Defending Adversarial Attacks via Semantic Feature Manipulation

Shuo Wang, Member, IEEE, Surya Nepal, Member, IEEE, Carsten Rudolph, Member, IEEE, Marthie Grobler, Member, IEEE, Shangyu Chen, Tianle Chen, and Zike An

Abstract—Machine learning models have demonstrated vulnerability to adversarial attacks, more specifically misclassification of adversarial examples. In this article, we propose a one-off and attack-agnostic Feature Manipulation (FM)-Defense to detect and purify adversarial examples in an interpretable and efficient manner. The intuition is that the classification result of a normal image is generally resistant to non-significant intrinsic feature changes, e.g., varying the thickness of handwritten digits. In contrast, adversarial examples are sensitive to such changes since the perturbation lacks transferability. To enable manipulation of features, a Combo-variational autoencoder is applied to learn disentangled latent codes that reveal semantic features. The resistance to classification change over the morphs, derived by varying and reconstructing latent codes, is used to detect suspicious inputs. Furthermore, Combo-VAE is enhanced to purify the adversarial examples with good quality by considering class-shared and class-unique features. We empirically demonstrate the effectiveness of detection and quality of purified instances. Our experiments on three datasets show that FM-Defense can detect nearly 100 percent of adversarial examples produced by different state-of-the-art adversarial attacks. It achieves more than 99 percent overall purification accuracy on the suspicious instances that close the manifold of clean examples.

Index Terms—Adversarial attacks, artificial intelligence, defense, latent representation, security

1 INTRODUCTION

The existence of adversarial examples causes serious security concerns, particularly in casting doubt on the reliability of Deep Neural Networks (DNNs) in the case of image classification. These adversarial examples can be generated by adding visually imperceptible perturbations to a normal image to cause a DNN to mislabel the perturbed images with high confidence [1], [2]. Such adversarial attacks may lead to catastrophic consequences in applications such as disease diagnosis and self-driving cars. Existing defensive approaches proposed in the literature to defeat adversarial threats can be categorized as adversarial training, defensive distillation, and detecting/purifying adversarial examples. The first two methods involve either modifying the protected classifier or requiring knowledge of the process to generate adversarial examples. The third method aims at identifying suspicious inputs from normal inputs using hand-crafted statistical features [3], separate classification networks [4], [5] or autoencoders [6]. Unfortunately, the Carlini-Wagner (CW) attack [7] has demonstrated that most existing detection approaches can be evaded. Therefore, the efficient detection of adversarial examples without knowledge of adversarial example generation remains a challenge for machine learning and security communities. This work aims at an efficient detecting and purifying defense.

Existing detection-based defensive approaches include Defense-GAN [8], MagNet [6], FBGAN [9] and Image Transformation-based detection [5]. Defense-GAN trains a GAN to generate the manifold of unperturbed images and then finds the nearest point on the manifold to the adversarial example as the denoising result. MagNet applies detector networks to learn and differentiate between normal and adversarial examples by approximating the normal examples’ manifold. Applying the reformer network moves the adversarial examples towards the manifold of normal examples to correctly reconstruct adversarial examples with small perturbation. FBGAN extracts the semantic features of the input images and reconstructs the denoised images from these features. It uses Bidirectional GAN’s generative capability and the mutual information (MI) regularization between all latent codes and the generated images for disentanglement.

Image Transformation-based detection applies certain transformation operations on an image to generate several transformed images. Then the classification results of these transformed images are used to distinguish between the normal and the adversarial. These approaches present two significant drawbacks, that is, feasibility and completeness.

Feasibility. Element-wise metrics, such as the pixel-wise squared error, are commonly adopted for reconstruction error-based adversarial detection, such as in MagNet. As a reconstruction error is a constant value, a threshold can be set as a hyperparameter to decide whether the input is normal or adversarial. However, the reconstruction error of the perturbed images derived from oblivious attacks (such as...
The CW attack) is very likely similar to normal images. The threshold should be as low as possible to identify slightly perturbed adversarial examples, considering that too low would significantly misjudge normal examples. Besides, for Magnet and similar detection approaches, a clean validation dataset is required to decide the distinguishment threshold. However, it is hard to ensure that all training dataset images are clean in practice. For image transformation-based detection, a mixture of normal and adversarial examples is required in the training set to train the classifier. This causes the high computational cost to generate adversarial examples and adversarial classifiers, particularly for more complex datasets and stronger attacks. Further, the impact of external feature transformation, such as image rotation and shifting, is not consistent and general for different instances. Besides, the background of an image adds a large number of extra features to the object, which is also sensitive to the external transformation.

