by our algorithm were used to study the robustness to such attacks using three factors: *external regularization*, *model architecture*, and *optimization technique*. Learning perturbations by evolution involves multiple training rounds in each generation. We use two custom CNN models as underlying models in the evolutionary learning stage: GrayNet (24C3-P-48C3-P-256FC-10S) for MNIST and Fashion-MNIST and ColorNet (32C3-32C3-P-64C3-64C3-P-128C3-128C3-P-512FC-10S) for CIFAR10 and SVHN dataset.

A. LEARNING CURVES AND PERTURBATION SAMPLES

Here, we analyze the performance of our proposed CMA-ES-based attack optimization algorithm with increasing generations. The results are shown in Figure 5(c) and Figure 5(d) for MNIST and Fashion-MNIST datasets, respectively. For early generations, the pixel attack is sampled from uniformly distributed pixel locations. However, with the CMA-ES optimization, the spatial location, and the intensity of class-wise pixel attacks are optimized, which leads to a significant loss of generalization. This is evidenced by the accuracy of test samples, which drops as the optimization advances indicating the soundness of our proposed algorithm. The final learned samples are shown in Figure 4 for all four datasets. Neural networks trained on these attacked datasets show significantly low testing accuracy on clean samples.

GradCAM visualization [59] has been used in several prior works to visualize the spatial distribution of gradients in the input space. Higher CAM values indicate an increased contribution of the input pixel location to the output label. We visualize the mean GradCAM distribution of 100 images per class from the testing dataset corresponding to the true class label for the MNIST dataset for models trained on our proposed attacked dataset in Figure 5(b). The CAM distribution shifts its density to non-salient background ROI in the image, thus learning non-discriminative features that do not generalize well. This might explain the drop in testing accuracy with increasing epochs.

B. COMPARISON TO PRIOR METHODS

We believe our work is the first attempt towards adversarial training time attack using discrete pixel attacks to induce overfitting in neural networks. There are not many previous works on this topic. We choose the work of Jacobsen *et al.* [29] as our baseline for prior models that use heuristic pixel placement for attacking the training images. Our method consistently outperforms the baseline method on the metric of test accuracy on the clean test set for all datasets, as shown in Table 1. Our method shows superior performance compared to [29] because we perform optimization to search for the best corruption pattern, whereas the baseline uses a

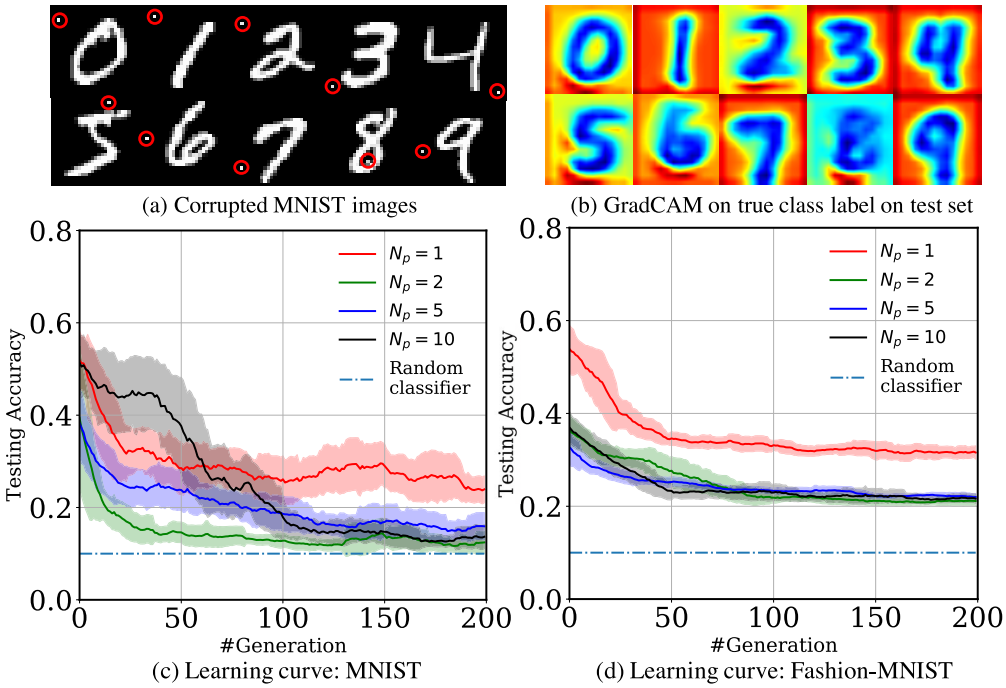


FIGURE 5. (a) Highlighting learned single pixel perturbations on MNIST images, (b) GradCAM visualization of the last Conv layer for $N_p = 1$. Dominant gradient distribution is in the background. Learning curve with increasing generations of CMA-ES is shown for (c) MNIST and (d) Fashion-MNIST.

TABLE 1. Testing accuracy (in %) on clean test samples, trained on attacked samples with data augmentation for 30 epochs on the SVHN dataset. Experiments are repeated three times. Our attack method outperforms the previous attack method outlined in [29] due to perturbation optimization using CMA-ES.

Method	ResNet-20	ResNet-32	DenseNet-40
SVHN (clean)	93.5 ± 0.9	92.8 ± 1.0	92.3 ± 1.2
$N_p = 1$ [29]	91.8 ± 0.2	90.9 ± 1.8	91.0 ± 0.4
$N_p = 1$ [ours]	31.3 ± 6.3	37.2 ± 10.4	32.1 ± 9.4
$N_p = 2$ [ours]	14.9 ± 2.4	18.4 ± 3.8	18.8 ± 4.7
$N_p = 5$ [ours]	$\mathbf{9.3 \pm 0.9}$	$\mathbf{11.0 \pm 0.3}$	$\mathbf{16.1 \pm 8.4}$

heuristic pixel placement to corrupt the data. These experiments were performed in the presence of random image crop data augmentation.

Qualitatively, our method has similar human imperceptibility as one-pixel evasive attacks such as [26] because both methods use one-pixel attacks for fooling CNNs. However, compared to JSMA [16] which corrupts multiple pixels, our method has better human imperceptibility making it harder to detect. Compared to other training time attacks such as [22], [29], our method has better imperceptibility because it only attacks a few pixels.

