A. Related Works

In this section, we briefly review related works used in our paper.

A.1. Adversarial attacks

A growing body of research shows that neural networks are vulnerable to adversarial attacks, i.e., test inputs that are modified slightly yet strategically to cause misclassification (Carlini & Wagner, 2017a; Kurakin et al., 2017; Wang et al., 2019; Zhang et al., 2020d), which seriously threaten the security-critical computer vision systems, such as autonomous driving and medical diagnostics (Chen et al., 2015; Ma et al., 2021; Nguyen et al., 2015; Szegedy et al., 2013). Thus, it is crucial to defend against adversarial attacks (Chen et al., 2020b; Wang et al., 2020a; Zhu et al., 2021), for example, by injecting adversarial examples into training data, adversarial training methods have been proposed in recent years (Madry et al., 2018; Bai et al., 2019; Wang et al., 2020b; Zhang et al., 2020a). However, these defenses can generally be evaded by *optimization-based* (Opt) attacks, either wholly or partially (Carlini & Wagner, 2017a; He et al., 2018; Li & Vorobeychik, 2014)

A.2. Adversarial data detection

For the adversarial defense, in addition to improving models' robustness by more effective adversarial training (Chen et al., 2020a; Wang et al., 2019; Wu et al., 2020; Zhang et al., 2021), recent studies have instead focused on detecting adversarial data. Based on features extracted from DNNs, most works train classifiers to discriminate adversarial data from both natural and adversarial data. Recent studies include, a cascade detector based on the PCA projection of activations (Li & Li, 2017), detection subnetworks based on activations (Metzen et al., 2017), a logistic regression detector based on Kernel Density KD, and Bayesian Uncertainty (BU) features (Grosse et al., 2017), an augmented neural network detector based on statistical measures, a learning framework that covers unexplored space invulnerable models (Rouhani et al., 2017), a *local intrinsic dimensionality* (LID) based characterization of adversarial data (Ma et al., 2018), a generative classifier based on Mahalanobis distance-based score (Lee et al., 2018).

A.3. Statistical adversarial data detection

In the safety-critical system, it is important to find reliable data (i.e., natural data) and eliminate adversarial data that is statistically different from natural data distribution. Thus, statistical detection methods are also proposed to detect if the upcoming data contains adversarial data (or saying that if upcoming data is from natural data distribution in the view of statistics). A number of these methods have been introduced, including the use of the Maximum Mean Discrepancy (MMD) (Gretton et al., 2012b; Borgwardt et al., 2006) with a simple polynomial-time approximation to test whether the upcoming data are all adversarial data, or all natural images (Grosse et al., 2017), and a kernel density estimation defense used a Gaussian Mixture Model to model outputs from the final hidden layer of a neural network, to test whether the upcoming data belongs to a different distribution than that of natural data (Feinman et al., 2017). However, recent studies have shown these statistical detection failed to work under attack evaluations (Carlini & Wagner, 2017a).

A.4. Two-sample Tests

Two-sample tests aim to check whether two datasets come from the same distribution. Traditional tests such as t-test and Kolmogorov-Smirnov test are the mainstream of statistical applications, but require strong assumptions on the distributions being studied. Researchers in statistics and machine learning have been focusing on relaxing these assumptions, with methods specific to various real-world domains (Sugiyama et al., 2011; Yamada et al., 2011; Kanamori et al., 2012; Gretton et al., 2012b; Jitkrittum et al., 2016; Sutherland et al., 2017; Chen & Friedman, 2017; Ghoshdastidar et al., 2017; Lopez-Paz & Oquab, 2017; Li & Wang, 2018; Kirchler et al., 2020; Liu et al., 2020b). In order to involve distributions with complex structure such as images, deep kernel approaches has been proposed (Sutherland et al., 2017; Wenliang et al., 2019; Jean et al., 2018), the foremost study has shown that kernels parameterized by deep neural nets, can be trained to maximize test power in high-dimensional distribution such as images (Liu et al., 2020b). They propose statistical tests of the null hypothesis that the two distributions are equal against the alternative hypothesis that the two distributions are different. Such tests have applications in a variety of machine learning problems such as domain adaptation, covariate shift, label-noise learning, generative modeling, fairness and causal discovery (Binkowski et al., 2018; Zhang et al., 2020c; Fang et al., 2020a; Gong et al., 2016; Fang et al., 2020b; Liu et al., 2019; Zhang et al., 2020e;b; Liu et al., 2020a; Zhong et al., 2021; Yu et al., 2020; Stojanov et al., 2019; Lopez-Paz & Oquab, 2017; Oneto et al., 2020).

