Journal of Ambient Intelligence and Humanized Computing (2024) 15:3229–3242
https://doi.org/10.1007/s12652-024-04791-1

ORIGINAL RESEARCH

Sliced Wasserstein adversarial training for improving adversarial robustness

Woojin Lee[1] · Sungyoon Lee[2] · Hoki Kim[3] · Jaewook Lee[4]

Received: 11 January 2022 / Accepted: 13 March 2024 / Published online: 5 May 2024

Abstract
Recently, deep-learning-based models have achieved impressive performance on tasks that were previously considered to be extremely challenging. However, recent works have shown that various deep learning models are susceptible to adversarial data samples. In this paper, we propose the sliced Wasserstein adversarial training method to encourage the logit distributions of clean and adversarial data to be similar to each other. We capture the dissimilarity between two distributions using the Wasserstein metric and then align distributions using an end-to-end training process. We present the theoretical background of the motivation for our study by providing generalization error bounds for adversarial data samples. We performed experiments on three standard datasets and the results demonstrate that our method is more robust against white box attacks compared to previous methods.

Keywords Adversarial attack · Adversarial training · Sliced Wasserstein Distance · Adversarial defense

1 Introduction

Deep learning models have made significant strides in a variety of fields. Yet, their sensitivity to subtle perturbations has been exposed by the presence of adversarial examples, which typically remain undetected by human observation. (Szegedy et al. 2013; Shaham et al. 2018; Li et al. 2018; Kim et al. 2023b). Adversarial examples, generated by introducing subtle distortions to original inputs, can significantly alter the output of deep learning models. These examples differ minimally from the original images, yet these small variances are amplified in the results of deep learning models. Since these perturbations are unnoticed by humans, they pose security risks in practical applications of deep learning technologies. Consequently, crafting defensive algorithms to counteract these adversarial attacks is crucial for the safe deployment of deep artificial intelligence systems.

Adversarial samples are generated by solving optimization problems. Since the first appearance of adversarial samples, many attack methods have been proposed, including the fast gradient sign method (FGSM) (Goodfellow et al. 2014), iterative FGSM (Kurakin et al. 2016), deep fool (Moosavi-Dezfooli et al. 2016), Carlini and Wagner (C &W) attack (Carlini and Wagner 2017), and projected gradient descent (PGD) (Madry et al. 2017). Based on their simple formulations, adversarial attacks are feasible in many tasks, including face recognition (Sharif et al. 2016), reinforcement learning (Huang et al. 2017), audio classification (Kim et al. 2023d), object detection (Wang et al. 2020), and medical imaging (Li et al. 2020).

Therefore, many defense mechanisms for handling such adversarial attacks have been proposed. Some mechanisms exploit additional heuristics, such as test-time randomness (Guo et al. 2017; Dhillon et al. 2018), non-differentiable

✉ Jaewook Lee
jaewook@snu.ac.kr

Woojin Lee
wj926@dgu.ac.kr

Sungyoon Lee
sungyoonlee@hanyang.ac.kr

Hoki Kim
hokikim@cau.ac.kr

[1] School of AI Convergence, Dongguk University-Seoul, Seoul, Republic of Korea

[2] Department of Computer Science, Hanyang University, Seoul, Republic of Korea

[3] Department of Industrial Security, Chung-Ang University, Seoul, Republic of Korea

[4] Department of Industrial Engineering, Seoul National University, Seoul, Republic of Korea

preprocessors (Xie et al. 2017; Samangouei et al. 2018), or detection of attacks (Martin and Elster 2020). Although these additional heuristics can defeat simple optimization-based attacks, recent studies have shown that such defenses can be easily defeated by stronger adversaries (Athalye et al. 2018). Recently, there have been works related to the smoothness of the deep learning models (Kim et al. 2023c; Lee et al. 2021a; Kim et al. 2023b; Stutz et al. 2021)

Another widely used approach is adversarial training (Goodfellow et al. 2014; Madry et al. 2017; Liu and Chan 2022), where adversarial samples generated intentionally during training are used as training inputs. Adversarial training is easy to implement and has not yet been completely defeated. However, adversarial training requires a specific attack algorithm (e.g., FGSM) to generate adversarial training samples and may exhibit weak generalization ability for other adversarial samples. Despite the passage of years since the proposal of PGD-based adversarial training, as referenced in Madry et al. (2017), Croce et al. (2020), it remains the leading method of defense, albeit with less than optimal performance.

Recently, many studies have attempted to improve the performance of adversarial training by introducing additional regularizers, such as the L_2 loss between logits for a pair of clean and adversarial examples (Kannan et al. 2018), rectified linear unit (ReLU) stability regularizers (Xiao et al. 2018), and domain adaptation loss (Song et al. 2018).

In this paper, we consider the problem of adversarial attacks from the perspective of domain adaptation. Domain adaptation is an aspect of transfer learning that attempts to train a model using labeled source domain data that performs well on a given set of target data. It assumes that two domains are defined for the same task, but with different distributions. Because domain adaptation handles the problem of two domains with different distributions, it is closely related to adversarial robustness. Even though adversarial noise is typically imperceptible to humans, the distributions of adversarial samples in a high-level representation space differ significantly from those of original images (Fig. 1). To construct a model robust against adversarial attacks, it is important to handle distribution distances in a high-level representation space.

Domain adaptation attempts to resolve the issue of different distributions between domains by using transferable representations, which cannot be distinguished by the representations of the source and target domains. By learning a representation that reduces the distance between two different domains, domain adaptation can construct a model that can be applied to two different domains. There are various approaches to minimizing the distance between two domains, such as maximum mean discrepancy (Long et al. 2017), $\mathcal{H}$ -divergence (Ganin et al. 2016), KL divergence (Lee et al. 2021b), Wasserstein distance (Yoon et al. 2020)and Jensen Shannon divergence (Tzeng et al. 2017).

Inspired by the domain adaptation approach, we aimed to construct a model that can reduce the differences between distributions in a high-level representation space for original and adversarial images. As shown in Fig. 1, designing a classifier that can reduce the differences between logit distributions can suppress the influence of adversarial perturbations, leading to a model robust against adversarial attacks.

In this paper, we propose the sliced Wasserstein adversarial training (SWAT) method to design a classifier that provides consistent performance on clean and adversarial samples. We make the output logit distributions of clean and adversarial samples more similar by minimizing the Wasserstein metric (Redko et al. 2017; Frogner et al. 2015), which is a meaningful notion of dissimilarity between probability distributions. Although calculating Wasserstein distance can be computationally expensive, our approach based on sliced Wasserstein distance (SWD) uses a simple numerical solution to handle this problem. Recently, several

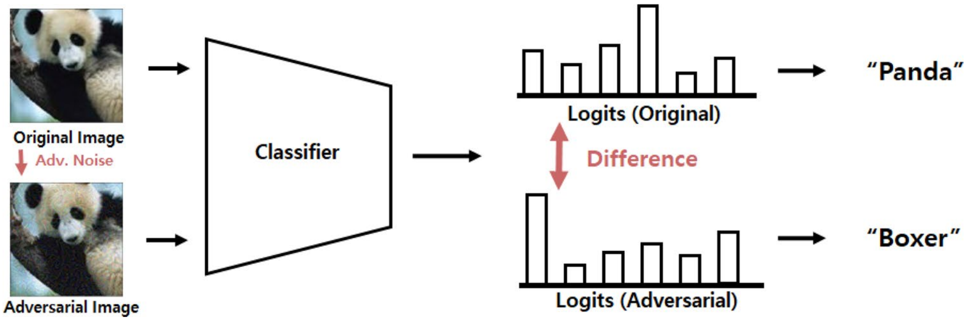


Fig. 1 Illustration of the differences between an adversarial image and an original image. Adversarial perturbations are so small that they are often imperceptible to humans. However, adversarial noise is amplified through the layers of a network, which maximizes the distance between an original sample and an adversarial sample in high-level representations (logits). As a result, the network incorrectly classifies the adversarial image "Panda" as "Boxer". We reduce the differences between the distributions of high-level representations to construct a model that is robust against adversarial attacks

studies have used SWD in various applications (Wu et al. 2019; Lee et al. 2019; Kolouri et al. 2018; Kim et al. 2023a). We also present new generalization bounds for adversarial samples that illustrate the need to reduce the Wasserstein distances between the logit distributions of clean and adversarial samples during adversarial training. The main contributions of this paper can be summarized as follows.

1. First, we propose a novel approach to aligning the output probability distributions of clean and adversarial data using the Wasserstein metric. We also present the SWAT method, which is a computationally efficient end-to-end network training method using SWD.
2. Second, we present the theoretical background motivating the SWAT method by providing generalization upper bounds for adversarial samples.
3. Third, we present empirical evaluations that demonstrate the robustness and accuracy of our method under various white box attacks.

2 Related work

2.1 Adversarial attack methods

Szegedy et al. (2013) demonstrated that small perturbations in original images can easily fool neural network models. In a follow-up paper (Goodfellow et al. 2014), a novel attack method called FGSM was proposed, which significantly reduced the computational time required to generate adversarial images using simple one-step back-propagation.

$$x_{adv} = x + \epsilon \text{sign}(\nabla_x \mathcal{L}(\theta, x, y)) \tag{1}$$

The symbols x, y, θ and $\mathcal{L}$ represent an input image, input label, network weights, and loss function, respectively. Using the above algorithm, one can obtain adversarial images that are denoted as x_{adv} within the ϵ-norm area surrounding x.