C. PARTIALLY ATTACKING FEW CLASSES

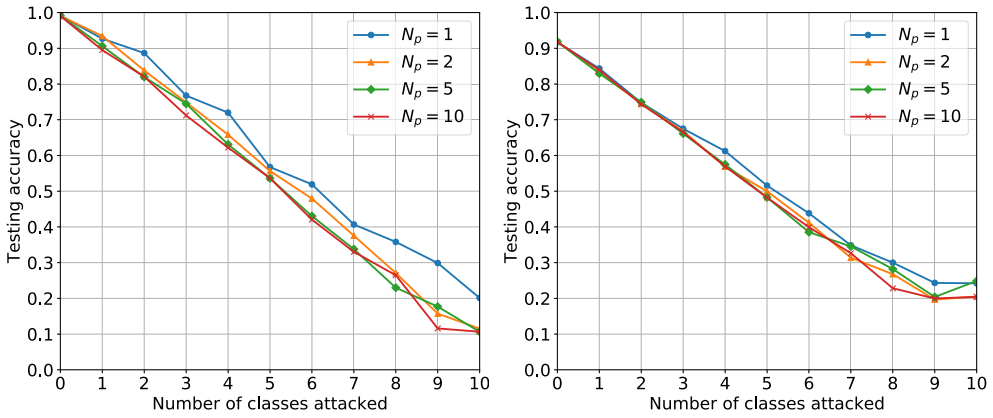
We also show the result of training CNNs (GrayNet) with partial class-wise attacks. For example, if we choose the number of classes to attack as k, then training images belonging to classes $\{0, 1, \ldots k-1\}$ are corrupted by our proposed pixel-based attack, and other images are kept intact. Figure 6 shows the result of testing accuracy on the test data for MNIST and Fashion-MNIST data when the training images are incrementally corrupted for each class. The results show a linear trend of decreasing testing accuracy as more classes are incrementally corrupted, suggesting that our attacks can partially confuse the CNNs for the specific attacked classes while the other classes are correctly classified. Therefore, our proposed attack acts as a mask that hides class-specific features and makes the CNN overfit the spurious pixel disturbance.

D. EXPLICIT REGULARIZATIONS ARE EASILY OVERFITTED

We are interested in understanding how different factors in neural network learning contribute to robustness against our proposed attacks. Zhang *et al.* [35] showed that explicit regularization methods have a limited effect in controlling neural networks fitting random noise and labels. In the same spirit, we set up experiments to study the robustness of commonly used regularization techniques against our crafted attacks. Testing accuracy on these methods is shown in Figure 7.

1) DATA AUGMENTATION

We use random image transformations such as cropping, flipping, and zooming, which are used to augment the training data. Data augmentation is the most effective explicit regularization according to our study. This is not surprising because it introduces disturbances to the training data distribution,



(a) Partial attack : MNIST (b) Partial attack: Fashion-MNIST

FIGURE 6. Partially attacking a few images in the training data for (a) MNIST and (b) Fashion-MNIST images. In the x-axis, we show the total number of classes that are attacked. We see that with increasing the number of classes of attack, the testing accuracy almost linearly decreases. This is because our pixel noise selectively attacks samples from a few classes only for the partial attack scenario.

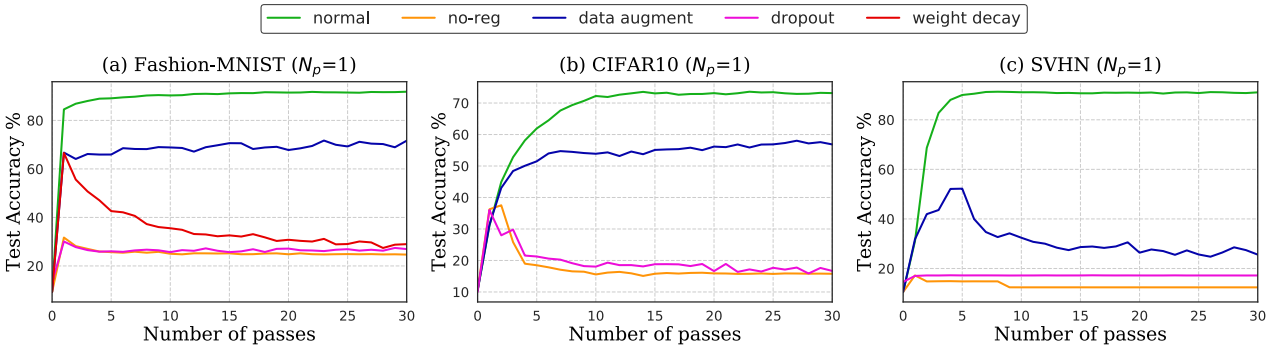


FIGURE 7. Testing accuracy with increasing training epochs for different regularization methods under a single-pixel ($N_p = 1$) attack. *normal* refers to no corruption in training data, and *no-reg* refers to the case where no explicit regularization was used. Experiments were repeated five times and the mean is reported. Training accuracies are close to 100%.

thus diminishing the effect of the perturbation, which was designed on a fixed dataset with data augmentation.

2) DROPOUT [60]

This regularization technique randomly masks layer outputs to reduce the reliance on the output on particular neurons. We used a dropout probability of 0.4. However, dropout seems to have little to no effect on the generalization ability. Since the perturbations are extremely localized in space, we believe that dropout has a negligible effect in consistently masking such spurious artifacts.

3) WEIGHT DECAY

This method constrains the norm of the parameters with a Euclidean ball whose radius is determined by the λ coefficient. It is also known as l_2 regularization or Tikhonov regularization [61]. We use $\lambda = 0.01$. Although weight decay marginally improved the test accuracy at initial epochs, final test accuracy after 30 epochs is similar to that without regularization.

E. ATTACK TRANSFER ACROSS MODELS

Previous works have shown that the choice of model architecture can act as implicit regularization. Statistical machine learning theory predicts that models with a larger number of parameters have higher complexity, making them more likely to converge at a local minimum with poor generalization ability. Li *et al.* [63] showed that ResNet [1] with skip connections produced a smooth loss surface compared to those without skip connections, hinting that the model architecture might play some role in generalization performance. We train state-of-the-art CNN models (with data augmentation), Resnet-20, Resnet-32 [1], and DenseNet-40 [2] on our attacked training samples learned from our custom-designed CNN models and measure the testing accuracy after 30 epochs, as shown in Table 2.

Empirical evaluation reveals a significant difference in test accuracy for unperturbed train images and even single-pixel perturbed data. For different perturbation levels, we do not find a strong correlation between depth and testing accuracy for ResNet models. For example, while ResNet-20 produces

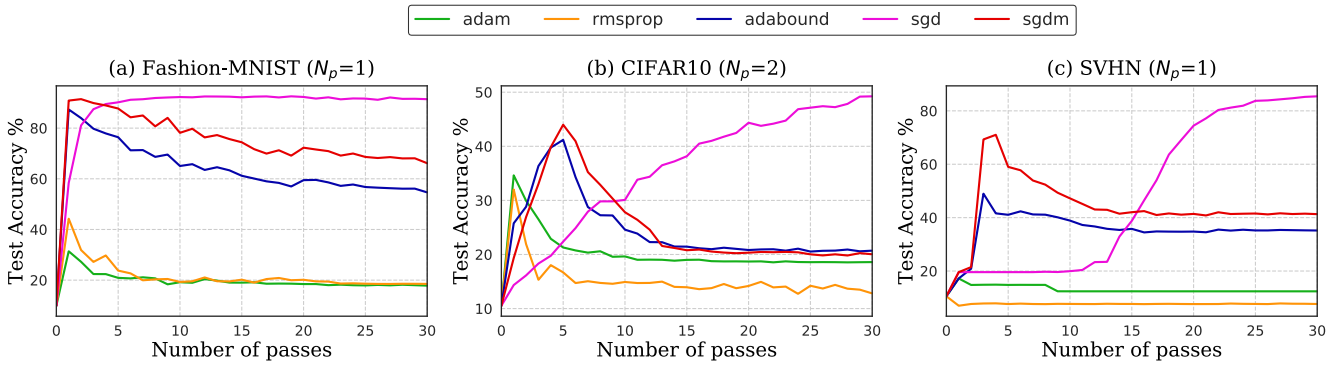


FIGURE 8. Testing accuracy using various optimization strategies under single-pixel perturbation shows SGD consistently performs better than adaptive optimization techniques. Each experiment was performed five times and the mean is reported.