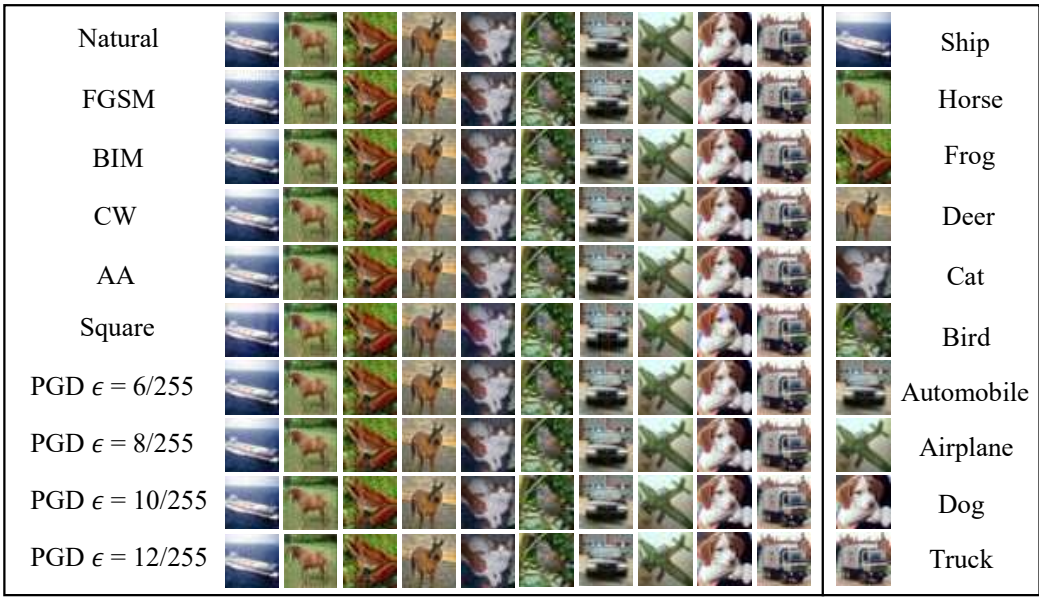


Figure 6. Adversarial data on *CIFAR-10*. We output adversarial examples on a pre-trained ResNet-18, which are attacked by different methods or under different bounded perturbation ϵ.

B. Real-world Scenarios regarding SADD

Scenario 1. As an artificial-intelligence service provider, we need to acquire a client by modeling his/her task well, such as modeling the risk level of manufacturing factory. To finish this task, we need to hire distributed annotators to obtain labeled natural data regarding the risk level in the factory. However, our competitors may *conspire* with several annotators against us, poisoning this training data by injecting malicious adversarial data (Barreno et al., 2010; Kloft & Laskov, 2012). If training data contains adversarial ones, the test accuracy will drop (Zhang et al., 2019), which makes us lose the client unexpectedly. To beware of such adversarial attacks, we can use the MMD test to find reliable annotators providing natural training data.

Scenario 2. As a client, we need to purchase artificial-intelligence services to model our task well, such as modeling the risk level of manufacturing factory mentioned above. Given a variety of models offered by many providers, we should select the optimal one and need to hire distributed annotators to obtain labeled natural data regarding the risk level in our factory. However, some artificial-intelligence service providers may *conspire* with several our annotators, poisoning our testing data by injecting malicious adversarial data (Barreno et al., 2010; Kloft & Laskov, 2012). If the testing data contains adversarial ones which are only in the training set of those conspired providers, the test accuracy of conspired providers' models will surpass that of their competitors (Madry et al., 2018), which makes us fail to select the optimal provider. To beware of such adversarial attacks, we can use the MMD test to find reliable annotators providing natural test data.

C. Hilbert-Schmidt Independence Criteria

The HSIC (Gretton et al., 2008; 2005) is a test statistic to work on independence testing (Gretton et al., 2005). HSIC can be interpreted as the distance between embeddings of the joint distribution and the product of the marginals in a RKHS. More importantly, HSIC between two random variables is zero if and only if the two variables are independent (Sriperumbudur et al., 2010). Under the null hypothesis of independence, $P_{XY} = P_X P_Y$, the minimum variance estimate of HSIC is a degenerate U-statistic. The formulation of the HSIC is as follows (more details can be found in (Gretton et al., 2005)). Given two sets of data S_X and S_Y (with size n), the HSIC can be computed using

$$\begin{aligned}\mathrm{HSIC}(S_X, S_Y) = &\mathbb{E}_{(x_1,y_1)\sim p(x,y),(x_2,y_2)\sim p(x,y)}[\kappa_X(x_1,x_2)\kappa_Y(y_1,y_2)] \\ &+ \mathbb{E}_{x_1\sim p(x),x_2\sim p(x),y_1\sim p(y),y_2\sim p(y)}[\kappa_X(x_1,x_2)\kappa_Y(y_1,y_2)] \\ &- 2\mathbb{E}_{(x_1,y_1)\sim p(x,y),x_2\sim p(x),y_2\sim p(y)}[\kappa_X(x_1,x_2)\kappa_Y(y_1,y_2)],\end{aligned} \tag{12}$$

where κ_X and κ_Y are two Gaussian kernel functions whose bandwidths are set to two constants, and

$$\mathbb{E}_{(x_1,y_1)\sim p(x,y),(x_2,y_2)\sim p(x,y)}[\kappa_X(x_1,x_2)\kappa_Y(y_1,y_2)] = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}[\kappa_X(x_i,x_j)\kappa_Y(y_i,y_j)]. \tag{13}$$

D. Asymptotics of the SAMMD

In this section, we will first prove the asymptotics of the SAMMD by assuming that the adversarial data $\{Y_i\}_{i\in\mathbb{Z}^+}$ are an absolutely regular process with mixing coefficients $\{\beta_k\}_{k>0}$ defined in the following.