One of the strongest types of adversarial attacks is PGD (Madry et al. 2017), which projects adversarial examples with a step size α onto a set of allowable perturbations S in every iteration. This attack often reduces the accuracy of normal models to nearly zero.

$$x^{t+1} = \Pi_{\mathcal{B}(x,\epsilon)}(x^t + \alpha \text{sign}(\nabla_x \mathcal{L}(\theta, x, y))), \tag{2}$$

where $\Pi_{\mathcal{B}(x,\epsilon)}$ refers the projection to the ϵ-ball $\mathcal{B}(x, \epsilon)$.

2.2 Adversarial training

Various defense methods have been proposed to preserve the stability of deep learning models under the types of attacks described above. The most widely used defense method is adversarial training, which simply includes adversarial examples when training a model. The two most popular adversarial training methods use FGSM (Goodfellow et al. 2014), and PGD (Madry et al. 2017), respectively. The first method uses FGSM because it can generate adversarial samples quickly (Goodfellow et al. 2014). The second method formulates the empirical adversarial risk minimization problem as the following minimax problem (Madry et al. 2017):

$$\min_{\theta} \mathbb{E}_{(x,y)\sim\mathcal{D}}[\max_{\delta\in\mathcal{B}(x,\epsilon)} \mathcal{L}(\theta, x + \delta, y)] \tag{3}$$

The inner maximization is approximated by a PGD attack with random restarts. In many previous studies, this method was determined to be effective, but it cannot defend against all adversarial samples. Above all, because adversarial training uses specific attack methods, the choice of which attack method to use is very important. We also attempt to use clean samples in addition to PGD adversarial training, as recommended in Kurakin et al. (2016). However, since PGD based adversarial training requires multiple steps of gradients, it suffers from computational burden.

Recently, many studies have focused on improving robustness (Madry et al. 2017; Ye et al. 2020; Drewek-Ossowicka et al. 2021; Cao et al. 2019). One such study resulted in a method called adversarial training domain adaptation (ATDA) (Song et al. 2018). The main concept of this method is to use an FGSM adversary as a target domain. Additionally, the authors exploit three types of loss to construct logit vectors from original images $\phi(x)$ and adversarial images $\phi(x_{adv})$. The three types of loss are covariance distance, maximum mean discrepancy (MMD) of mean vectors, and supervised domain adaptation loss, which consists of the intra-class variations and inter-class similarities of $\phi(x)$ and $\phi(x_{adv})$. Adversarial training attempts to assign clean and corresponding adversarial samples to the same class, but Kannan et al. (2018) proposed a method called adversarial logit pairing (ALP), which encourages logits from two images to be similar. Moreover, there have been approaches that tried to improve the adversarial training by redundant batches and cumulative perturbations (Shafahi et al. 2019), or uniform random initialization (Wong et al. 2020), or by avoiding catastrophic overfitting problems in single-step adversarial training (Kim et al. 2021). There has also been a trend of analyzing the smoothness of adversarial attacks (Lee et al. 2021a; Kim et al. 2023c; Liu and Chan 2022). However, there has not been a significant improvement in the performance of defense mechanisms yet.

To extend these adversarial-training-based approaches, we propose a novel design for distribution-matching adversarial training. The method in Song et al. (2018) requires calculating the covariance distance and MMD of each data pair and the optimization of three different complicated loss functions. In contrast, our approach is computationally efficient and easily converges. Both the method in Kannan et al. (2018) and our method attempt to minimize the distance between between two logits, but our method provides a tighter error bound.

3 Wasserstein distance in robust training

3.1 Notations

We consider classification tasks in which $\mathcal{X}$ is an input space and $\mathcal{Y} = \{0, \dots, c-1\}$ is an output space. Given a hypothesis set $\mathcal{H} = \{h : \mathcal{X} \to \mathbb{R}^{|\mathcal{Y}|}\}$, we define a classification network $Q_\theta \in \mathcal{H}$ with parameters θ that outputs *logits* $Q_\theta(x)$. We denote $\mathcal{D}_S = \langle \mathcal{D}_S^{\mathcal{X}}, c^* \rangle$ and $\mathcal{D}_A = \langle \mathcal{D}_A^{\mathcal{X}}, c^* \rangle$ be the clean source and adversarial domains with the true concept (labeling function) $c^* : \mathcal{X} \to \mathcal{Y}$, a clean source distribution as $x \sim \mathcal{D}_S^{\mathcal{X}}$ and adversarial distribution $x^{adv} \sim \mathcal{D}_A^{\mathcal{X}}$.

3.2 Wasserstein distance

For any $p \geq 1$, the p-Wasserstein distance between probability measures μ and ν where $\mu, \nu \in \{\mu : \int d(x,y)^p d\mu \leq \infty, \forall y \in \mathcal{Z}\}$, is the p-th root of

$$W_p(\mu, \nu)^p = \inf_{\pi \in \Pi(\mu,\nu)} \mathbb{E}_{(x,y)\sim\pi}[d(x,y)^p], \tag{4}$$

where $\Pi(\mu, \nu)$ is the set of all joint distribution whose marginals are μ and ν. According to the Kantorovich duality theorem, the 1-Wasserstein distance can be simplified as

$$W_1(\mu, \nu) = \sup_{f \in \text{Lip1}} \mathbb{E}_{z\sim\mu}[f(z)] - \mathbb{E}_{z\sim\nu}[f(z)] \tag{5}$$

where Lip1 is the set of real-valued 1-Lipschitz continuous functions on $\mathcal{Z}$, i.e. $\text{Lip1} \equiv \{f : \mathcal{Z} \to \mathbb{R} : |f(x) - f(y)| \leq d(x,y), \forall x, y \in \mathcal{Z}\}$.

We propose to minimize the Wasserstein distance between two logit distributions $Q_\theta \# \mathcal{D}_S^{\mathcal{X}}$ and $Q_\theta \# \mathcal{D}_A^{\mathcal{X}}$, to build a robust model, respectively. We use the push-forward notation # for transferring measures $\mathcal{D}_S^{\mathcal{X}}$ and $\mathcal{D}_A^{\mathcal{X}}$ on input space $\mathcal{X}$ toward logit space $\mathcal{Z}$ by using parametrized network Q_θ. Then the Wasserstein distance between two logit distributions can be written as

$$\begin{aligned} &W_1(Q_\theta \# \mathcal{D}_S^{\mathcal{X}}, Q_\theta \# \mathcal{D}_A^{\mathcal{X}}) \\ &\quad = \sup_{f \in \text{Lip1}} \mathbb{E}_{z\sim Q_\theta \# \mathcal{D}_S^{\mathcal{X}}}[f(z)] - \mathbb{E}_{z\sim Q_\theta \# \mathcal{D}_A^{\mathcal{X}}}[f(z)] \\ &\quad = \sup_{f \in \text{Lip1}} \mathbb{E}_{x\sim \mathcal{D}_S^{\mathcal{X}}}[f(Q_\theta(x))] - \mathbb{E}_{x\sim \mathcal{D}_A^{\mathcal{X}}}[f(Q_\theta(x))] \end{aligned} \tag{6}$$

Wasserstein distance is weaker than many other distance metrics between probability distributions, such as Jensen–Shannon divergence and total variation distance. Furthermore, convergence with respect to the topology induced by Wasserstein distance is equivalent to convergence in a distribution. Therefore, it is not only an appropriate metric for the distribution space, but also has better convergence properties, particularly for distributions with low-dimensional supports (Arjovsky and Bottou 2017).

3.3 Upper bound on robust training

In this section, we present an upper bound on objective of robust training. Adversarial risk of a hypothesis $h \in \mathcal{H}$ in a domain $\mathcal{D}_S = \langle \mathcal{D}_S^{\mathcal{X}}, c^* \rangle$ is defined as follows:

$$\mathcal{R}_{robust}(h;\mathcal{D}_S) = \mathbb{E}_{(x,y)\sim\mathcal{D}_S}[\max_{x' \in \mathbb{B}(x)} l(h(x'), y)]. \tag{7}$$

We implicitly use y as a label of the input x, i.e., $y = c^*(x)$. The goal of robust training is to minimize the worst misclassification rate on the data domain $\mathcal{D}_S$, using (7) with the $0-1$ loss, i.e., $l(\hat{y}, y) = \mathbf{1}\{\arg\max_i \hat{y}_i \neq y\}$. However, in training phase, we instead use the cross-entropy loss as a surrogate since the $0-1$ loss is intractable (Hoffgen et al. 1995). Recent work on over-parameterized neural networks (Allen-Zhu et al. 2019) have shown that the loss function $l_y \circ h \equiv l(h(\cdot), y)$ is Lipschitz-smooth for all y, i.e., $|l_y \circ h(x) - l_y \circ h(x') - \nabla_x l_y \circ h(x)^T (x' - x)| \leq \frac{1}{2} L \|x' - x\|_2^2, \forall x', x \in \mathcal{X}$ for some constants L, and called $l_y \circ h$ is L-smooth. The following theorem provides a new upper bound based on the combination of clean source data and the first-order adversarial data.