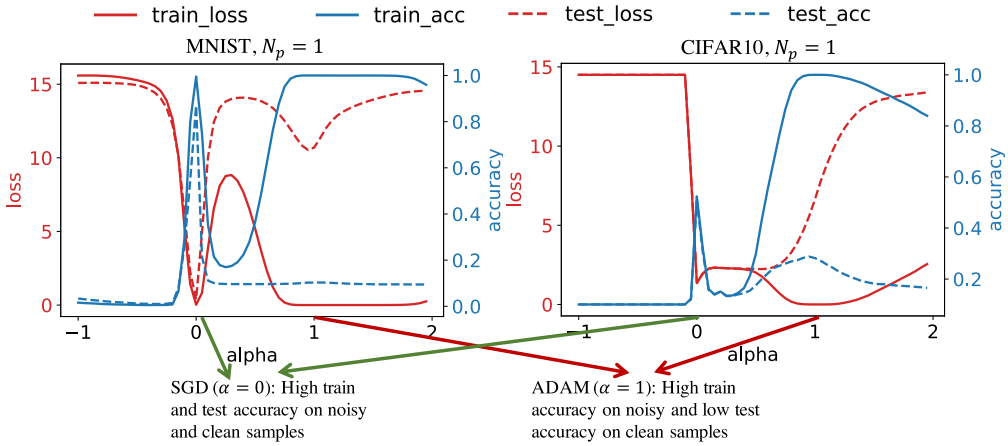


FIGURE 9. Loss surface by interpolating from SGD ($\alpha = 0$) to Adam ($\alpha = 1$) weights. The loss surface around the SGD parameter is sharper; however, it has better generalization.

TABLE 2. Testing accuracy (in %) on clean test samples when trained on our proposed attacked samples with data augmentation for 30 epochs. Experiments are repeated three times. Our attacks learned on our custom ColorNet model can transfer to state-of-the-art CNN architecture, causing overfitting with low test and high training accuracy.

Dataset	N_p	ResNet-20	ResNet-32	DenseNet-40
CIFAR10	0	78.5 ± 1.2	75.2 ± 3.0	82.3 ± 1.5
	1	33.3 ± 8.4	30.7 ± 4.2	29.9 ± 2.9
	2	25.5 ± 1.1	$\mathbf{20.7 \pm 6.2}$	23.4 ± 0.7
	5	$\mathbf{14.5 \pm 2.6}$	21.4 ± 1.1	$\mathbf{24.6 \pm 5.2}$
SVHN	0	93.5 ± 0.9	92.8 ± 1.0	92.3 ± 1.2
	1	31.3 ± 6.3	37.2 ± 10.4	32.1 ± 9.4
	2	14.9 ± 2.4	18.4 ± 3.8	18.8 ± 4.7
	5	$\mathbf{9.3 \pm 0.9}$	$\mathbf{11.0 \pm 0.3}$	$\mathbf{16.1 \pm 8.4}$

better test accuracy on the CIFAR10 dataset than ResNet-32, the opposite is true for SVHN. Therefore, in the presence of such conflicting evidence, it is difficult to convincingly conclude that shallower models are more robust to overfitting than their deeper counterparts.

F. ATTACK TRANSFER ACROSS DATASETS

Similar to testing the transfer of our attack across models, we also show that our proposed attack can be transferred across datasets as well. For this purpose, we trained our CNN model (**ColorNet**) on the source dataset, CIFAR10, with and without the proposed attack. The same model is then tested on the STL10 dataset [64] which has similar labels to the CIFAR10 datasets. We choose the common labels between two datasets, `{airplane, cat, deer, dog, ship, truck}`, for reporting the test accuracy. Table 3 shows the accuracy on the test set for both the datasets. For clean images, images belonging to the same labels show similar accuracy. However, when trained on our proposed attacked images, the testing accuracy shows a drop for both the source and target datasets. Although the attack

TABLE 3. Transferring our attack across datasets from CIFAR10 to STL10 dataset. We show that the drop in test accuracy due to our attacks on the source dataset can also be transferred to a target dataset.

	Clean		$N_p = 1$		$N_p = 5$	
	CIFAR	STL	CIFAR	STL	CIFAR	STL
Airplane	75.1	74.5	22.8	26.5	18.3	11.3
Cat	60.5	31.25	16.9	13.1	15.9	7.3
Deer	69.3	61.1	71.8	24.6	4.9	2.6
Dog	56.9	18.9	14.6	8.3	19.7	13.6
Ship	82.0	64.5	7.7	3.5	6.6	13.8
Truck	76.6	52.9	25.8	30.6	41.1	23.3

TABLE 4. Comparison of various learning objectives based on proposed performance metrics. Our proposed loss function (vibCE) has significantly higher semantic feature sensitivity and lower nuisance feature sensitivity (which is desirable for robust classifiers) compared to CE loss due to better semantic feature preservation.

		Semantic Feature Sensitivity (α_S) ↑				Nuisance Feature Sensitivity (α_N) ↓			
		MNIST		F-MNIST		MNIST		F-MNIST	
		Random	EvoShift	Random	EvoShift	Random	EvoShift	Random	EvoShift
Clean	CE	0.99		0.9		0.02		0.01	
	vibCE (ours)	0.98		0.84		0.02		0.01	
$N_p = 1$	CE	0.30	0.01	0.32	0.16	0.92	0.93	0.94	0.89
	vibCE (ours)	**0.95**	**0.94**	**0.72**	**0.73**	**0.01**	**0.01**	**0.09**	**0.09**
$N_p = 2$	CE	0.32	0.01	0.30	0.13	0.99	1.00	0.99	0.72
	vibCE (ours)	**0.94**	**0.94**	**0.75**	**0.64**	**0.03**	**0.02**	**0.09**	**0.11**
$N_p = 5$	CE	0.15	0.00	0.27	0.12	1.00	1.00	0.97	0.99
	vibCE	**0.93**	**0.90**	**0.70**	**0.63**	**0.09**	**0.07**	**0.20**	**0.27**

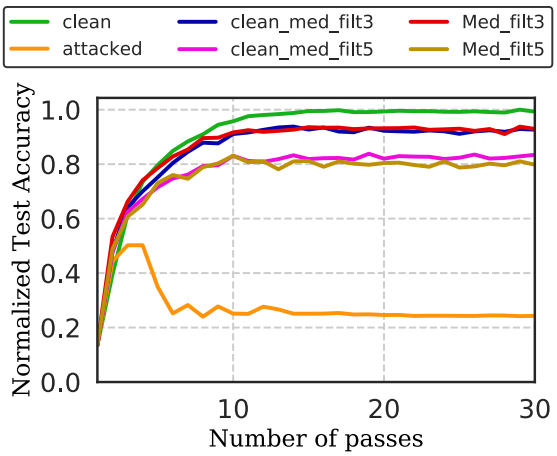


FIGURE 10. Median filtering based defense against our proposed pixel-based noise for CIFAR10 dataset. Median filtering improves the performance due to the removal of pixel-noise. However, it also brings down the performance of training on clean images due to the removal of certain high-frequency features due to the filtering process.

pixels were not trained for the STL10 dataset, our attack is shown to transfer to other datasets also in a zero-shot manner.