Definition 1 (Absolutely regular process)**.** *(i) Let $(\Omega,\mathcal{A},\mathbb{Q})$ be a probability space, and let $\mathcal{A}_1,\mathcal{A}_2$ be sub-σ-field of $\mathcal{A}$. We define*

$$\beta(\mathcal{A}_1,\mathcal{A}_2) = \sup_{A_1,\ldots,A_n,B_1,\ldots,B_m}\sum_{i=1}^{n}\sum_{j=1}^{m}|\mathbb{Q}(A_i\cap B_j)-\mathbb{Q}(A_i)\mathbb{Q}(B_j)|, \tag{14}$$

where the supremum is taken over all partitions $A_1,\ldots,A_n$ and $B_1,\ldots,B_m$ of Ω into elements of $\mathcal{A}_1$ and $\mathcal{A}_2$, respectively. (ii) Given a stochastic process $\{Y_i\}_{i\in\mathbb{Z}^+}$ and integers $1\le a\le b$, we denote by $\mathcal{A}_a^b$ the σ-field generated by the random variables $Y_{a+1},\ldots,Y_b$. We define the mixing coefficients of absolute regularity by

$$\beta_k = \sup_{n\in\mathbb{Z}^+}\beta(\mathcal{A}_1^n,\mathcal{A}_{n+k}^{\infty}). \tag{15}$$

The process $\{Y_i\}_{i\in\mathbb{Z}^+}$ is called absolutely regular if $\lim_{k\to\infty}\beta_k = 0$.

Then, we can obtain the main theorem in the following.

Theorem 2 (Asymptotics under H_1)**.** *Under the alternative, $H_1: S_Y$ are from a stochastic progress $\{Y_i\}_{i=1}^{+\infty}$, if $\{Y_i\}_{i=1}^{+\infty}$ is an absolutely regular process with mixing coefficients $\{\beta_k\}_{k>0}$ satisfying $\sum_{k=1}^{+\infty}\beta_k^{\delta/(2+\delta)}<+\infty$ for some $\delta>0$, then $\widehat{\text{SAMMD}}_u^2$ is $\mathcal{O}_P(1/n)$, and in particular*

$$\sqrt{n}(\widehat{\text{SAMMD}}_u^2 - \text{SAMMD}^2) \xrightarrow{\boldsymbol{d}} \mathcal{N}(0, C_1^2\sigma_{H_1}^2),$$

where $Y_i = \mathcal{G}_{\ell,\hat{f}}(\mathcal{B}_\epsilon[X_i'']) \sim \mathbb{Q}$, $X_i''\sim\mathbb{P}$, $\sigma_{H_1}^2 = 4(\mathbb{E}_Z[(\mathbb{E}_{Z'}h(Z,Z'))^2] - [(\mathbb{E}_{Z,Z'}h(Z,Z'))^2])$, $h(Z,Z') = k_\omega(X,X') + k_\omega(Y,Y') - k_\omega(X,Y') - k_\omega(X',Y)$, $Z:=(X,Y)$, $X\sim\mathbb{P}$ and X'' are independent and $C_1<+\infty$ is a constant for a given ω.

Proof. Without loss of generality, let Z be a random variable on a probability space $(\Omega^Z,\mathcal{A}^Z,\mathbb{Q}^Z)$. We will first prove that $\{Z\}_{i=1}^{+\infty}$ is an absolutely regular process. According to Eq. (14), we have

$$\beta^Z(\mathcal{A}_1^Z,\mathcal{A}_2^Z) = \sup_{A_1^Z,\ldots,A_n^Z,B_1^Z,\ldots,B_m^Z}\sum_{i=1}^{n}\sum_{j=1}^{m}|\mathbb{Q}^Z(A_i^Z\cap B_j^Z)-\mathbb{Q}^Z(A_i^Z)\mathbb{Q}^Z(B_j^Z)|, \tag{16}$$

where $\mathcal{A}_1^Z,\mathcal{A}_2^Z$ are sub-σ-field of $\mathcal{A}^Z$ generated by $\{Z\}_{i=1}^{+\infty}$ and the supremum is taken over all partitions $A_1^Z,\ldots,A_n^Z$ and $B_1^Z,\ldots,B_m^Z$ of Ω into elements of $\mathcal{A}_1^Z$ and $\mathcal{A}_2^Z$, respectively. Since X and Y are independent and $Z=(X,Y)$, $\mathbb{Q}^Z(Z\in A^Z) = \mathbb{Q}^Z(X\in A^X, Y\in A) = \mathbb{P}(A^X)\mathbb{Q}(A)$. Thus, we have $\mathbb{Q}^Z(A_i^Z\cap B_j^Z) = \mathbb{P}(A_i^X\cap B_i^X)\mathbb{Q}(A_i\cap B_i)$, $\mathbb{Q}^Z(A_i^Z) = \mathbb{P}(A_i^X)\mathbb{Q}(A_i)$, and $\mathbb{Q}^Z(B_i^Z) = \mathbb{P}(B_i^X)\mathbb{Q}(B_i)$. Since X and X' are independent, we have $\mathbb{P}(A_i^X\cap B_i^X) = \mathbb{P}(A_i^X)\mathbb{P}(B_i^X)$, meaning that

$$\beta^Z(\mathcal{A}_1^Z,\mathcal{A}_2^Z) = \sup_{A_1^Z,\ldots,A_n^Z,B_1^Z,\ldots,B_m^Z}\sum_{i=1}^{n}\sum_{j=1}^{m}\mathbb{P}(A_i^X\cap B_i^X)|\mathbb{Q}(A_i\cap B_j)-\mathbb{Q}(A_i)\mathbb{Q}(B_j)|. \tag{17}$$

Due to the supremum, we can safely make $\mathbb{P}(A_i^X\cap B_i^X)$ be 1. Thus, we have $\beta^Z(\mathcal{A}_1^Z,\mathcal{A}_2^Z) = \beta(\mathcal{A}_1,\mathcal{A}_2)$. Namely, $\{Z\}_{i=1}^{+\infty}$ is an absolutely regular process with mixing coefficients $\{\beta_k\}_{k>0}$ satisfying $\sum_{k=1}^{+\infty}\beta_k^{\delta/(2+\delta)}<+\infty$. Based on Theorem 1 in (Denker & Keller, 1983), since $h(\cdot,\cdot)\le 2$, we know that

$$\sqrt{n}(\widehat{\text{SAMMD}}_u^2 - \text{SAMMD}^2) \xrightarrow{\mathbf{d}} \mathcal{N}(0, 4\sigma^2), \tag{18}$$

where

$$\sigma^2 = \underbrace{\mathbb{E}[h_1(Z_1)]^2}_{\sigma_{H_1}^2} + 2\sum_{j=1}^{+\infty} \mathrm{cov}(h_1(Z_1), h_1(Z_j)), \tag{19}$$