Theorem 1 *Given a hypothesis $h \in \mathcal{H} = \{h : \mathcal{X} \to \mathbb{R}^c\}$ satisfying that $l_y \circ h$ is β-smooth. Let $\mathcal{D}_S$ and $\mathcal{D}_{A|h}$ be the clean source and the first-order adversarial domains with respect to the hypothesis h, respectively, and ϵ be an adversarial perturbation, the following inequality holds*:

$$\begin{aligned} &\mathcal{R}_{robust}(h;\mathcal{D}_S) \\ &\quad \leq \frac{1}{2}(\mathcal{R}_S(h) + \mathcal{R}_{A|h}(h) \\ &\qquad + \sqrt{\frac{c}{c-1}} W_1(h\#\mathcal{D}_{A|h}^{\mathcal{X}}, h\#\mathcal{D}_S^{\mathcal{X}}) + \beta \|\epsilon\|_2^2), \end{aligned} \tag{8}$$

where $\mathcal{R}_S(h) \equiv \mathbb{E}_{\mathcal{D}_S}[l(h(x), y)]$ *and* $\mathcal{R}_{A|h}(h) \equiv \mathbb{E}_{(\tilde{x},y)\sim\mathcal{D}_{A|h}}[l(h(\tilde{x}), y)]$.

As a result, the upper bound on the adversarial risk can be decomposed into four parts. The first two terms are derived from the first-order adversarial samples and the source clean samples, respectively. The third term is the Wasserstein distance between logit distributions of the source and adversarial domains. As will be discussed later, our proposed method tries to minimized the terms in the upper bound. To compute the Wasserstein distance in the third term, we use SWD for computational efficiency.

3.4 Advantage of Wasserstein distance

In this section, we provide an analysis of using Wasserstein distance between normal logits and adversarial logits. To minimize the adversarial risk in our upper bound it is required to reduce the Wasserstein distance between logit distributions, which is the third term of the Theorem 1. From the perspective of matching two logit distributions, there were approaches (Kannan et al. 2018; Pang et al. 2020) that applied L_2 distances with the paired logits, such as ALP. However in this paper, our suggested upper bound reduces the optimal transport cost rather than the paired L_2 distance.

The L_2 regularizer minimizes the (expected) difference between a pair of logits z (normal example) and z^* (corresponding adversarial example) as follows:

$$\mathcal{L}_{ALP} = \mathbb{E}_{\mathcal{D}}\left[d(z, z^*)\right] = \int d(z, z^*)dp(z, z^*), \tag{9}$$

The Wasserstein distance regularizer minimizes the optimal transport cost between two distributions of logits, where

$$W_1(\mu_z, \mu_{z^*}) = \inf_{\pi \in \Pi} \mathbb{E}_{\pi}\left[d(z, z^*)\right] = \int d(z, z^*)d\tilde{\pi}(z, z^*) \tag{10}$$

where μ_z and μ_{z^*} are the measure for normal logits and adversarial logits, and $\tilde{\pi}$ is an optimal plan for the transport between μ_z and μ_{z^*}. The difference is which transport plan is used between μ_z and μ_{z^*}. Therefore, $W_1(\mu_z, \mu_{z^*}) \leq \mathcal{L}_{ALP}$ holds, implying that the Wasserstein regularizer has a tighter bound than that of paired L_2 regularizer.

From more intuitive perspective, paired L_2 regularizer tries to match the $Q_\theta(x_i^{adv})$ to the corresponding normal logits $Q_\theta(x_i)$. On the other hand, since our upper bound tries to find the optimal plan between μ_z and μ_{z^*}, it tries to match the samples $Q_\theta(x_i^{adv})$ to the nearest normal logits $Q_\theta(x_j)$.

Comparison between paired L_2 regularizer and our upper bound is illustrated in Fig. 2. The left figure visualizes the training procedure of paired L_2 regularizer while the right figure shows our Wasserstein based upper bound. The black points represent the logits of normal samples while the red points are the logits of adversarial samples. In Fig. 2, we can find that ALP focuses on matching the paired sample z_i and z_i', while our upper bound focuses on matching the global distribution of μ_z and μ_{z^*} by minimizing the optimal transport.

Since our proposed method tries to reduce the optimal transport between μ_z and μ_{z^*}, it can prevent the over-regularization. For example, in Fig. 2, the logits of adversarial sample z_1' can be robust if it can be embedded near the normal logit distribution μ_z. In this case, paired L_2 regularizer tries to reduce the distance between z_1' and z_1 and our proposed method reduces the distance between z_1' and z_2. If the label of z_1 and z_2 is identical, reducing $d(z_1', z_2)$ can be easier to learn a robust embedding, and prevent over-regularization. We provide more analysis on this on real datasets in Sect. 5.2.

4 Proposed method

4.1 Sliced Wasserstein distance

To minimize the upper bound of robust training in Theorem 1, we need to compute the optimal transport between adversarial and normal logits, which is computationally expensive. In this paper, we propose using sliced Wasserstein distance (SWD) to approximate Wasserstein distance between two different distributions. SWD shares similar properties to the original Wasserstein distance, but easier to compute (Kolouri et al. 2018). It projects the higher-dimensional densities into set of one-dimensional distributions and compares the projected distributions via Wasserstein distance. Since SWD shares the same topology with Wasserstein distance in a compact set (Bonnotte 2013), for example, in the logit image space $h(\mathcal{X})$ for a bounded domain $\mathcal{X} = [0, 1]^n$, we used SWD to empirically calculate the Wasserstein distance in Theorem 1.

The sliced Wasserstein distance between μ and ν can be defined as follows:

$$\mathrm{SWD}(\mu, \nu) \equiv \int_{\Omega} W(\mu^w, \nu^w)d\mu_{\Omega}(w), \tag{11}$$

where μ_Ω is a uniform measure on the unit sphere Ω such that $\int_\Omega d\mu_\Omega(w) = 1$, and the measures $\mu^w = w^T\mu$ and $\mu^w = w^T\mu$ are one-dimensional projections of the measure μ and ν onto

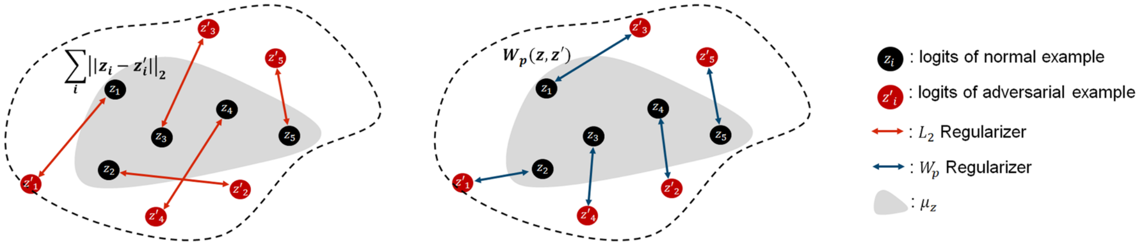


Fig. 2 Illustration of paired L_2 regularizer and our proposed upper bound (W_p). Black points represent logits of normal samples while the red points represent the logits of adversarial samples. Left: effect of paired L_2 regularizer, Right: effect of our upper bound (color figure online)

the direction $w \in \Omega$. Then we extend the definition to finite sets $\mathcal{S}$ and $\mathcal{T}$ as $\mathrm{SWD}(\mathcal{S},\mathcal{T}) \equiv \mathrm{SWD}(\mu_{\mathcal{S}},\mu_{\mathcal{T}})$ where $\mu_{\mathcal{S}} = \frac{1}{|\mathcal{S}|}\sum_{s\in\mathcal{S}}\delta_s$ and $\mu_{\mathcal{T}} = \frac{1}{|\mathcal{T}|}\sum_{t\in\mathcal{T}}\delta_t$ with the Dirac measure δ_x centered on a point x.

The integration (11) for the finite set $\mathcal{S},\mathcal{T}\in\mathbb{R}^p$ with the same cardinality $|\mathcal{S}| = |\mathcal{T}| = n$ can be approximated as follows:

$$\begin{aligned}\mathrm{SWD}(\mathcal{S},\mathcal{T}) = \mathrm{SWD}(\mu_{\mathcal{S}},\mu_{\mathcal{T}}) &\approx \frac{1}{|\hat{\Omega}|}\sum_{w\in\hat{\Omega}} W(\mu^w_{\mathcal{S}},\mu^w_{\mathcal{T}}) \\ &= \frac{1}{|\hat{\Omega}|}\sum_{w\in\hat{\Omega}}\sum_{i=1}^{n}|w^T s_{i,w} - w^T t_{i,w}|^2,\end{aligned} \tag{12}$$

where $\hat{\Omega} = \{w_j\}$ is a finite set of uniform samples from the $(p-1)$-dimensional unit sphere Ω, s_i, t_i are elements of $\mathcal{S},\mathcal{T}$, respectively, and the $s_{i,w}, t_{i,w}$ are the rearrangement of s_i, t_i such that $w^T s_{i,w} \le w^T s_{i',w}$ and $w^T t_{i,w} \le w^T t_{i',w}$ for all $i \le i'$ and $w \in \hat{\Omega}$.