G. ADAPTIVITY CAN OVERFIT TO PROPOSED ATTACKS

High out-of-sample error is generally attributed to poor convergence of the neural network parameters to an unfavorable local minimum. By examining the robustness of well-known optimization strategies to our pixel-wise attacks, we wish to study if a certain algorithm is more liable to memorizing small perturbations while ignoring other salient statistical patterns in the training data. To this end, we trained CNN models on single-pixel perturbed data using Adam [65], SGD, RMSProp [66], and Adabound [67] optimization. The results are shown in Figure 8.

Wilson *et al.* [37] showed that adaptive methods are affected by spurious features that do not contribute to out-of-sample generalization by crafting a smart artificial linear regression example. Our method can be viewed as a generalization of such methods for the automatic creation of such spurious examples that scale to arbitrarily sized datasets by gradient-free evolutionary strategies. Figure 8 reveals that Adam and RMSProp show prohibitively low testing accuracy for all cases while vanilla SGD is surprisingly resilient to such perturbations showing better out-of-sample performance consistently for all datasets. Adabound uses strategies from both SGD and Adam, thus showing intermediate performance. It can be concluded that adaptive methods heavily overfit training input perturbations while vanilla SGD is considerably robust to such changes.

1) LOSS SURFACE

Keskar *et al.* [68], Hochreiter and Schmidhuber [69] claimed that flatter minima solutions generalize better than their sharper counterparts. To investigate this phenomenon, we visualize the loss surface around the learned parameters by interpolating the weights obtained from SGD and Adam optimization following the strategy by Goodfellow *et al.* [70]. We plot the loss function values and train/test accuracies at intermediate intervals given as

$$\boldsymbol{w}_\alpha = \alpha \boldsymbol{w}_{\text{Adam}} + (1-\alpha)\boldsymbol{w}_{\text{SGD}}, \tag{14}$$

as shown in Figure 9. Interestingly, we find that SGD finds sharper minima solutions where both testing and training losses are lower ($\alpha = 0$) than Adam, whereas the training loss exhibits a flatter geometry ($\alpha = 1$). This pattern is repeatedly visible for all datasets suggesting that sharpness of minima does not guarantee a solution that has better generalization robustness to training perturbations, which is along the same line of argument as claimed by Dinh *et al.* [71].

H. EFFECT OF INPUT TRANSFORMATIONS

Previous defensive methods in adversarial defense [46] show that input transformation-based defenses are effective against a certain class of adversarial attacks. Therefore, we compare our proposed method against two kinds of input transformations: (i) median filtering and (ii) additive Gaussian noise of varying severity.

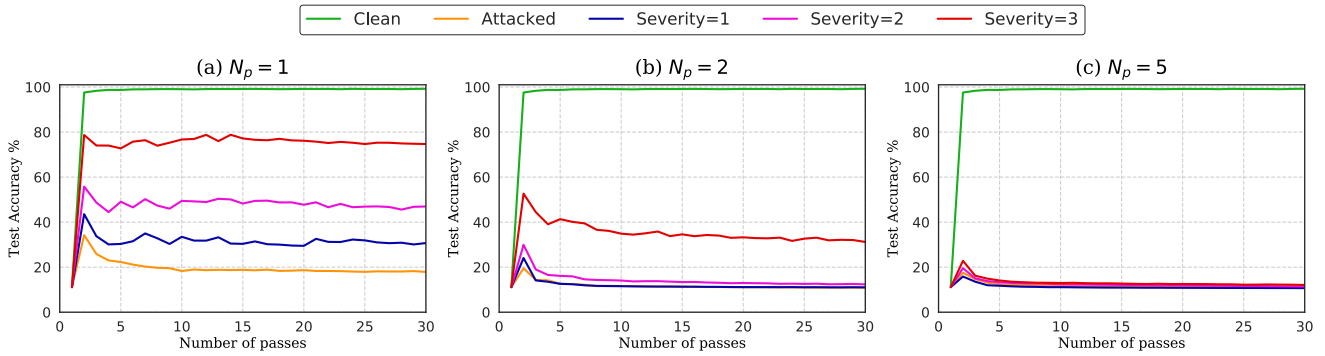


FIGURE 11. Effect of adding Gaussian noise [62] on the attacked images using our proposed pixel-based attack for MNIST dataset. With the increasing severity of the additive Gaussian noise, the defensive properties against our pixel attack are improved. However, for more strength of the pixel attack, additive Gaussian noise is incapable of providing suitable defense.

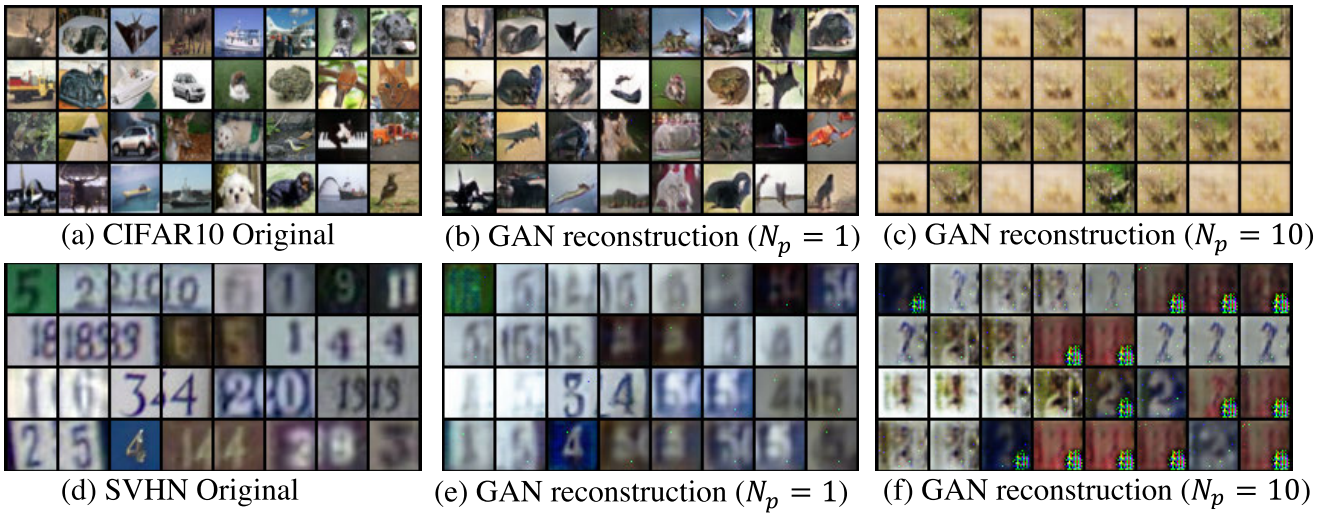


FIGURE 12. Generated samples by GANs on attacked data distribution show that semantic features in the true samples are suppressed by our proposed training attacks resulting in poor reconstruction of images. The quality of reconstructed images degrades with increasing attack strength. Spurious features are, however, faithfully reconstructed, indicating over-reliance on such artifacts by the discriminator.