$h_1(Z_j) = \mathbb{E}_{Z_i} h(Z_i, Z_j) - \theta$, and $\theta = \mathbb{E}_{Z_i, Z_j} h(Z_i, Z_j)$. Note that, due to $\mathbb{P} \neq \mathbb{Q}$, we know $\sigma > 0$; due to the absolute regularity, $\sigma < +\infty$. Since the possible dependence between Z_1 and Z_j are caused by Y_1 and Y_j, we will calculate the second term in the right side of Eq. (19) in the following. First, we introduce two notations for the convenience.

$$\mathbb{E}_X^{(i)} = \mathbb{E}_X[k_\omega(X_i, X) - k_\omega(Y_i, X)], \tag{20}$$

$$\mathbb{E}_Y^{(i)} = \mathbb{E}_Y[k_\omega(X_i, Y) - k_\omega(Y_i, Y)]. \tag{21}$$

Thus, we know

$$h_1(Z_1) = \underbrace{\mathbb{E}_X^{(1)} + \mathbb{E}_Y^{(1)}}_{\tilde{h}_1(Z_1)} - \theta,\ h_1(Z_j) = \underbrace{\mathbb{E}_X^{(j)} + \mathbb{E}_Y^{(j)}}_{\tilde{h}_1(Z_j)} - \theta,\ \theta = \mathbb{E}_{Z_1}[\mathbb{E}_X^{(1)} + \mathbb{E}_Y^{(1)}], \tag{22}$$

and

$$\theta^2 = \mathbb{E}_{Z_1}[\mathbb{E}_X^{(1)} + \mathbb{E}_Y^{(1)}]\mathbb{E}_{Z_j}[\mathbb{E}_X^{(j)} + \mathbb{E}_Y^{(j)}] = \Big(\mathbb{E}_{Z_1}[\mathbb{E}_X^{(1)}] + \mathbb{E}_{Z_1}[\mathbb{E}_Y^{(1)}]\Big)\Big(\mathbb{E}_{Z_j}[\mathbb{E}_X^{(j)}] + \mathbb{E}_{Z_j}[\mathbb{E}_Y^{(j)}]\Big). \tag{23}$$

Then, we can compute the $\mathrm{cov}(h_1(Z_1), h_1(Z_j))$.

$$\begin{aligned}
\mathrm{cov}(h_1(Z_1), h_1(Z_j)) =& \mathbb{E}_{Z_1, Z_j}[(\tilde{h}_1(Z_1) - \theta)(\tilde{h}_1(Z_j) - \theta)] \\
=& \mathbb{E}_{Z_1, Z_j}[\tilde{h}_1(Z_1)\tilde{h}_1(Z_j) - \theta\tilde{h}_1(Z_j) - \theta\tilde{h}_1(Z_1) + \theta^2] \\
=& \mathbb{E}_{Z_1, Z_j}[\tilde{h}_1(Z_1)\tilde{h}_1(Z_j)] - \theta^2 \\
=& \mathbb{E}_{Z_1, Z_j}[(\mathbb{E}_X^{(1)} + \mathbb{E}_Y^{(1)})(\mathbb{E}_X^{(j)} + \mathbb{E}_Y^{(j)})] - \theta^2 \\
=& \mathbb{E}_{Z_1, Z_j}[\mathbb{E}_X^{(1)}\mathbb{E}_X^{(j)} + \mathbb{E}_X^{(1)}\mathbb{E}_Y^{(j)} + \mathbb{E}_Y^{(1)}\mathbb{E}_X^{(j)} + \mathbb{E}_Y^{(1)}\mathbb{E}_Y^{(j)}] - \theta^2 \\
=& \mathbb{E}_{Z_1}[\mathbb{E}_X^{(1)}]\mathbb{E}_{Z_j}[\mathbb{E}_X^{(j)}] + \mathbb{E}_{Z_1}[\mathbb{E}_X^{(1)}]\mathbb{E}_{Z_j}[\mathbb{E}_Y^{(j)}] + \mathbb{E}_{Z_1}[\mathbb{E}_Y^{(1)}]\mathbb{E}_{Z_j}[\mathbb{E}_X^{(j)}] + \mathbb{E}_{Z_1, Z_j}[\mathbb{E}_Y^{(1)}\mathbb{E}_Y^{(j)}] - \theta^2.
\end{aligned} \tag{24}$$

Substituting Eq. (23) into Eq. (24), we have

$$\mathrm{cov}(h_1(Z_1), h_1(Z_j)) = \mathbb{E}_{Z_1, Z_j}[\mathbb{E}_Y^{(1)}\mathbb{E}_Y^{(j)}] - \mathbb{E}_{Z_1}[\mathbb{E}_Y^{(1)}]\mathbb{E}_{Z_j}[\mathbb{E}_Y^{(j)}]. \tag{25}$$