Unlike the original Wasserstein distance $W(\mu_{\mathcal{S}},\mu_{\mathcal{T}})$ between high-dimensional datasets $\mathcal{S}$ and $\mathcal{T}$, SWD uses one-dimensional linear projections $\mu^w_{\mathcal{S}}$ and $\mu^w_{\mathcal{T}}$ to measure distance. Because the computation of one-dimensional Wasserstein distance only requires sorting and computing the absolute distances between sorted pairs, SWD has a significantly lower computational cost than original Wasserstein distance and it enables end-to-end learning using a single deep learning classifier network. In our experiments, we used a number of projection samples $|\hat{\Omega}| = 10$.

Therefore, since SWD has the same topology with Wasserstein distance in a compact set, we can provide a new upper bound using SWD. Using the inequality in Theorem 5.1.5 of Bonnotte (2013), the upper bound on the objective of robust training using SWD becomes the following corollary.

Corollary 4.1 *Under the same condition with Theorem 1, for a constant C_c the following inequality holds:*

$$\begin{aligned}&\mathcal{R}_{robust}(h;\mathcal{D}_S) \\ &\quad\le \frac{1}{2}\Bigg(\mathcal{R}_S(h) + \mathcal{R}_{A|h}(h) \\ &\qquad + C_c\sqrt{\frac{c}{c-1}}\mathrm{SWD}(h\#\mathcal{D}^{\mathcal{X}}_{A|h}, h\#\mathcal{D}^{\mathcal{X}}_{S})^{1/(c+1)} + \beta\|\epsilon\|_2^2\Bigg).\end{aligned} \tag{13}$$

Recently there have been concerns about SWD since it might not approximate the true Wasserstein distance as the dimension increases. However, since we have matched the distribution between two logit distributions, which is not high-dimensional, SWD could successfully approximate the true distribution. In Sect. 5.4, we provide more details related to the approximation.

4.2 Sliced Wasserstein adversarial training (SWAT)

In this section, we introduce how we trained the suggested model empirically in real dataset. At the beginning of training, we sample a mini batch of data $B = \{x_i, y_i\}_{i=1}^m, (B^X = \{x_i\}_{i=1}^m)$ and from a clean dataset where m is the size of the batch and define the corresponding set as $\{X\}$. Using an adversarial attack, we generate adversarial data $B_{adv} = \{x_i^{adv}, y_i\}_{i=1}^m, (B^X_{adv} = \{x_i\}_{i=1}^m)$ in each epoch. In this paper, we used FGSM method to generate adversarial samples.

Initially, we apply the supervised loss function for both clean data $\{x_i, y_i\}$ and adversarial data $\{x_i^{adv}, y_i\}$ for classifier Q_θ, and define the loss function as

$$\mathcal{L}_S = \frac{1}{m}\sum_{i=1}^{m} l(Q_\theta(x_i), y_i)$$

$$\mathcal{L}_A = \frac{1}{m}\sum_{i=1}^{m} l(Q_\theta(x_i^{adv}), y_i).$$

Next, we attempt to minimize the Wasserstein distance between probability distributions of the logit $Q_\theta(B^X)$ and $Q_\theta(B^X_{adv})$ in order to design a classifier that can perform consistently on both adversarial data and clean data. We formulate the loss function using SWD as follows:

$$\begin{aligned}\mathcal{L}_{SWD} &= \mathrm{SWD}(\mu_{B^X}, \mu_{B^X_{adv}}) \\ &= \frac{1}{|\hat{\Omega}|}\sum_{w\in\hat{\Omega}}\sum_{i=1}^{n}|w^T Q_\theta(x_{i,w}) - w^T Q_\theta(x^{adv}_{i,w})|^2,\end{aligned} \tag{14}$$

During the optimization phase, we combine the adversarial training loss function and SWD loss function as follows:

$$\mathcal{L}_{total} = \mathcal{L}_S + \mathcal{L}_{adv} + \lambda\mathcal{L}_{SWD} \tag{15}$$

where λ is a hyperparameter for balancing the regularization term. The first and second term in Eq. (15) is supervised loss function on clean dataset and adversarial dataset that is corresponding to the first two terms of Theorem 1. We optimized the classifier Q_θ by minimizing the loss function $\mathcal{L}_S$ and $\mathcal{L}_A$ in the single batches B and B_{adv} iteratively. The third term of equation (15) is the sliced Wasserstein distance between logit distributions of clean and adversarial datasets that is related to third term in our Theorem 1. We summarize our framework in Algorithm 1 and illustrate its overall architecture in Fig. 3.

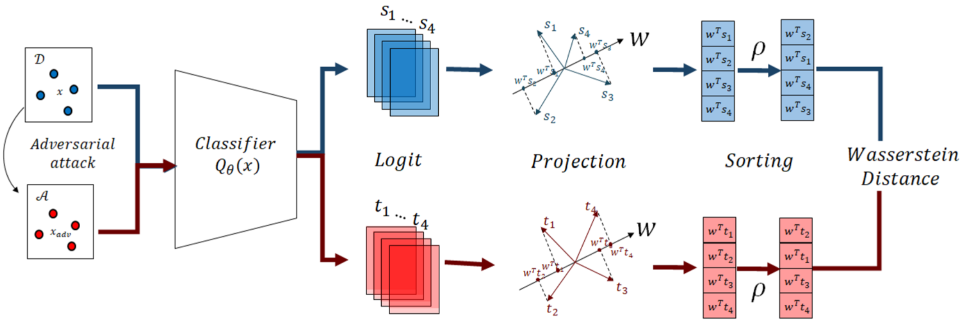


Fig. 3 Illustration of the architecture of our proposed method. Our method is designed to reduce the SWD between two logits $Q_\theta(x)$ and $Q_\theta(x^{adv})$. By using SWD, we can reduce the Wasserstein distance between two measures based on linear projections with uniform measures on a unit sphere to perform end-to-end training

Algorithm 1 SWAT(Sliced Wasserstein Adversarial Training)

```
Require: Labeled dataset $\mathcal{D}$, an adversarial attack method FGSM($\epsilon, \alpha$), and a randomly initialized classifier $Q_\theta$
Ensure: Trained classifier $Q_\theta$.
  for Mini-batch $B$ where $|B| = m$ do
    Generate adversarial dataset $B_{adv}$ with FGSM
    Calculate adversarial training loss
      $\mathcal{L}_S = \sum_{(x_i,y_i)\in B} l(x_i, y_i)$
      $\mathcal{L}_{adv} = \sum_{(x_i^{adv},y_i)\in B_{adv}} l(x_i^{adv}, y_i)$
    Calculate sliced wasserstein distance loss
      Draw a sample vector set $\hat{\Omega} = \{w_k\}$ from unit sphere $\Omega$
      Project $Q_\theta(B^X)$ and $Q_\theta(B^X_{adv})$ on $w$, then rearrange them with a sorting function $\rho$ respectively
        $Q_\theta(B^X)^w = \rho(w^T Q_\theta(B^X))$
        $Q_\theta(B^X_{adv})^w = \rho(w^T Q_\theta(B^X_{adv}))$
      $\mathcal{L}_{SWD} = \frac{1}{|\hat{\Omega}|} \sum_{w\in\hat{\Omega}} |Q_\theta(x_{1\dots m})^w - Q_\theta(x^{adv}_{1\dots m})^w|^2$
    Calculate total loss
      $\mathcal{L}_{total} = \mathcal{L}_S + \mathcal{L}_{adv} + \lambda\mathcal{L}_{SWD}$
    Update $\theta$ with stochastic gradient descent
      $\theta \leftarrow \theta - \nabla_\theta \mathcal{L}_{total}$
  end for
```

5 Experiments

5.1 Dataset and model architecture

In this section, we evaluate our method using three standard classification benchmark datasets. The CIFAR-10 dataset (Krizhevsky and Hinton 2009) consists of 50,000 training images and 10,000 testing images. The size of each image is 32 × 32 × 3 and the dataset contains 10 classes. SVHN (Netzer et al. 2011), which was obtained from house numbers in Google Street View images, is a digit classification dataset with an image dimension of 30 × 30 × 3. It contains 73,257 images for training and 26,032 images for testing with 10 classes (one class for each digit). Fashion-MNIST (Xiao et al. 2017) contains 28 × 28 grayscale images with 10 label classes, where each class denotes one fashion item category, such as “t-shirt” or “sneakers.” This dataset contains 60,000 images for training and 10,000 images for testing. We have summarized the structure of our deep learning structure in Table 1. For each dataset, we constructed different architectures to ease comparisons to other state-of-the-art methods as follows:

5.2 Comparison methods

We compared our method to the following six baseline methods. (1) *Normal*: Basic model that uses only clean training data with a classification loss function. (2) *AT (PGD)*: adversarial training using PGD adversarial samples (Madry et al. 2017). (3) *ATDA*: ATDA training (Song et al. 2018), with a regularization hyperparameter of $\frac{1}{3}$. (4) *ALP*: ALP training (Kannan et al. 2018) with the same logit pairing weight of 0.5 for all data. (5) *Free*: Free single step adversarial training (Shafahi et al. 2019). (6) *Fast*: Fast adversarial training (Wong et al. 2020) (7) *SSAT*: Single step adversarial training (Kim et al. 2021). (8) *Ours*: the proposed method using sliced Wasserstein distance. (9) *Ours**: Our proposed model with additional labeling information.