1) MEDIAN FILTERING

Since median filtering is well-known to prevent salt-and-pepper noise, we applied median filtering on our attacked images to find the resilience of our method against such a defensive strategy. Figure 10 shows the effect of median filtering on the CIFAR10 dataset for the number of pixel attacks, $N_p = 1$ as measured by normalized test accuracy with respect to the clean training data accuracy. In the presence of median filtering, we observe improvement in the testing accuracy when compared to the attacked training scenario. However, there is a slight drop in the clean testing accuracy as well when median filtering is applied. This is due to the removal of certain detailed features in the image due to the filtering process.

Although median filtering improves the testing accuracy in the presence of our proposed training-time attack, this is also the case for well-known adversarial attacks [5]–[7], [15] that can be defended against by simple input transformation-based methods proposed in [46], [72]. Therefore, similar to previous methods in adversarial attacks, we believe this does not undermine the contribution of this paper, which proposes a novel method for attacking training images, thus uncovering a new kind of vulnerability in neural network learning.

2) ADDITIVE GAUSSIAN NOISE

We wanted to test the effect of adding random noise to our proposed pixel-based attack on training images. To that effect, we added Gaussian noise from [62] with various severity on the attacked images. Details on the severity levels for the Gaussian noise is explained in [62]. Figure 11 shows the effect of testing accuracy when the CNN model is trained in the presence of both our proposed pixel attack and random Gaussian noise of various severity. The results show that for a few pixel attacks, high severity Gaussian noise can provide defense against our attacks. We hypothesize this is due to the masking of the pixel attacks by the additive noise which diffuses its strength. However, with more number of pixel attacks, additive Gaussian noise does not provide much defense as shown by the low test accuracy for the

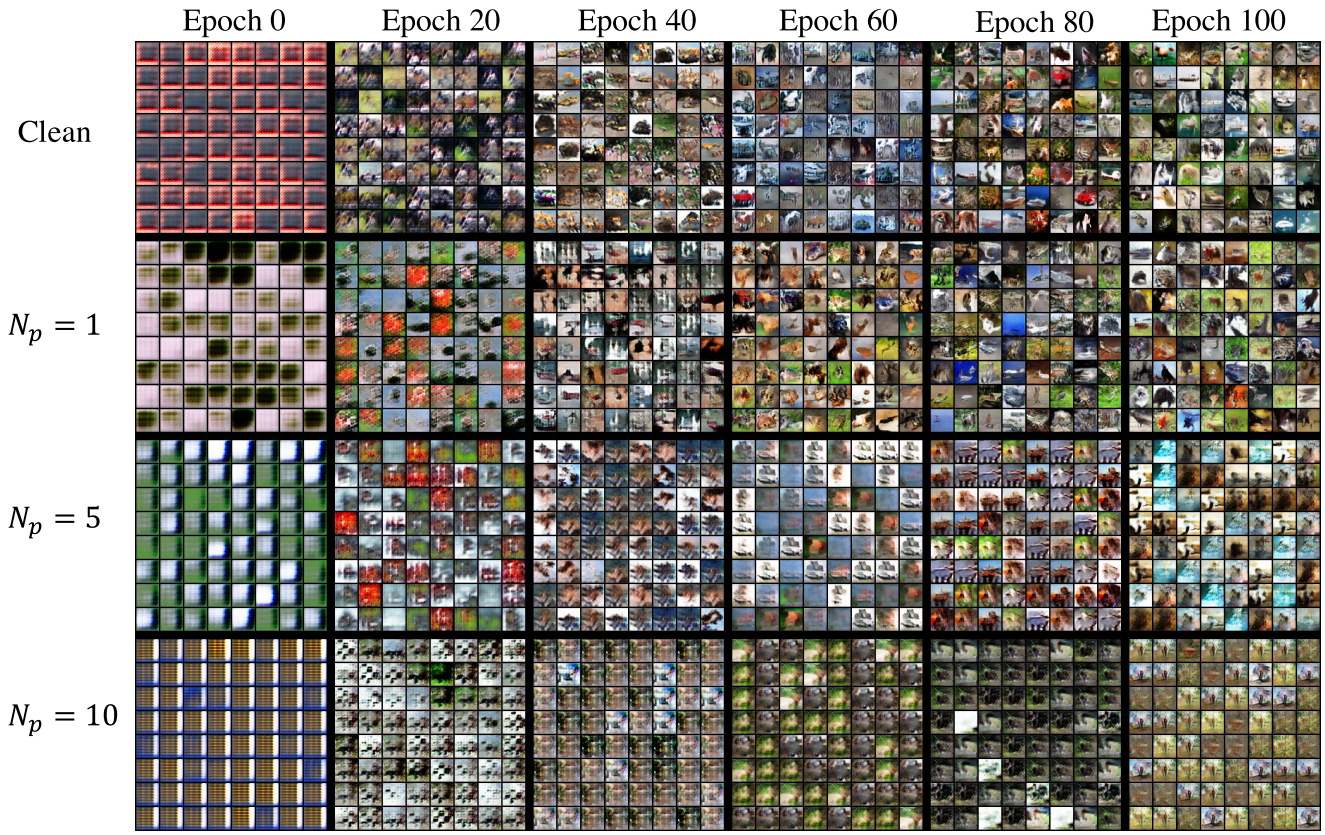


FIGURE 13. Progression of GAN training in the presence of our proposed training-time attack data using VAEGAN [14]. For the 'clean' data, the generated images resemble natural images with the progression of training. However, in the presence of our proposed pixel attack, the generated images do not generate natural features such as object shape and color but enhance the attacked pixels.

$N_p = 5$ case. Therefore, our proposed attack (with a high number of pixels) is resilient against additive random noise.