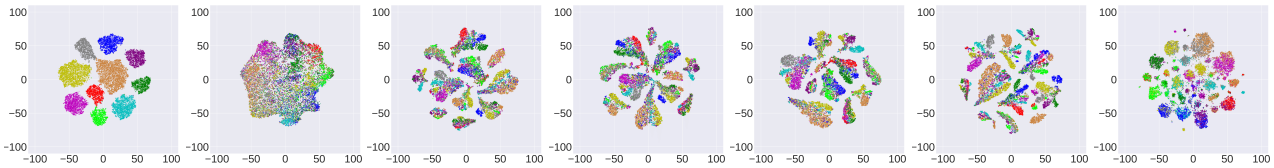
Then, substituting Eq. (21) into Eq. (25), we have

$$\begin{aligned}
\mathrm{cov}(h_1(Z_1), h_1(Z_j)) =& \mathbb{E}_{Y_1, Y_j}\left[\mathbb{E}_Y\mathbb{E}_Y[k_\omega(Y_1, Y)k_\omega(Y_j, Y)]\right] - \mathbb{E}_{Y_1}\left[\mathbb{E}_Y[k_\omega(Y_1, Y)]\right]\mathbb{E}_{Y_j}\left[\mathbb{E}_Y[k_\omega(Y_j, Y)]\right] \\
=& \mathbb{E}_Y\mathbb{E}_Y\left[\mathbb{E}_{Y_1, Y_j}[k_\omega(Y_1, Y)k_\omega(Y_j, Y)] - \mathbb{E}_{Y_1}[k_\omega(Y_1, Y)]\mathbb{E}_{Y_j}[k_\omega(Y_j, Y)]\right].
\end{aligned} \tag{26}$$

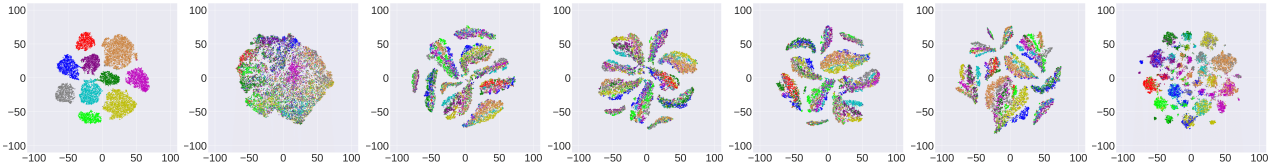
Since $k_\omega(\cdot, \cdot) \leq 1$, according to Lemma 1 in (Yoshihara, 1976), we have $\mathrm{cov}(h_1(Z_1), h_1(Z_j)) < 4\beta_j^{\delta/(2+\delta)}$. Because $\sum_{k=1}^{+\infty} \beta_k^{\delta/(2+\delta)} < +\infty$, we know, $\forall \epsilon' \in (0, 1)$, there exists an N such that $\sum_{k=N+1}^{+\infty} \beta_k^{\delta/(2+\delta)} < \epsilon'$. Hence

$$\sum_{j=1}^{+\infty} \mathrm{cov}(h_1(Z_1), h_1(Z_j)) = \sum_{j=1}^{N} \mathbb{E}_Y\mathbb{E}_Y\left[\mathbb{E}_{Y_1, Y_j}[k_\omega(Y_1, Y)k_\omega(Y_j, Y)] - \mathbb{E}_{Y_1}[k_\omega(Y_1, Y)]\mathbb{E}_{Y_j}[k_\omega(Y_j, Y)]\right] + c', \tag{27}$$

where c' is a small constant. Without loss of generality, we assume the small constant c' is smaller than $\mathbb{E}[h_1(Z_1)]^2$. Thus, there exists a constant $C_1^2 - 1$ such that $2\sum_{j=1}^{+\infty} \mathrm{cov}(h_1(Z_1), h_1(Z_j)) = (C_1^2 - 1)\mathbb{E}[h_1(Z_1)]^2$. Namely, $\sigma^2 = C_1^2\sigma_{H_1}^2$, which completes the proof. □



(a) RN18-Natural (b) RN18-FGSM (c) RN18-BIM (d) RN18-PGD (e) RN18-CW (f) RN18-AA (g) RN18-Square



(h) RN34-Natural (i) RN34-FGSM (j) RN34-BIM (k) RN34-PGD (l) RN34-CW (m) RN34-AA (n) RN34-Square

Figure 7. Visualization of outputs using t-SNE. This figure visualizes outputs of the second to last layers in ResNet-18 and ResNet-34. Different colors represent different semantic meanings (i.e., different classes in the testing set of the *SVHN*).

Next, we will show that the bootstrapped SAMMD (shown in the following) has the same asymptotic null distribution as the empirical SAMMD. First, we restate the bootstrapped SAMMD in the following.

$$\begin{aligned}&\widehat{\mathrm{SAMMD}}_w(S_X, S_Y; k_\omega)\\ =\ &\frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n}\tilde{W}_i^x\tilde{W}_j^x k_\omega(x_i,x_j)+\frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\tilde{W}_i^y\tilde{W}_j^y k_\omega(y_i,y_j)-\frac{2}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m}\tilde{W}_i^x\tilde{W}_j^y k_\omega(x_i,y_j),\end{aligned}\tag{28}$$

where

$$\{\tilde{W}_i^X\}_{i=1}^n=\{W_i^X\}_{i=1}^n-\frac{1}{n}\sum_{i=1}^{n}W_i^X,\ \{\tilde{W}_i^Y\}_{i=1}^m=\{W_i^Y\}_{i=1}^m-\frac{1}{m}\sum_{i=1}^{m}W_i^Y.\tag{29}$$