To push further, we also consider an additional modified version of the proposed method, Ours*. In Fig. 6, we have shown the projected normal logits $Q_\theta(x_i)^w$ in the bottom and the projected adversarial logits $Q_\theta(x_i^{adv})^w$ in the top, where the colors of each points represents the label. The black line links between the paired samples in ALP (left), and the optimal transport in Ours (right). Compared to ALP, which tries to reduce distance even if the corresponding sample is far, our method reduces the distance of the adversarial samples to the nearest normal sample. Moreover, from Fig. 6 we can find that most of the samples in a single batch have been matched with the samples that have the same label. However, in some settings, SWAT might suffer from matching the different labeled samples during training. Therefore, to remove the possibility that the samples can be matched with other samples with a different label, we suggest a variation of our proposed method. Ours * also reduces the sliced Wasserstein distance between μ_z and μ_{z^*}, however when finding the optimal transport plan, Ours* considers label information. Therefore, it always matches the normal sample and adversarial sample that has the same label.

In our proposed method, we use $\lambda = 1$ for the CIFAR 10 dataset and $\lambda = 0.5$ for the SVHN and Fashion-MNIST datasets. For the CIFAR-10 dataset, we generated adversarial images using FGSM with $\epsilon = 8/255, \alpha = \epsilon/4$ in the training phase. For PGD, we used seven iterations with a single random restart. For the SVHN dataset, we used FGSM with $\epsilon = 0.02$, $\alpha = \epsilon/10$ and PGD with 20 iterations and single random restart. Finally, we set $\epsilon = 0.1, \alpha = \epsilon/10$ and used 40 iteration steps with a single random restart for the Fashion-MNIST dataset.

5.3 Results

5.3.1 Classification performance under white box attacks

To evaluate the robustness of our method against adversarial attacks, we measured its classification accuracy under various distortion levels. We evaluated classification performance under three white box attacks: FGSM (Goodfellow et al. 2014), PGD (Madry et al. 2017), Carlini and Wagner (C &W) (Carlini and Wagner 2017) attacks, and EAD attacks (Chen et al. 2018).

FGSM: *CIFAR 10*: Distortion levels ranging from 0 to 10/255 with steps of 2/255. *SVHN*: $\epsilon \in [0, 0.025, 0.005]$. *F-MNIST*: $\epsilon \in [0, 0.25, 0.05]$

PGD: We used the same distortion levels as those used for FGSM for each dataset. *CIFAR 10*: $\alpha = \epsilon/4$ with 20 iteration steps. *SVHN & Fashion-MNIST*: $\alpha = \epsilon/10$ with 20 iteration steps.

C &W: We used constant c values ranging from 10^{-3} to 10^2 on a log scale of 10 for every dataset with 100 optimization steps.

EAD: We used nine binary search steps and run 1000 iterations with initial learning rate 0.01.

The test results are presented in Fig. 4. In this figure, one can see that our method exhibits performance similar to that of the other models (Fig. 4a, d, g) for FGSM attacks. Because every compared method exhibits decent performance under FGSM attacks, which we mainly used as adversarial samples during the training phase, we can assume that all of the models converged during the training phase.

However, in Fig. 4b and e, one can see that our method exhibits the highest robustness against strong PGD attacks. The results of the C &W attacks also demonstrate the robustness of our model against different white box attacks that were not used during the training phase.

Compared to ALP (cyan line in Fig. 4), our method exhibits similar results for FGSM attacks. However, the model performances on PGD and C &W attacks indicate that our method is better at aligning logits in the presence of unknown adversarial attacks. Compared to the PGD training model (light blue line in Fig. 4), our model performs better on the three datasets for strong PGD attacks. In Fig. 4h, the AT (PGD) model performs better than the other methods, but one can see that it fails on C &W attacks in Fig. 4i.

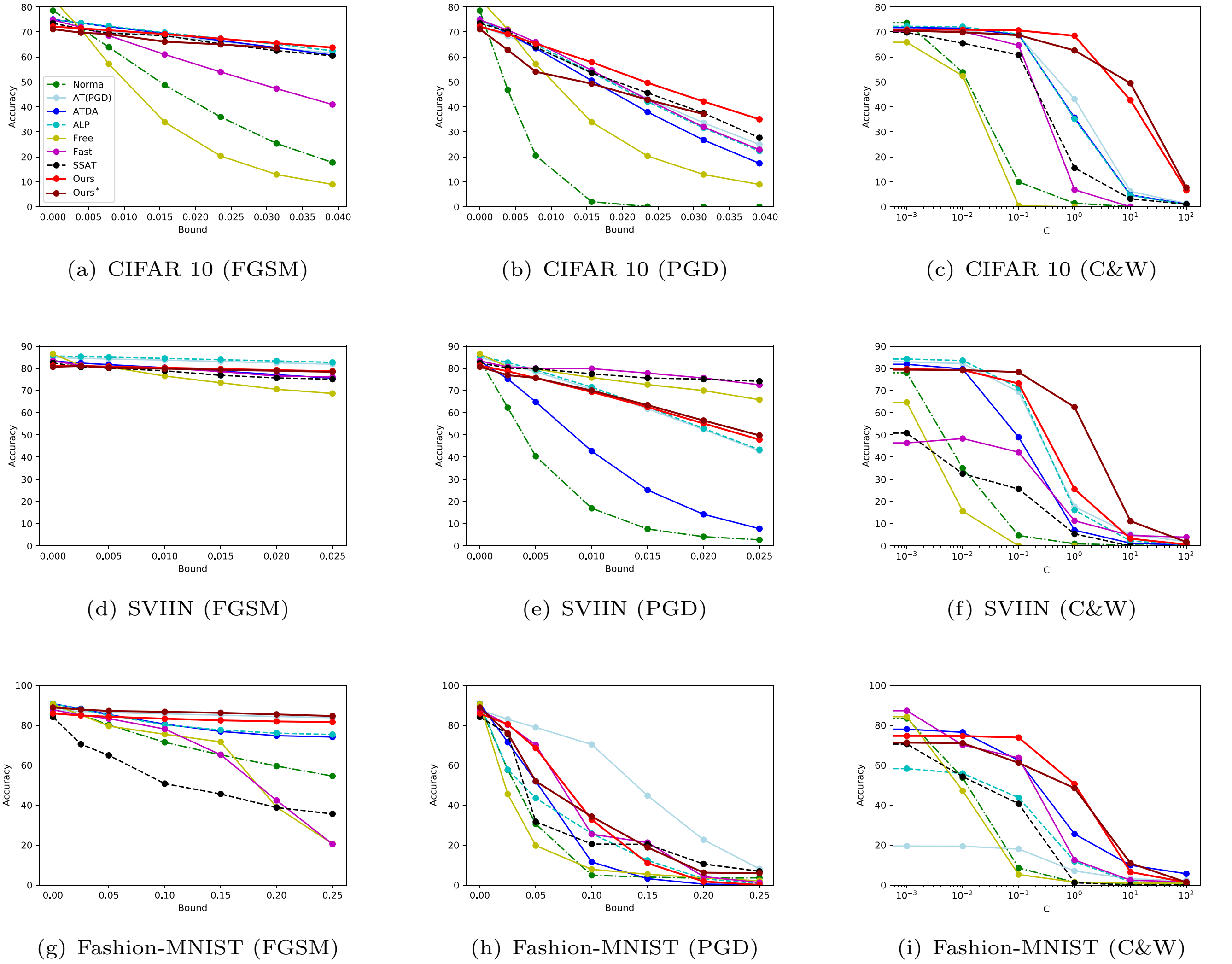
Empirical results demonstrate that while other models fail to construct a generalized defense model for adversarial attacks that were not used in the training phase, our model exhibits consistent results for various types of attacks. One can conclude that our method may exhibit robustness against unknown future adversarial attacks.

We have measured the robustness against EAD attack (Chen et al. 2018). Since the attack success rate of EAD in all three datasets was close to 100%, we have measured the distance between the original image and the EAD attacked adversarial image to evaluate the robustness against EAD attacks. We call these as distortion metrics, and the larger the distortion metric, the more robust the model. In this paper, we have measured the distance in three different metrics (L_1 , L_2, and L_∞). We have summarized the average distortion metrics over successful EAD adversarial examples in Table 2. It is observed that our method has shown the best results in seven out of nine metrics.

5.3.2 Certified radius

While high classification accuracy under white box PGD attacks provides strong empirical evidence that the proposed model is robust against many types of adversarial attacks, it cannot guarantee robustness to norm-bounded attacks. Therefore, it is necessary to compute certified accuracy metrics to determine the effect of SWD regularization on the robustness of the classifier.

We evaluated certified accuracy by computing the certified radius proposed in Cohen et al. (2019). Certified accuracy is defined as the fraction of a test set in which no example is misclassified within the r-neighborhood. We used an induced randomized classifier g with normal noise with a variance of $\sigma = 2.0$ for Fashion-MINST and $\sigma = 0.25$ for the other methods. We selected 100 samples for classification and 105 samples for certified radius estimation for each type of test sample. To reduce computation time, we used 1/100 samples from the testing set to evaluate robustness.