I. POOR GENERATIVE ADVERSARIAL NETWORK RECONSTRUCTION UNDER EvoShift

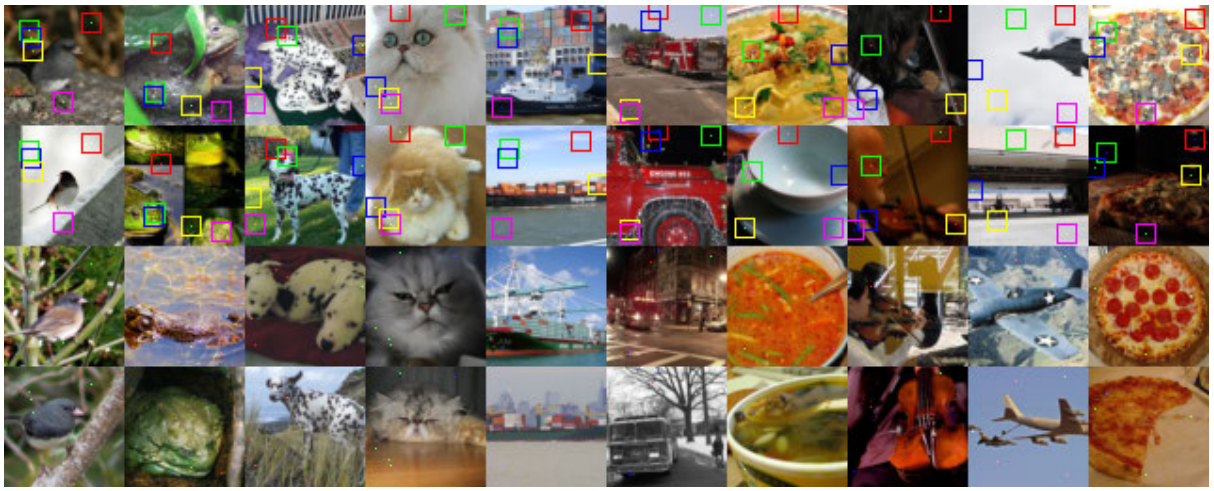
Since GANs use the CE loss in the discriminator for classification between generated images and images from data distribution, our proposed training time attacks can confuse the discriminator. To study this effect, we sample images from our proposed EvoShift-ed version of standard datasets, $x \sim \mathcal{D}_{adv}$, and learn VAEGAN [14] on such images. We show qualitative comparisons of the GAN reconstruction under our proposed attack in Figure 12. We find that spurious few-pixel perturbations can effectively mask the true data distribution, resulting in large degradation of reconstructed images from the GAN. With the increasing strength of the number of pixels in the attack, the quality of reconstruction increasingly degrades, indicating the high nuisance sensitivity of GAN discriminators.

Corresponding to the reconstructions in Figure 12, for CIFAR10 images, we obtain a Peak Signal to Noise ratio (PSNR) of 14.42 dB for $N_p = 1$ and 11.06 dB for $N_p = 10$. For the SVHN dataset, these values are 16.20 dB and 13.49 dB, respectively. Quantitative analysis demonstrates poor reconstructed image qualities in terms of low PSNR by VAEGAN under our proposed EvoShift. This implies that with the increasing strength of the attack, GANs ignore semantic features in the images and confuse the spurious artifacts as true data distribution, which is an undesirable vulnerability in generative models that have not been studied in detail by previous works.

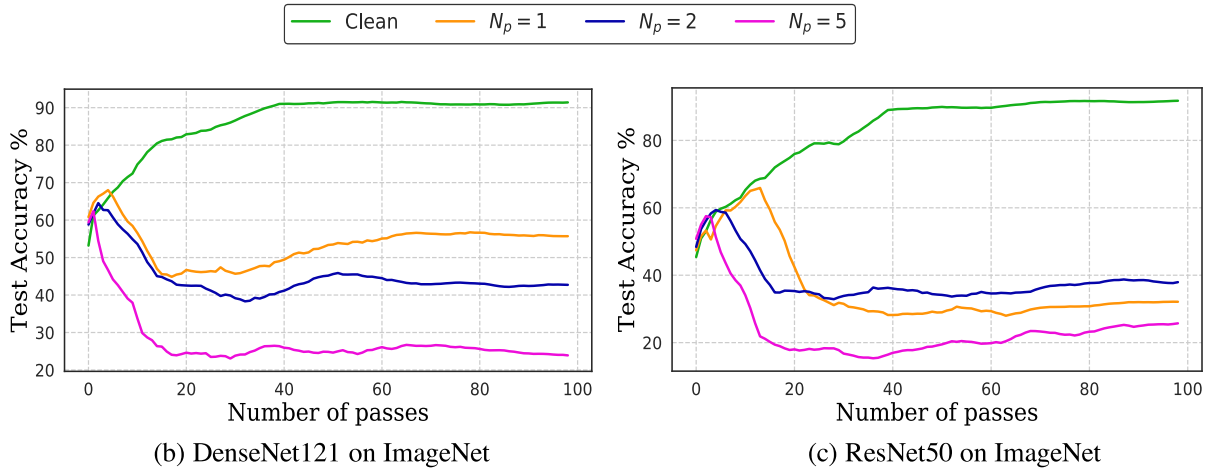
Figure 13 shows the progression of GAN training in the presence of our training-time pixel-based attacks for the CIFAR10 dataset. For clean images, the GAN progressively learns to generate semantic features that are related to natural images. On the other hand, in the presence of our training-time attacks, the GAN model cannot generate the semantic features but focuses on generating the noisy pixel features. With the increasing strength of the attack, as measured by the number of pixels, the generated images show increased natural feature suppression with an increased focus on generating the noisy pixels. This is due to the discriminator which can be attacked by our proposed pixel-based method that overfits the noisy pixel features.

J. ROBUSTNESS OF OUR VARIATIONAL OBJECTIVE

Table 4 provides the performance comparison in terms of the proposed metrics for different loss functions: CE and our proposed vibCE corresponding to the MNIST and Fashion-MNIST dataset, respectively. *Random* refers to a



(a) Training attack on ImageNet dataset with 10 classes for $N_p = 5$ sampled from CIFAR10 as the source dataset using spatial value function (SVF) method. The top two rows show the attack pixels highlighted by colored rectangles.



(b) DenseNet121 on ImageNet (c) ResNet50 on ImageNet

FIGURE 14. Degrading discriminative performance on ImageNe dataset with increasing pixel perturbation strength.

uniform spatial sampling of pixel perturbation as the attack. This is the case where no training-time attack optimization has been performed using the evolutionary strategy and corresponds to the initial solution of the optimization. We infer two major insights from the results: (1) Our proposed EvoShift outperforms the random attack sampling case shown by lower α_S and high α_N for both CE and vibCE loss. This shows that our proposed EvoShift algorithm finds suitable pixel attack parameters that overfit the model to training data that is not possible by attacking with random pixel placement, (2) Our robust objective demonstrates significantly higher semantic sensitivity (α_S) and low nuisance sensitivity (α_N) compared to the CE loss. Training with the vibCE loss function retains the semantic features related to shape and color information and thus does not overfit the additive adversarial pixel attacks during training.