Completeness. Generally, the limitation of disentangling VAE, e.g., β-VAE, is that the disentanglement ability is at the cost of reconstruction quality. Besides, there are two types of semantic features: class-shared (such as the thickness of the handwritten digits, the facial expression of the face images) and class-unique (such as different handwriting styles for each digit or the identification of face). However, the ordinary disentangling VAEs, e.g., β-VAE, and the reformer of MagNet and the generator of FBGAN and Defense-GAN can only reconstruct/generate the purified instance using some commonly shared feature. It leads to a loss of class-unique features, for example, different writing styles exclusive to the digit “2” (such as a flat stroke or across the loop bottom); such features become inactive for some classification tasks, e.g., writer identification or face recognition. As the high dimension space input, e.g., images, always lies in a complex manifold, the underlying data distribution could be very complex.

The purifier is used to build complex enough models to capture the true posterior by utilizing both class-shared and class-unique features.

This paper proposes an adversarial example detection and purification method, named Feature-Manipulation defense (FM-Defense), to address these two concerns. It can effectively defeat state-of-the-art adversarial attacks, including CW attacks. The intuition is that the classification result of a normal image is generally resistant to non-significant intrinsic feature changes, e.g., varying thickness of handwritten digits or facial expression. Namely, the classification results of a normal digit and its morphs, derived by varying thickness, are very likely to be stable, since significant features are retained. In contrast, adversarial examples are sensitive to such changes. The reason is that the unstructured perturbation is designed for a single image, and may cause various impacts on its morphs due to the lack of transferability. Fig. 1 demonstrates our intuition using a handwritten digit, trouser shape and face image.

The key point to implement our intuition is how to manipulate the intrinsic feature. Consequently, a one-off Combo-variational autoencoder (Combo-VAE) is applied to learn disentangled low-dimensional latent codes, i.e., one latent code only affects one semantic feature. The learned latent codes are disentangled, easy to control, and composed of abundant internal semantic features, instead of external features such as image rotation and shifting. When manipulating the intrinsic features via disentangled latent codes, the resistance of classification accuracy is used to detect suspicious inputs. As demonstrated in the histograms of Fig. 1, the classification accuracy change of clean instances is more consistent than that of adversarial instances. Hence, a simple threshold of classification accuracy resistance can be set to easily distinguish normal and adversarial images instead of training the adversarial classifier. Furthermore, the Combo-VAE is also
applied to purify the suspicious instances close to the manifold of normal examples by reconstructing using both class-shared and class-unique features to move them towards the manifold. To the best of our knowledge, the FM-Defense is the first attempt to apply disentangled learning for effective defense against oblivious adversarial attacks via detection and purification, with good interpretability and feasibility and completeness.

Our contributions are summarized as follows. We first present a fundamental intuition that adversarial examples are generally more sensitive to intrinsic feature changes than normal images. Based on this intuition, we then propose an adversarial example detection and purification method based on feature manipulation – FM-Defense. We use a Combo-VAE to manipulate the feature in an easy and interpretable manner. Besides detection, the Combo-VAE is used to purify the suspicious inputs by reconstructing images based on class-unique and class-shared components and balancing the trade-off between disentanglement and reconstruction. It can improve the completeness of the reconstructed instance for purification. We implement and evaluate the FM-Defense on three image datasets, MNIST, FMNIST, and CelebA, which show superior performance in defending against various adversarial attacks.