(a) CIFAR 10 (FGSM)

(b) CIFAR 10 (PGD)

(c) CIFAR 10 (C&W)

(d) SVHN (FGSM)

(e) SVHN (PGD)

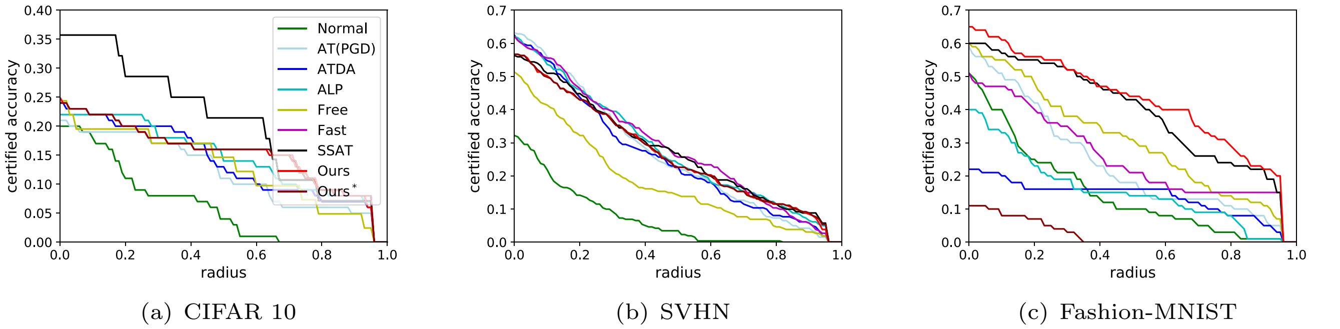
(f) SVHN (C&W)

(g) Fashion-MNIST (FGSM)

(h) Fashion-MNIST (PGD)

(i) Fashion-MNIST (C&W)

Fig. 4 Accuracy of white box attacks (FGSM, PGD, and C &W) in three standard dataset(CIFAR 10, SVHN, and Fashion-MNIST) in test set. X axis refers to perturbation level (ϵ) and Y axis is accuracy (%) (Best viewed in color) (color figure online)



(a) CIFAR 10

(b) SVHN

(c) Fashion-MNIST

Fig. 5 Certified Accuracy in l_2 norm using randomized smoothing (Cohen et al. 2019) . We followed the color scale same as Fig. 4 (Best viewed in color) (color figure online)

Table 1 Architecture of our deep learning model

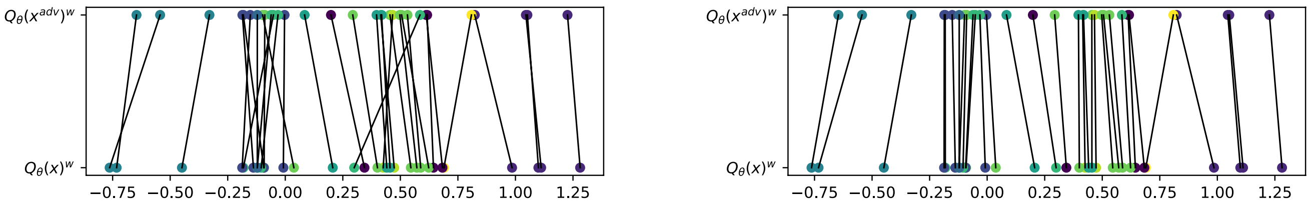
Model	Classifier
IFAR-10	C96-C96-C96-D-C192-C192-C192-D-C192-C192-C100-AP
SVHN, Fashion-MNIST	C16-C32-FC100-FC10

As shown in Fig. 5, SWD regularization (red) improves certified accuracy compared to ALP (cyan) regularization and achieves better results than the other methods.

5.4 Approximation of Wasserstein distance

Recently, there have been concerns with sliced Wasserstein distance. In high dimensional settings, random projection might not capture the properties of the original distribution. In response, there was a research (Kolouri et al. 2019) that suggested generalized sliced Wasserstein distances (GSW) that uses an additional optimization that can better approximate the Wasserstein distance.

In this paper, we have used SWD between two logit distributions where the dimension is 10. Since it is not high-dimensional, using SWD can be an appropriate approach for efficiently computing Wasserstein distance. To show that using SWD is enough for approximating the Wasserstein distance in 10-dimensional distribution, we have calculated

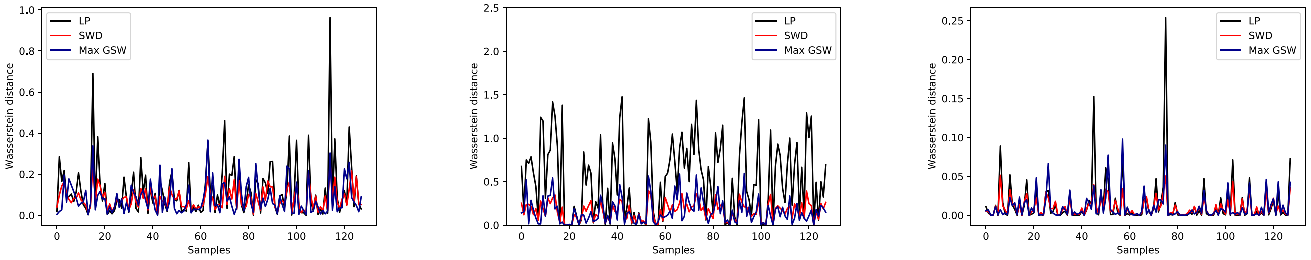


(a) ALP (b) OURS

Fig. 6 Visualization of normal samples and adversarial samples in CIFAR 10 dataset

Table 2 Average distortion metrics over successful adversarial examples generated by EAD attack. The distortion metrics are measured using three different metrics (L_1, L_2 , and L_∞)

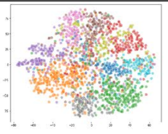
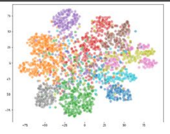
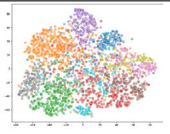
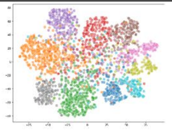
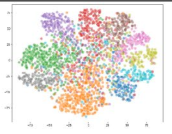
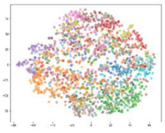
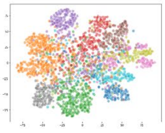
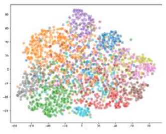
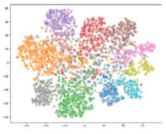
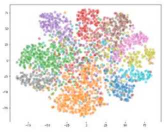
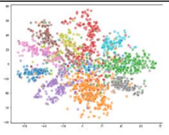
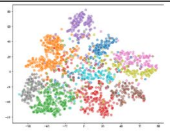
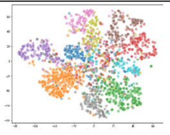
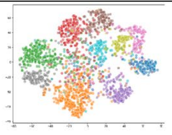
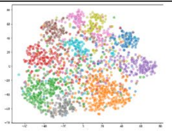
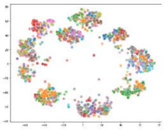
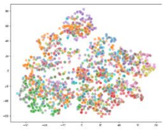
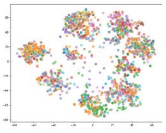
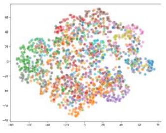
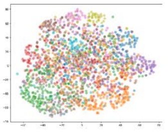
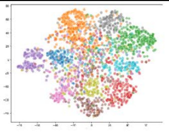
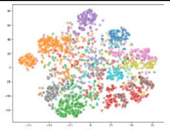
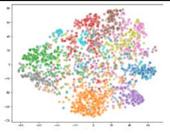
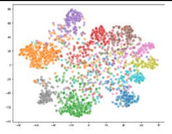
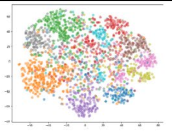
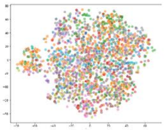
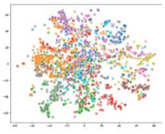
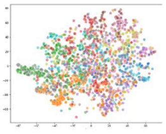
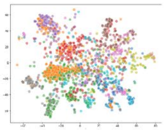
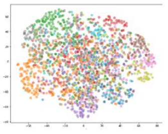
	CIFAR-10			SVHN			Fashion-MNIST		
	L_1	L_2	L_∞	L_1	L_2	L_∞	L_1	L_2	L_∞
Normal	2.4975	0.2809	0.0899	4.3969	0.2930	0.0656	3.5603	0.4560	0.1749
AT(PGD)	7.4457	0.9185	0.2937	4.6486	0.5986	0.2021	6.9916	1.1426	0.5148
ALP	6.9727	0.8704	0.2710	**6.868**	0.4906	0.1124	3.5457	0.8071	0.4244
ATDA	8.5754	0.9178	0.2641	4.8633	0.6181	0.2010	2.9637	0.6737	0.3481
Free	6.3284	0.9134	0.2357	5.6187	0.6135	0.1934	6.9574	0.8745	0.3947
Fast	7.9215	0.8437	0.2986	5.9487	**0.6381**	0.1967	8.6197	0.9134	0.4887
SSAT	8.6157	0.9687	0.2687	4.5138	0.6311	0.1987	5.6137	1.2163	0.1340
Ours	8.9719	0.9785	**0.2997**	4.8726	0.6090	0.2025	6.0525	1.2214	**0.5330**
Ours*	**9.8951**	**0.9932**	0.2744	4.6794	0.5937	**0.2086**	**13.0275**	**1.2674**	0.4587



(a) CIFAR 10 (b) SVHN (c) Fashion-MNIST

Fig. 7 Approximation of Wasserstein distance with SWD and GSW

Table 3 t-SNE visualization for clean data x and white box adversarial data x^{adv} the logit space for SVHN dataset. It shows representation of 2000 clean examples and 2000 corresponding examples where each color represents different labels. We compared with four comparison models (Normal Training, Normal + AT (FGSM), ATDA, ALP) against three different adversarial attacks (FGSM, PGD, C &W)

		Normal	Normal + AT	ATDA	ALP	Ours
FGSM	Clean					
	White					
PGD	Clean					
	White					
CW	Clean					
	White					

the Wasserstein distance using three different methods during training.