K. SCALING ATTACK TO ImageNet DATASET

In this paper, we are trying to show vulnerabilities in neural networks by attacking training data (not test data) which requires multiple training on the perturbed dataset. Due to computational expenses, this is difficult to achieve. Finding the worst-case training time noise on the full ImageNet dataset would require training 1000-way classifiers for each noise for each generation of our evolutionary algorithm. However, we propose a method using Spatial Value Functions (SVF) that alleviates this problem by sampling from a smaller attacked dataset sample. We show that using the SVF sampling method, we can successfully scale such attacks from CIFAR10 to ImageNet samples. Figure 14(a) shows a few samples from our training pixel perturbations for 64×64 ImageNet [73] dataset using our proposed SVF based transfer from CIFAR10 dataset which has an image shape of 32×32. Figures 14(b)-14(e) show degradation in classification performance under attacked training images for the ResNet-50, ResNet-101 [1], and DenseNet-161 [2] models. We see that even a single pixel attack on the training images can bring down the testing accuracy on clean images to almost 50%. Increasing the attack severity to $N_p = 5$ and $N_p = 10$ can further degrade the testing accuracy. Thus, our proposed SVF

methods for transferring attacks from source to target dataset is effective and can reduce the testing accuracy, without the need for recomputing the pixel optimization by CMA-ES. However, performing CMA-ES in addition to SVF based transfer might even strengthen the attack further.

IX. CONCLUSION

We presented an adversarial training time attack using a population-based evolutionary strategy along with a novel fitness score designed to explicitly maximize domain divergence and generalization gap. We observed that it is possible to fool neural networks with each passing generation suggesting that specific spatial locations exist on the input image that are more vulnerable to being attacked than others. This result exposed serious vulnerabilities in CNNs. Our analysis revealed that a proper selection of the optimization technique is paramount to good generalization properties. We found that SGD performs significantly better than adaptive optimization methods in ignoring spurious training features that do not contribute to the out-of-sample generalization. Our analysis of loss surface revealed that SGD finds sharper minima solutions despite good generalization performance. Such training distribution attacks can also be extended to GAN discriminators causing poor reconstruction of semantic components in the image. This work is one of the first works in the field of attacking GANs using spurious adversarial noise in training data. Furthermore, we showed that this vulnerability in neural networks is related to the inefficiency of the CE loss. We also proposed a robust loss function based on variational inference principles that increase the mutual information between semantic features and the labels resulting in improved performance measured by the sensitivity measures. In this paper, we provided an extensive analysis of the behavior of CNNs in the presence of intelligently crafted adversarial training noise. We believe that this work will fuel further research into understanding the robustness of deep learning algorithms regarding generalization in the presence of training time adversarial attacks.

REFERENCES

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[2] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[4] V. Vapnik, *The Nature of Statistical Learning Theory*. Springer, 2013.

[5] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*. [Online]. Available: http://arxiv.org/abs/1312.6199

[6] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*. [Online]. Available: http://arxiv.org/abs/1706.06083

[7] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.

[8] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," 2018, *arXiv:1802.00420*. [Online]. Available: http://arxiv.org/abs/1802.00420

[9] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[11] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," 2018, *arXiv:1802.05365*. [Online]. Available: http://arxiv.org/abs/1802.05365

[12] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," 2018, *arXiv:1810.04805*. [Online]. Available: http://arxiv.org/abs/1810.04805

[13] N. Hansen, "The CMA evolution strategy: A tutorial," 2016, *arXiv:1604.00772*. [Online]. Available: http://arxiv.org/abs/1604.00772

[14] A. Boesen Lindbo Larsen, S. Kaae Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," 2015, *arXiv:1512.09300*. [Online]. Available: http://arxiv.org/abs/1512.09300

[15] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*. [Online]. Available: http://arxiv.org/abs/1412.6572

[16] 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 (EuroS&P)*, Mar. 2016, pp. 372–387.

[17] N. Narodytska and S. Prasad Kasiviswanathan, "Simple black-box adversarial perturbations for deep networks," 2016, *arXiv:1612.06299*. [Online]. Available: http://arxiv.org/abs/1612.06299

[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," 2017, *arXiv:1709.04114*. [Online]. Available: http://arxiv.org/abs/1709.04114

[19] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9185–9193.

[20] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," 2016, *arXiv:1611.01236*. [Online]. Available: http://arxiv.org/abs/1611.01236

[21] S. Sabour, Y. Cao, F. Faghri, and D. J. Fleet, "Adversarial manipulation of deep representations," 2015, *arXiv:1511.05122*. [Online]. Available: http://arxiv.org/abs/1511.05122

[22] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," 2012, *arXiv:1206.6389*. [Online]. Available: http://arxiv.org/abs/1206.6389

[23] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6103–6113.

[24] J. Steinhardt, P. W. W. Koh, and P. S. Liang, "Certified defenses for data poisoning attacks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 3517–3529.

[25] S. Chaudhury and T. Yamasaki, "Investigating generalization in neural networks under optimally evolved training perturbations," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 3612–3617.

[26] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.*, vol. 23, no. 5, pp. 828–841, Oct. 2019.

[27] P. W. Koh and P. Liang, "Understanding black-box predictions via influence functions," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1885–1894.

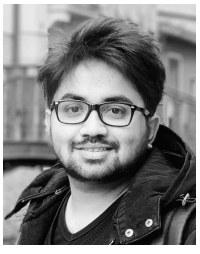
[28] T. Goldstein, C. Studer, and R. Baraniuk, "A field guide to forward-backward splitting with a FASTA implementation," 2014, *arXiv:1411.3406*. [Online]. Available: http://arxiv.org/abs/1411.3406

[29] J.-H. Jacobsen, J. Behrmann, R. Zemel, and M. Bethge, "Excessive invariance causes adversarial vulnerability," 2018, *arXiv:1811.00401*. [Online]. Available: http://arxiv.org/abs/1811.00401

[30] T. Tanay, J. T. A. Andrews, and L. D. Griffin, "Built-in vulnerabilities to imperceptible adversarial perturbations," 2018, *arXiv:1806.07409*. [Online]. Available: http://arxiv.org/abs/1806.07409

[31] L. Wu, Z. Zhu, and E. Weinan, "Towards understanding generalization of deep learning: Perspective of loss landscapes," 2017, *arXiv:1706.10239*. [Online]. Available: http://arxiv.org/abs/1706.10239