The W_i^X and W_j^Y are generated by

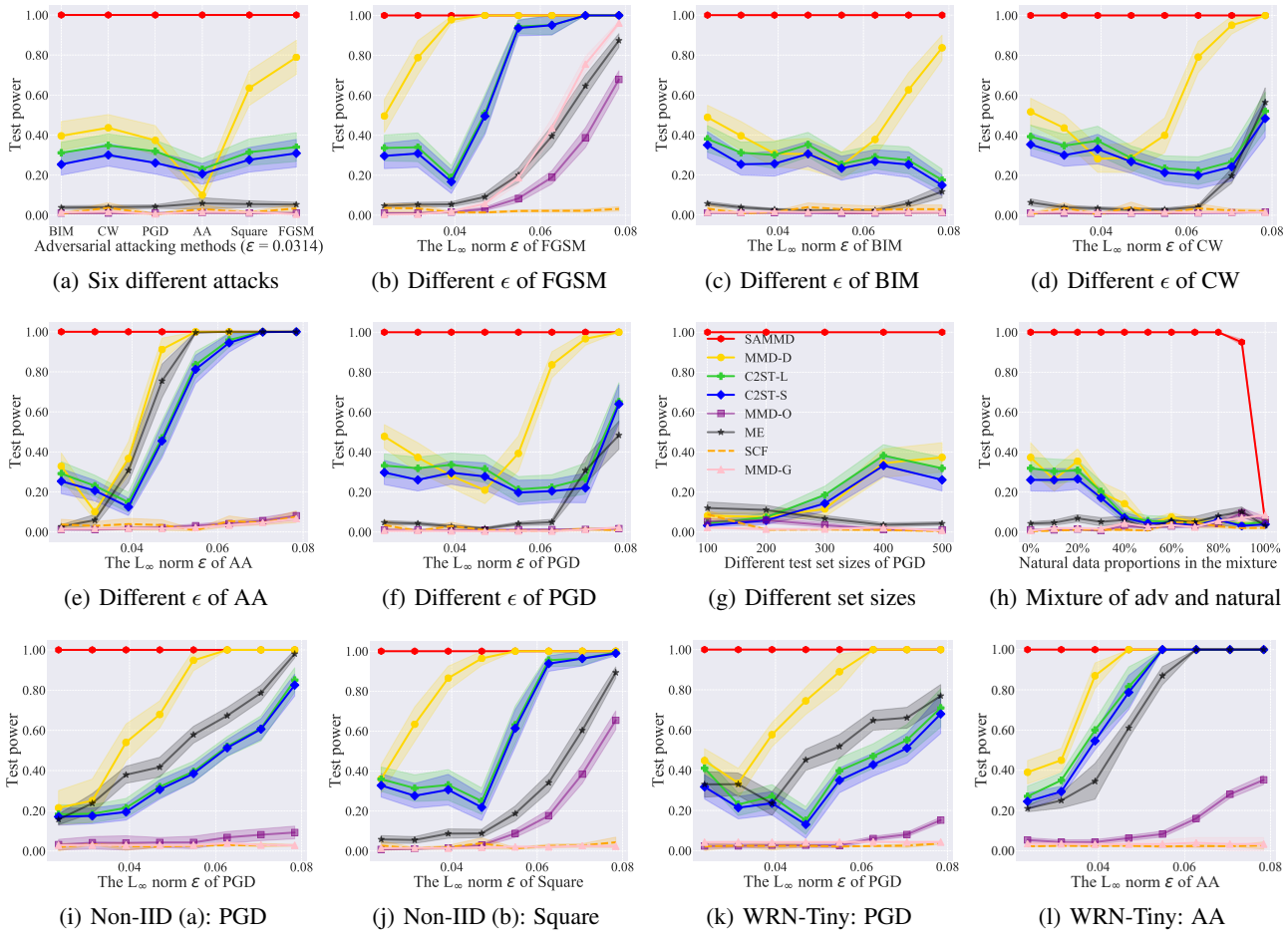
$$W_t=e^{-1/l}W_{t-1}+\sqrt{1-e^{-2/l}}\epsilon_t,\tag{30}$$

where $W_0,\epsilon_0,\ldots,\epsilon_t$ are independent standard normal random variables. Then, following the Proposition 1 in (Chwialkowski et al., 2014), we can directly obtain the following proposition using the relation between β-mixing and τ-mixing presented in Eq. (18) in (Chwialkowski et al., 2014).

Proposition 1. *Let $\{Y_i\}_{i=1}^{+\infty}$ be an absolutely regular process with mixing coefficients $\{\beta_k\}_{k>0}$ satisfying $\beta_k = O(k^{(-6-\epsilon'')(1+\delta)})$ for some $\epsilon>0$ and $\delta>0$, $n=\rho_x n'$ and $m=\rho_y n'$, where $n'=n+m$. Then, under the null hypothesis $Y_i\sim\mathbb{P}$, $\psi(\rho_x\rho_y n'\widehat{\mathrm{SAMMD}}_w(S_X,S_Y;k_\omega),\rho_x\rho_y n'\widehat{\mathrm{SAMMD}}(S_X,S_Y;k_\omega))\to 0$ in probability as $n'\to+\infty$, where ψ is the Prokhorov metric.*

E. Experiments Setup

We implement all methods on Python 3.7 (Pytorch 1.1) with a NVIDIA GeForce RTX2080 Ti GPU. The *CIFAR-10* dataset and the *SVHN* dataset can be downloaded via Pytorch. See the codes submitted. Given the $50,000$ images from the *CIFAR-10* training set and $73,257$ digits from the *SVHN* training set, we conduct a standard training on ResNet-18 and ResNet-34 for classification. Given the $100,000$ images from the *Tiny-Imagenet* training set, we conduct a standard training on WRN-32-10 classification. DNNS are trained using SGD with 0.9 momentum, the initial learning rate of 0.01 and the batch size of 128 for 150 epochs. Based on these pre-trained models, adversarial data is generated from *fast gradient sign method* (FGSM) (Goodfellow et al., 2015), *basic iterative methods* (BIM) (Kurakin et al., 2017), *project gradient descent* (PGD) (Madry et al., 2018), *Carlini and Wagner attack* (CW) (Carlini & Wagner, 2017b), *AutoAttack* (AA) (Croce & Hein, 2020) and *Square attack* (Square) (Andriushchenko et al., 2020).