In Fig. 7, we provide the actual values of Wasserstein distances that were approximated by using linear OT approach, SWD, and GSW. In this figure, the x-axis denotes distribution samples and the y-axis represents the distance. We can find that SWD and GSW both have successfully approximated the Wasserstein distance (LP). Considering that computing the distance with SWD and GSW takes 0.385 s and 6.276 s respectively, using SWD was appropriate in our settings.

6 Conclusion

In this paper, we proposed a novel defense framework called SWAT that minimizes the Wasserstein distance between the logits of clean and adversarial data samples. We used SWD to design a computationally efficient end-to-end training framework that is robust to adversarial attacks. Empirical results demonstrated that our model is more robust than previous defense models on three standard datasets in terms of four different adversarial attacks and certified accuracy. Our method significantly outperformed previous methods against adversarial attacks that were not used for adversarial training and achieved the highest certified accuracy. Visualizations of the logit spaces of clean and adversarial samples indicated that SWAT successfully aligns output distributions.

Appendix A: Proof of Theorem 1

We first define functions as follows:

$$\begin{aligned} \alpha^*(x, y, h) &= \underset{x' \in \mathbb{B}(x)}{\arg\max}\, l(h(x'), y), \\ \tilde{\alpha}(x, y, h) &= x + \underset{\delta \in \mathbb{B}(0)}{\arg\max}\, \delta^T \nabla_x l(h(x), y), \end{aligned} \tag{16}$$

shortly we call $\alpha^*(x) = \alpha^*(x, y, h)$ and $\tilde{\alpha}(x) = \tilde{\alpha}(x, y, h)$. In the following we rewrite the adversarial risk:

$$\begin{aligned}\mathcal{R}_{robust}(h;\mathcal{D}_S) &= \mathbb{E}_{(x,y)\sim\mathcal{D}_S}[\max_{x'\in\mathbb{B}(x)} l(h(x'),y)] \\ &= \mathbb{E}_{(x,y)\sim\mathcal{D}_S}[l(h(\alpha^*(x)),y)].\end{aligned} \tag{17}$$

Then for the first-order adversary $\tilde{x}$, (17) can be decomposed as follows:

$$\begin{aligned}&\mathbb{E}_{\mathcal{D}_S}\left[l(h(\alpha^*(x)),y)\right] \\ &\quad= \mathbb{E}_{\mathcal{D}_S}\left[l(h(\alpha^*(x)),y) - l(h(\tilde{\alpha}(x)),y)\right] \\ &\qquad+ \mathbb{E}_{\mathcal{D}_S}\left[l(h(\tilde{\alpha}(x)),y)\right] \\ &\qquad- \frac{1}{2}\mathbb{E}_{\mathcal{D}_S}[l(h(x),y)] + \frac{1}{2}\mathbb{E}_{\mathcal{D}_S}\left[l(h(x),y)\right] \\ &\quad= \mathbb{E}_{\mathcal{D}_S}[l_y \circ h(\alpha^*(x)) - l_y \circ h(\tilde{\alpha}(x))] \\ &\qquad+ \frac{1}{2}\mathbb{E}_{\mathcal{D}_S}[l(h(\tilde{\alpha}(x)),y)] + \frac{1}{2}\mathbb{E}_{\mathcal{D}_S}[l_y \circ h(\tilde{\alpha}(x)) \\ &\qquad- l_y \circ h(x)] + \frac{1}{2}\mathcal{R}_S(h) \\ &\quad= \mathbb{E}_{\mathcal{D}_S}[l_y \circ h(\alpha^*(x)) - l_y \circ h(\tilde{\alpha}(x))] \\ &\qquad+ \frac{1}{2}\mathcal{R}_{A|h}(h) + \frac{1}{2}\mathbb{E}_{\mathcal{D}_S}[l_y \circ h(\tilde{\alpha}(x)) \\ &\qquad- l_y \circ h(x)] + \frac{1}{2}\mathcal{R}_S(h).\end{aligned} \tag{18}$$

Since $l_y \circ h$ is β-smooth,

$$\begin{aligned}l_y \circ h(x+\epsilon) &= l_y \circ h(x) + \nabla_x l_y \circ h(x)^T \epsilon + \rho_x(\epsilon) \\ &\le l_y \circ h(x) + \nabla_x l_y \circ h(x)^T \epsilon \\ &\quad+ \frac{1}{2}\beta\|\epsilon\|^2 \le l_y \circ h(\tilde{\alpha}(x)) + \frac{1}{2}\beta\|\epsilon\|_2^2\end{aligned} \tag{19}$$

and thus the first term in (A3) is upper bounded as follows:

$$\mathbb{E}_{(x,y)\sim\mathcal{D}_S}\left[l_y \circ h(\alpha^*(x)) - l_y \circ h(\tilde{\alpha}(x))\right] \le \frac{1}{2}\beta\|\epsilon\|_2^2. \tag{20}$$

Moreover, the third term is upper bounded as follows:

$$\begin{aligned}&\mathbb{E}_{(x,y)\sim\mathcal{D}_S}[l_y \circ h(\tilde{\alpha}(x)) - l_y \circ h(x)] \\ &\quad= \mathbb{E}_{(\tilde{x},y)\sim D_{A|h}}[l_y \circ h(\tilde{x})] - \mathbb{E}_{(x,y)\sim D_S}[l_y \circ h(x)] \\ &\quad\le \max_{y\in\mathcal{Y}} Lip(l_y) \sup_{f\in \mathrm{Lip1}} \\ &\qquad\left[\mathbb{E}_{\tilde{x}\sim D^{\mathcal{X}}_{A|h}}[f(h(\tilde{x}))] - \mathbb{E}_{x\sim D^{\mathcal{X}}_S}[f(h(x))]\right] \\ &\quad= Lip(l) \sup_{f\in \mathrm{Lip1}} \left[\mathbb{E}_{\tilde{z}\sim h\#D^{\mathcal{X}}_{A|h}}[f(\tilde{z})] - \mathbb{E}_{z\sim h\#D^{\mathcal{X}}_S}[f(z)]\right] \\ &\quad= Lip(l) W_1(h\#D^{\mathcal{X}}_{A|h}, h\#D^{\mathcal{X}}_S) \\ &\quad= \sqrt{\frac{c}{c-1}} W_1(h\#D^{\mathcal{X}}_{A|h}, h\#D^{\mathcal{X}}_S),\end{aligned} \tag{21}$$

where $Lip(l) \equiv \max_{y\in\mathcal{Y}} Lip(l_y)$. Combining all (18), (20), and (21) with rearrangement, we can upper bound the adversarial risk as follows:

$$\begin{aligned}\mathcal{R}_{robust}(h;D_S) \le \frac{1}{2}(&\mathcal{R}_S(h) + \mathcal{R}_{A|h}(h) \\ &+ \sqrt{\frac{c}{c-1}} \cdot W_1(h\#D^{\mathcal{X}}_{A|h}, h\#D^{\mathcal{X}}_S) + \beta\|\epsilon\|_2^2).\end{aligned} \tag{22}$$

A.0.1 Feature visualization

To perform qualitative analysis of how well the high-level representations of two distributions are aligned by the proposed approach, we performed t-SNE visualization (Maaten and Hinton 2008). We visualized 2000 clean samples x with corresponding adversarial examples x^{adv} in two dimensions using the SVHN dataset, as shown in Table 3. We showed representations of five models: *Normal*, *Normal + AT (FGSM)*, *ATDA*, *ALP*, and our suggested model (SWAT). For adversarial attacks, we used FGSM ($\epsilon = 8/255, \alpha = \epsilon/4$), PGD ($\epsilon = 8/255, \alpha = \epsilon/10$ with 20 iteration steps), and C &W($c = 1$).

In Table 3, if the logit space distributions of clean and white box attacks are similar to each other, the proposed model exhibits robust performance on both clean and adversarial samples. Regarding the results of FGSM attacks, (i.e., same adversarial attack setting used for adversarial training), one can see that the distributions of clean and adversarial samples are similar. This finding agrees with the results presented in Fig. 4d, where all five models exhibit robustness against FGSM attacks. However, the visualization results for PGD and C &W attacks demonstrate that the distributions of adversarial samples are significantly different from those of clean samples for the previously developed adversarial training models. In contrast, the overall distributions are very similar for our method.