[32] B. Neyshabur, S. Bhojanapalli, D. McAllester, and N. Srebro, "Exploring generalization in deep learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5947–5956.
[33] G. Valle-Pérez, C. Q. Camargo, and A. A. Louis, "Deep learning generalizes because the parameter-function map is biased towards simple functions," 2018, *arXiv:1805.08522*. [Online]. Available: http://arxiv.org/abs/1805.08522
[34] M. Hardt, B. Recht, and Y. Singer, "Train faster, generalize better: Stability of stochastic gradient descent," 2015, *arXiv:1509.01240*. [Online]. Available: http://arxiv.org/abs/1509.01240
[35] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," 2016, *arXiv:1611.03530*. [Online]. Available: http://arxiv.org/abs/1611.03530
[36] C. Zhang, Q. Liao, A. Rakhlin, K. Sridharan, B. Miranda, N. Golowich, and T. Poggio, "Theory of deep learning III: Generalization properties of SGD," Center Brains, Minds Mach. (CBMM), State of Minas Gerais, Brazil, Tech. Rep. CBMM Memo 067, 2017.
[37] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, "The marginal value of adaptive gradient methods in machine learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4148–4158.
[38] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," 2016, *arXiv:1612.00410*. [Online]. Available: http://arxiv.org/abs/1612.00410
[39] Y. Song, T. Kim, S. Nowozin, S. Ermon, and N. Kushman, "PixelDefend: Leveraging generative models to understand and defend against adversarial examples," 2017, *arXiv:1710.10766*. [Online]. Available: http://arxiv.org/abs/1710.10766
[40] D. Meng and H. Chen, "MagNet: A two-pronged defense against adversarial examples," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 135–147.
[41] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *Proc. IEEE Inf. Theory Workshop (ITW)*, Apr. 2015, pp. 1–5.
[42] R. Shwartz-Ziv and N. Tishby, "Opening the black box of deep neural networks via information," 2017, *arXiv:1703.00810*. [Online]. Available: http://arxiv.org/abs/1703.00810
[43] D. B. F. Agakov, "The IM algorithm: A variational approach to information maximization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, vol. 16, no. 320, p. 201.
[44] M. Ishmael Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R. Devon Hjelm, "MINE: Mutual information neural estimation," 2018, *arXiv:1801.04062*. [Online]. Available: http://arxiv.org/abs/1801.04062
[45] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2016, pp. 582–597.
[46] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," 2017, *arXiv:1704.01155*. [Online]. Available: http://arxiv.org/abs/1704.01155
[47] J. Cohen, E. Rosenfeld, and Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 1310–1320.
[48] H. Lee, S. Han, and J. Lee, "Generative adversarial trainer: Defense to adversarial perturbations with GAN," 2017, *arXiv:1705.03387*. [Online]. Available: http://arxiv.org/abs/1705.03387
[49] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting classifiers against adversarial attacks using generative models," 2018, *arXiv:1805.06605*. [Online]. Available: http://arxiv.org/abs/1805.06605
[50] S. Ben-David, J. Blitzer, K. Crammer, and F. Pereira, "Analysis of representations for domain adaptation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 137–144.
[51] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, nos. 1–2, pp. 151–175, May 2010.
[52] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2030–2096, May 2015.
[53] H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, and M. Marchand, "Domain-adversarial neural networks," 2014, *arXiv:1412.4446*. [Online]. Available: http://arxiv.org/abs/1412.4446
[54] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
[55] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: http://arxiv.org/abs/1701.07875
[56] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5767–5777.
[57] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc., B (Methodol.)*, vol. 39, no. 1, pp. 1–22, 1977.
[58] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*. [Online]. Available: http://arxiv.org/abs/1312.6114
[59] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.
[60] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[61] G. H. Golub, P. C. Hansen, and D. P. O'Leary, "Tikhonov regularization and total least squares," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 1, pp. 185–194, 1999.
[62] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," 2019, *arXiv:1903.12261*. [Online]. Available: http://arxiv.org/abs/1903.12261
[63] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, "Visualizing the loss landscape of neural nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 6389–6399.
[64] A. Coates, A. Ng, and H. Lee, "An analysis of single-layer networks in unsupervised feature learning," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, 2011, pp. 215–223.
[65] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980
[66] T. Tieleman and G. Hinton, "Divide the gradient by a running average of its recent magnitude," COURSERA: Neural Netw. Mach. Learn., Mountain View, CA, USA, Tech. Rep. Lecture 6.5-rmsprop, 2017.
[67] L. Luo, Y. Xiong, Y. Liu, and X. Sun, "Adaptive gradient methods with dynamic bound of learning rate," 2019, *arXiv:1902.09843*. [Online]. Available: http://arxiv.org/abs/1902.09843
[68] N. Shirish Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. Tak Peter Tang, "On large-batch training for deep learning: Generalization gap and sharp minima," 2016, *arXiv:1609.04836*. [Online]. Available: http://arxiv.org/abs/1609.04836
[69] S. Hochreiter and J. Schmidhuber, "Flat minima," *Neural Comput.*, vol. 9, no. 1, pp. 1–42, 1997.
[70] I. J. Goodfellow, O. Vinyals, and A. M. Saxe, "Qualitatively characterizing neural network optimization problems," 2014, *arXiv:1412.6544*. [Online]. Available: http://arxiv.org/abs/1412.6544
[71] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio, "Sharp minima can generalize for deep nets," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 1019–1028.
[72] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," 2017, *arXiv:1711.00117*. [Online]. Available: http://arxiv.org/abs/1711.00117
[73] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.



SUBHAJIT CHAUDHURY (Member, IEEE) received the B.E. degree in electrical engineering from Jadavpur University, Kolkata, India, in 2012, the M.Tech. degree in electrical engineering from the Indian Institute of Technology Bombay, in 2014, and the Ph.D. degree in information and communication engineering from The University of Tokyo, in 2021. From September 2014 to March 2017, he worked as a Researcher at the NEC Research Laboratories, Japan. Since April 2017, he has been working as a Research Scientist at IBM Research, Tokyo. His current research interests include adversarial attacks, reinforcement learning, and computer vision.

HIYA ROY (Member, IEEE) received the B.E. degree in electrical engineering from Jadavpur University, Kolkata, India, in 2012, and the M.S. and Ph.D. degrees in electrical engineering and information systems from The University of Tokyo, in 2017 and 2021, respectively. From July 2012 to July 2015, she worked as a Lead Engineer at Tata Power Company Ltd., India. She worked as an Intern at the NASA Jet Propulsion Laboratory, California Institute of Technology, and the NEC Central Research Laboratories, Japan. From September 2015 to August 2020, she was an MEXT Scholar. She is currently working as a Research Scientist at Rakuten Institute of Technology. Her research interests include computer vision, machine learning, planetary sciences, and creative AI.

SOURAV MISHRA (Student Member, IEEE) received the Ph.D. degree in information and communication engineering from The University of Tokyo, in 2021. Before that, he was a Graduate Student in electrical and computer engineering at Virginia Tech, where he worked on designing the next generation quantitative imaging systems in collaboration with Carl Zeiss and Siemens. He has worked with Microsoft Research (Beijing) on machine learning solutions in emerging technological domains. He has been a Student Member of IEEE at Virginia Tech. He is currently with The University of Tokyo. He is working as a Data Scientist at AnyMind, Tokyo.

TOSHIHIKO YAMASAKI (Member, IEEE) received the B.S. degree in electronic engineering, the M.S. degree in information and communication engineering, and the Ph.D. degree from The University of Tokyo, in 1999, 2001, and 2004, respectively. From April 2004 to October 2006, he was an Assistant Professor with the Department of Frontier Informatics, Graduate School of Frontier Sciences, The University of Tokyo, where he is currently an Associate Professor with the Department of Information and Communication Engineering, Graduate School of Information Science and Technology. He was a JSPS Fellow for Research Abroad and a Visiting Scientist at Cornell University, from February 2011 to February 2013. His current research interests include attractiveness computing based on multimedia big data analysis, pattern recognition, and machine learning. He is a member of ACM, AAAI, IEICE, ITE, and IPSJ.