(a) Six different attacks (b) Different ϵ of FGSM (c) Different ϵ of BIM (d) Different ϵ of CW

(e) Different ϵ of AA (f) Different ϵ of PGD (g) Different set sizes (h) Mixture of adv and natural

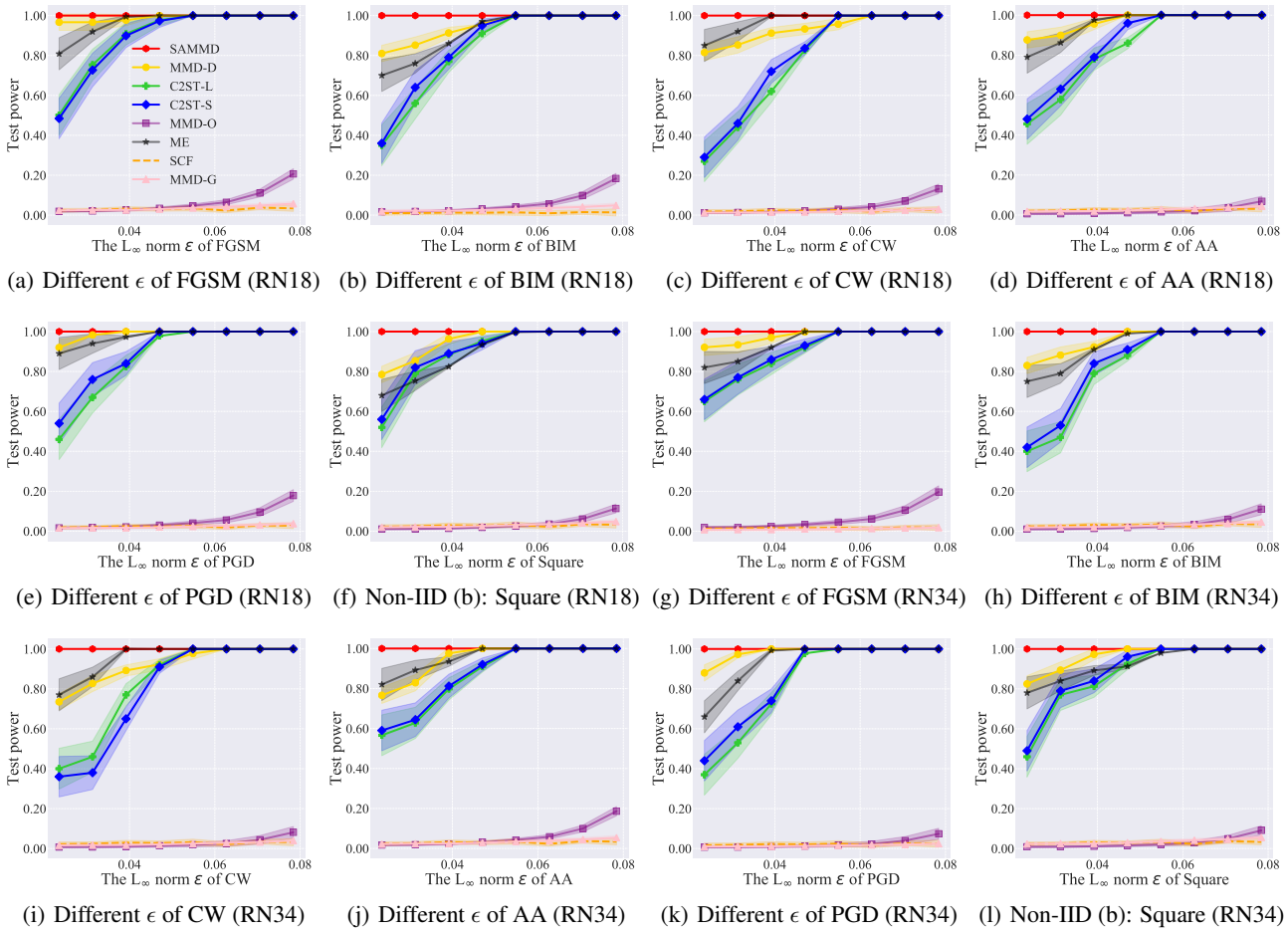
(i) Non-IID (a): PGD (j) Non-IID (b): Square (k) WRN-Tiny: PGD (l) WRN-Tiny: AA

Figure 8. Results of adversarial data detection. Subfigures (a)-(l) report the test power (i.e., the detection rate) when S_Y are adversarial data. The ideal test power is 1 (i.e., 100% detection rate). Subfigure (a) - (j) are the experiments on the adversarial data of the *CIFAR-10* acquired by ResNet-34. Subfigure (k)-(l) are the experiments on the adversarial data of the *Tiny-Imagenet* acquired by WRN-32-10.