Acknowledgements This work was supported by the National Research Foundation of Korea (NRF) funded by the Korean Government through the Ministry of Science and Information and Communications Technology (MSIT) under Grant 2022R1F1A1074393, 2019R1A2C2002358 and RS-2023-00244896, and by the ITRC (Information Technology Research Center) support program (IITP-2023-2020-0-01789) and the Artificial Intelligence Convergence Innovation Human Resources Development (IITP-2023-RS-2023-00254592). This work was partly supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-01373, Artificial Intelligence Graduate School Program (Hanyang University)). This work is supported in part by Institute of Information & communications Technology Planning & Evaluation (IITP) Grant funded by the Korean Government (MSIT) (No. 2022-0-00984).

Funding Open Access funding enabled and organized by Seoul National University.

Data availability The dataset we used in this paper is a public dataset (CIFAR 10, SVHN, and Fashion MNIST), therefore, it is accessible to everyone and can be reproduced.

References

Allen-Zhu Z, Li Y, Song Z (2019) A convergence theory for deep learning via over-parameterization. In: international conference on machine learning, pp 242–252

Arjovsky M, Bottou L (2017) Towards principled methods for training generative adversarial networks. arxiv e-prints, art. arXiv preprint arXiv:1701.04862

Athalye A, Carlini N, Wagner D (2018) Obfuscated gradients give a false sense of security: circumventing defenses to adversarial examples. In: International conference on machine learning, pp 274–283

Bonnotte N (2013) Unidimensional and evolution methods for optimal transportation. Ph.D. Thesis, Paris 11

Cao N, Li G, Zhu P et al (2019) Handling the adversarial attacks. J Ambient Intell Humaniz Comput 10(8):2929–2943

Carlini N, Wagner D (2017) Towards evaluating the robustness of neural networks. In: 2017 IEEE symposium on security and privacy (SP), IEEE, pp 39–57

Chen PY, Sharma Y, Zhang H, et al (2018) Ead: elastic-net attacks to deep neural networks via adversarial examples. In: Proceedings of the AAAI conference on artificial intelligence, pp 1–19

Cohen J, Rosenfeld E, Kolter Z (2019) Certified adversarial robustness via randomized smoothing. In: International conference on machine learning, pp 1310–1320

Croce F, Andriushchenko M, Sehwag V et al (2020) Robustbench: a standardized adversarial robustness benchmark. arXiv preprint arXiv:2010.09670

Dhillon GS, Azizzadenesheli K, Lipton ZC et al (2018) Stochastic activation pruning for robust adversarial defense. arXiv preprint arXiv:1803.01442

Drewek-Ossowicka A, Pietrołaj M, Rumiński J (2021) A survey of neural networks usage for intrusion detection systems. J Ambient Intell Humaniz Comput 12(1):497–514

Frogner C, Zhang C, Mobahi H et al (2015) Learning with a wasserstein loss. Adv Neural Inf Process Syst 28:1–8

Ganin Y, Ustinova E, Ajakan H et al (2016) Domain-adversarial training of neural networks. J Mach Learn Res 17(1):1–35

Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572

Guo C, Rana M, Cisse M et al (2017) Countering adversarial images using input transformations. arXiv preprint arXiv:1711.00117

Hoffgen KU, Simon HU, Vanhorn KS (1995) Robust trainability of single neurons. J Comput Syst Sci 50(1):114–125

Huang S, Papernot N, Goodfellow I et al (2017) Adversarial attacks on neural network policies. arXiv preprint arXiv:1702.02284

Kannan H, Kurakin A, Goodfellow I (2018) Adversarial logit pairing. arXiv preprint arXiv:1803.06373

Kim H, Lee W, Lee J (2021) Understanding catastrophic overfitting in single-step adversarial training. In: Proceedings of the AAAI conference on artificial intelligence, pp 8119–8127

Kim C, Choi J, Yoon J et al (2023a) Fairness-aware multimodal learning in automatic video interview assessment. IEEE Access 11:122677–122693

Kim H, Lee W, Lee S et al (2023b) Bridged adversarial training. Neural Netw 167:266–282

Kim H, Park J, Choi Y, et al (2023c) Fantastic robustness measures: the secrets of robust generalization. In: Thirty-seventh conference on neural information processing systems

Kim H, Park J, Lee J (2023d) Generating transferable adversarial examples for speech classification. Pattern Recogn 137(109):286

Kolouri S, Pope PE, Martin CE et al (2018) Sliced wasserstein autoencoders. In: International conference on learning representations, pp 1–19

Kolouri S, Nadjahi K, Simsekli U et al (2019) Generalized sliced wasserstein distances. In: NeurIPS 2019, pp 1–12

Krizhevsky A, Hinton G (2009) Learning multiple layers of features from tiny images. Tech. rep, Citeseer

Kurakin A, Goodfellow I, Bengio S (2016) Adversarial examples in the physical world. arXiv preprint arXiv:1607.02533

Lee CY, Batra T, Baig MH, et al (2019) Sliced wasserstein discrepancy for unsupervised domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 10,285–10,295

Lee S, Lee W, Park J et al (2021) Towards better understanding of training certifiably robust models against adversarial examples. Adv Neural Inf Process Syst 34:953–964

Lee W, Kim H, Lee J (2021) Compact class-conditional domain invariant learning for multi-class domain adaptation. Pattern Recogn 112(107):763

Li G, Zhu P, Li J, et al (2018) Security matters: a survey on adversarial machine learning. arXiv preprint arXiv:1810.07339

Li Y, Zhang H, Bermudez C et al (2020) Anatomical context protects deep learning from adversarial perturbations in medical imaging. Neurocomputing 379:370–378

Liu Z, Chan AB (2022) Boosting adversarial robustness from the perspective of effective margin regularization. arXiv preprint arXiv:2210.05118

Long M, Zhu H, Wang J, et al (2017) Deep transfer learning with joint adaptation networks. In: Proceedings of the 34th international conference on machine learning-volume 70, JMLR. org, pp 2208–2217

Lvd Maaten, Hinton G (2008) Visualizing data using t-SNE. J Mach Learn Res 9(Nov):2579–2605

Madry A, Makelov A, Schmidt L et al (2017) Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083

Martin J, Elster C (2020) Inspecting adversarial examples using the fisher information. Neurocomputing 382:80–86

Moosavi-Dezfooli SM, Fawzi A, Frossard P (2016) Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 2574–2582

Netzer Y, Wang T, Coates A, et al (2011) Reading digits in natural images with unsupervised feature learning. NIPS workshop on deep learning and unsupervised feature learning 2011

Pang T, Yang X, Dong Y et al (2020) Boosting adversarial training with hypersphere embedding. arXiv preprint arXiv:2002.08619

Redko I, Habrard A, Sebban M (2017) Theoretical analysis of domain adaptation with optimal transport. In: Joint European conference on machine learning and knowledge discovery in databases. Springer, pp 737–753

Samangouei P, Kabkab M, Chellappa R (2018) Defense-GAN: protecting classifiers against adversarial attacks using generative models. arXiv preprint arXiv:1805.06605

Shafahi A, Najibi M, Ghiasi MA et al (2019) Adversarial training for free! Adv Neural Inf Process Syst 32:1–9

Shaham U, Yamada Y, Negahban S (2018) Understanding adversarial training: increasing local stability of supervised models through robust optimization. Neurocomputing 307:195–204

Sharif M, Bhagavatula S, Bauer L et al (2016) Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. ACM, pp 1528–1540

Song C, He K, Wang L, et al (2018) Improving the generalization of adversarial training with domain adaptation. arXiv preprint arXiv:1810.00740

Stutz D, Hein M, Schiele B (2021) Relating adversarially robust generalization to flat minima. In: Proceedings of the IEEE/CVF international conference on computer vision, pp 7807–7817

Szegedy C, Zaremba W, Sutskever I, et al (2013) Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199

Tzeng E, Hoffman J, Saenko K, et al (2017) Adversarial discriminative domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7167–7176

Wang Y, Wang K, Zhu Z et al (2020) Adversarial attacks on faster r-cnn object detector. Neurocomputing 382:87–95

Wong E, Rice L, Kolter JZ (2020) Fast is better than free: revisiting adversarial training. arXiv preprint arXiv:2001.03994

Wu J, Huang Z, Acharya D, et al (2019) Sliced Wasserstein generative models. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 3713–3722

Xiao H, Rasul K, Vollgraf R (2017) Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv preprint arXiv:1708.07747

Xiao KY, Tjeng V, Shafiullah NM et al (2018) Training for faster adversarial robustness verification via inducing relu stability. arXiv preprint arXiv:1809.03008

Xie C, Wang J, Zhang Z et al (2017) Mitigating adversarial effects through randomization. arXiv preprint arXiv:1711.01991

Ye H, Liu X, Li C (2020) Dscae: a denoising sparse convolutional autoencoder defense against adversarial examples. J Ambient Intell Humaniz Comput 1–11

Yoon T, Lee J, Lee W (2020) Joint transfer of model knowledge and fairness over domains using wasserstein distance. IEEE Access 8:123,783-123,798