2 BACKGROUND AND RELATED WORK

2.1 Autoencoders and β-VAE

Autoencoders (AEs) are common deep models in unsupervised learning [10]. They aim to represent high-dimensional data through the low-dimensional latent layer, a.k.a. bottleneck vector or code. Architecturally, AEs consist of two parts, the encoder and decoder. The encoder part takes the input $x \in \mathbb{R}^d$ and maps it to $z$ (the latent variable of the bottleneck vector). The decoder tries to reconstruct the input data from $z$. The training process of AEs is to minimize the reconstruction error. Formally, we can define the encoder and the decoder as transitions $t_1$ and $t_2$

$$t_1(X) \rightarrow Z$$

$$t_2(Z) \rightarrow \hat{X}$$

$$t_1, t_2 = \arg\min_{t_1, t_2} \|X - \hat{X}\|^2.$$  

(1)

The VAEs model shares the same structure with the AEs, but is based on the assumption that the latent variables follow some kind of distribution, such as Gaussian or uniform distribution. It uses variational inference for the learning of the latent variables. In VAEs, the hypothesis is that the data is generated by a directed graphical model $p(x|z)$ and the encoder is to learn an approximation $q_b(z|x)$ to the posterior distribution $p_b(z|x)$. The VAE optimizes the variational lower bound

$$L_{VAE} = \mathbb{E}_{p_{data}(x)}[L(\theta, \phi; x)] = \mathbb{E}_{p_{data}(x)}[KL(q_b(z|x)||p_b(z))] - \mathbb{E}_{p_{data}(x)}[\log p_b(x|z)].$$

(2)

The left part is the regularization term to match the posterior of $z$ conditional on $x$, i.e., $q_b(z|x)$, to a target distribution $p_b(z)$ by the KL divergence. The right part denotes the reconstruction loss for a specific sample $x$. In a training batch, the loss can be averaged as

$$L_{VAE} = \mathbb{E}_{p_{data}(x)}[L(\theta, \phi; x)] = \mathbb{E}_{p_{data}(x)}[KL(q_b(z|x)||p_b(z))] - \mathbb{E}_{p_{data}(x)}[\log p_b(x|z)].$$

(3)

β-VAE is a modification of the VAE framework that introduces an adjustable hyperparameter $\beta$ to the original VAE objective

$$L = \mathbb{E}_{q_b}(\log p_b(x|z)) - \beta D_{KL}(q_b(z|x)||p_b(z)).$$

(4)

Well chosen values of $\beta$ (usually $\beta > 1$) result in more disentangled latent representations $z$. When $\beta = 1$, the β-VAE becomes equivalent to the original VAE framework. It was suggested that the stronger pressure for the posterior $q_b(z|x)$, to match the factorized unit Gaussian prior $p(z)$ introduced by the β-VAE objective, puts extra constraints on the implicit capacity of the latent bottleneck $z$. The higher values of $\beta$ necessary to encourage disentangling often lead to a trade-off between the fidelity of β-VAE reconstruction and the disentangled nature of its latent code $z$. This is due to the loss of information as it passes through the restricted capacity latent bottleneck $z$.

2.2 Adversarial Attacks

Evasion attacks have long been studied against machine learning classifiers [11], [12], and are practical against many types of models [13]. These evasion attacks over neural networks are referred to as adversarial examples [14]. Namely, for a given input sample $x$, the adversarial example is a sample $x'$ that is similar to $x$ (according to particular measure metrics) but so that the classifier’s decision $C(x) \neq C(x')$ [13]. A classifier can misclassify an adversarial example for two reasons. (1) The adversarial example is far from the boundary of the manifold of the task. For example, the task is a handwritten digit classification, and the adversarial example is an image containing no digits, but the classifier has no option to reject this example and is forced to output a class label. (2) The adversarial example is close to the boundary of the manifold. If the classifier poorly generalizes the manifold in the vicinity of the adversarial example, then misclassification occurs.

Let $\mathbb{U}$ be the set of all instances in the sample space. A classification function is denoted by $C$, which outputs for each instance $x \in \mathbb{U}$ a predicted class $C(x) = y$. Let $\mathcal{Y} = \{y_1, \ldots, y_i, \ldots\}$ denote the set of classes for a certain classification task. Each classification function assumes a data generation process that produces each instance $x \in \mathbb{U}$ with probability $p(x)$. Let $\mathbb{N}$ be a manifold that consists of instances that act naturally with regard to a certain classification task, following a data generation process. $\mathbb{N}$ can be approximated by a set of natural instances for a classification task [6], e.g., MNIST. The goal of the adversarial example is to find a certain perturbation on $x$ to generate an adversarial example $x' \in \mathbb{U} \setminus \mathbb{N}$ that fools a specific $C$ to misclassify, i.e., $C(x') \neq C(x)$. The adversary is assumed to have the knowledge of the original classifier but is not aware of the detector and purifier. Therefore, the goal of the adversary is only to fool the unsecured classifier.
2.3 Adversarial Defenses

To defeat these adversarial attacks against DNNs, various defending approaches have been investigated. They are typically categorized as three types: Gradient Obfuscation, Robust Optimization, and Adversarial Detection.