For each attack method, we generate eight adversarial datasets with the L_∞-norm bounded perturbation $\epsilon \in [0.0235, 0.0784]$. For attack methods except FGSM, maximum step $K = 20$ and step size $\alpha = \epsilon/10$. For the $10,000$ images from the *CIFAR-10* testing set and $26,032$ digits from the *SVHN* testing set, we only choose the adversarial data, whose original images are correctly classified by the pre-trained models. We extract semantic features from the second to last full connected layer of the well-trained ResNet-18 and ResNet-34. In the wild bootstrap process of SAMMD, given an alternative value set $\{0.1, 0.2, 0.5, 1, 5, 10, 15, 20\}$, we choose the optimal l (in Eq. (6)) depending on whose type I error on natural data is the most close to α ($\alpha = 0.05$ in the paper). The optimal value of l we choose is 0.2. For images from the *CIFAR-10* (or the *SVHN*) testing set and adversarial datasets generated above, we select a subset containing 500 images of the each for S_p^{tr} and S_q^{tr}, and train on that; we then evaluate on 100 random subsets of each, disjoint from the training set, of the remaining data. We repeat this full process 20 times, and report the mean rejection rate of each test. The learning rate of our SAMMD test and all baselines is 0.0002.

F. Additional Experiments

Results of ResNet-34 on the *CIFAR-10*. For the adversarial data acquired by ResNet-34 on *CIFAR-10*, we also compare our SAMMD test with baselines in Figure 8. Figure 8a is a supplement of ResNet-18 to Figure 4 that different ϵ of PGD. For 6 different attacks, FGSM, BIM, PGD, AA, CW and Square (the Non-IID (b)), Figure 8b reports the test power of all tests when S_Y are adversarial data (L_∞ norm $\epsilon = 0.0314$; set size $= 500$). Figure 8(c)-(h) report the average test power on



(a) Different ϵ of FGSM (RN18) (b) Different ϵ of BIM (RN18) (c) Different ϵ of CW (RN18) (d) Different ϵ of AA (RN18)

(e) Different ϵ of PGD (RN18) (f) Non-IID (b): Square (RN18) (g) Different ϵ of FGSM (RN34) (h) Different ϵ of BIM (RN34)

(i) Different ϵ of CW (RN34) (j) Different ϵ of AA (RN34) (k) Different ϵ of PGD (RN34) (l) Non-IID (b): Square (RN34)

Figure 9. Results of adversarial data detection on the *SVHN*. Subfigure (a)-(l) report the test power (i.e., the detection rate) when S_Y are adversarial data. The ideal test power is 1 (i.e., 100% detection rate).

Table 2. Average type I error within natural data and natural data on the *SVHN* and *Tiny-ImageNet*.

Attack	SAMMD	MMD-D	C2ST-L	C2ST-S	MMD-O	ME	SCF	MMD-G
SVHN	0.053±0.017	0.037±0.011	0.036±0.010	0.043±0.012	0.017±0.004	0.022±0.005	0.022±0.006	0.015±0.004
Tiny-ImageNet	0.049±0.015	0.046±0.019	0.051±0.023	0.052±0.016	0.048±0.010	0.039±0.007	0.021±0.008	0.047±0.013

different ϵ of FGSM, BIM, AA, CW and Square (set size $= 500$). Figure 8i reports the average test power on different set sizes. Figure 8j reports the average test power when adversarial data and natural data mix. Results show that our SAMMD test also achieves the highest test power.

Results of ResNet-18 and ResNet-34 on the *SVHN*. We compare the SAMMD test with 6 existing two-sample tests on the *SVHN*. All baselines and experiments setting are the same as those stated in Section 7. We report the type I error in Table 2. The ideal type I error should be around α (0.05 in this paper). For 6 different attacks that FGSM, BIM, PGD, CW, AA, Square and different L_∞-norm bounded perturbation ϵ, we report the test power of all tests when S_Y are adversarial data in Figure 9. Results show that our SAMMD test also performs the best. Compared to results on *CIFAR-10*, adversarial data generated on the *SVHN* is more easily detected by these state-of-the-art tests.

Results of Wide ResNet on the *Tiny-Imagenet*. We also validate the effectiveness of SAMMD on the larger network WRN-32-10 and the larger dataset *Tiny-Imagenet*. All baselines and experiments setting are the same as those stated in

Table 3. The average runtime of the SAMMD test and baselines.

Attack	SAMMD	MMD-D	C2ST-L	C2ST-S	MMD-O	ME	SCF	MMD-G
Runtime(s)	12.51±2.97	47.26±5.92	48.82±4.28	160.78±13.47	11.13±2.15	56.25±8.34	3.59±1.08	1.23±0.17

Section 7. We report the type I error in Table 2. For different attacks PGD and AA, we report the test power of all tests when S_Y are adversarial data in Figure 8 (k)-(l). Results show that our SAMMD test also performs the best.

Time complexity of the SAMMD test. Let E denote the cost of computing an embedding $\phi_p(\boldsymbol{x})$, and K denote the cost of computing $s_{\hat{f}}(\boldsymbol{x}, \boldsymbol{y})$ given $\phi_p(\boldsymbol{x})$, $\phi_p(\boldsymbol{y})$ in Eq. (7). Then each iteration of training in Algorithm 1 costs $O(mE + m^2K)$, where m is the minibatch size.

The average runtime. For images from the *CIFAR-10* testing set and adversarial datasets generated by PGD, we select the subset containing 500 images of the each for S_p^{tr} and S_q^{tr}, and train on that; we then evaluate on 100 random subsets of each, disjoint from the training set, of the remaining data. We repeat this full process 1 times and report the average runtime of our SAMMD test and baselines in Table 3, and the units are seconds.