2.3.1 Gradient Obfuscation

As most of the DNN models are learned via gradient, gradient obfuscation defenses are applied to mask or obfuscate the gradient information to defeat the adversarial attacks. One common gradient-based defense is defensive distillation [15]. It implements distillation training methods [16] and hides the gradient between the pre-softmax layer and softmax outputs. However, [17] has demonstrated that it is feasible to bypass the defensive distillation defense via procedures: (1) adopt a more proper loss function, (2) determine the gradient directly from the pre-softmax layer, (3) strike an easy-to-attack network initially and then transfer to the distilled network. Besides, Defense-GAN [8] applied generative models to project an adversarial example onto the benign data manifold before passing to the classifier. The underlying intuition is that the cumulative product of partial derivatives from each layer will cause the gradient to be very tiny or large, restricting the attacker from precisely predicting the adversarial examples’ location. The generative models are considered as a purifier, added before the classifier. However, such an additional network will make the final deep classifier to be greatly deep.

2.3.2 Robust Optimization

Robust optimization strategies are applied to enhance the robustness of DNN classifier via relearning its parameters. One idea of robust optimization is to train a better classifier via adversarial training [18]. A robust classifier is learned by considering adversarial information during training. A mixture of normal and adversarial examples are combined in training set [14], or mix the adversarial objective with the classification objective as regularizer [19], to build the robust classifier. Although this idea is promising, it is not easy to reason about what attacks to train on and how important the adversarial component should be. Currently, these questions are still unanswered. Other robust optimization strategies contain altering the underlying architecture or learning procedure, e.g., adding more layers, ensemble/adversarial training, or changing the loss/activation functions. For example, citepang2019improving applies adaptive diversity promoting regularizer to encourage diversity, leading to globally better robustness for the ensemble by making adversarial examples difficult to transfer among individual members. An adversarial defense by restricting the hidden space is proposed in [20]. They conduct a constraint to force the network to learn distinct and distant decision regions for each class, aiming to enhance the robustness of learned models. Namely, the features for each class are forced to lie inside a convex polytope that is maximally separated from the polytopes of other classes. Such constraints could even defeat against the strongest white-box attacks, without degrading the classification performance on clean images.

2.3.3 Adversarial Detection

Adversarial detection techniques distinguish whether the input is benign or adversarial first and then reject (or/and purify) suspicious inputs. The detection-based defense against adversarial examples for a classifier \( C \) aims to establish a detector \( d_C : \mathcal{U} \rightarrow \mathcal{Y} \cup \{J\} \). \( J \) is the judgment that the input is unlikely from the manifold of normal/clean instances. Additionally, the purification-based defense could also be to build a purifier \( p : \mathcal{U} \setminus \mathcal{N} \rightarrow \mathcal{N} \) to reconstruct suspicious instances with small distortion only using some essential features, in order to move adversarial examples towards the manifold of normal/clean examples. For example, such a defense is to detect adversarial examples with hand-crafted statistical features [3] or separate classification networks [4]. For each attack generating method considered, it constructs a DNN classifier (detector) to tell whether the input is normal or adversarial. The detector was directly trained on both normal and adversarial examples. The detector showed good performance when the training and testing attack examples were generated from the same process, and the perturbation was large enough, but it did not generalize well across different attack parameters and attack generation processes. [21] detects the adversarial examples by checking the impact of the training data on the network decision using influence functions and k-nearest neighbor (k-NN) classifier. However, the impact of every training sample should be evaluated on a trust validation set data. MagNet [6] is proposed to defend adversarial attacks by two defense procedures: (1) detecting the input as an adversarial example or a normal image while rejecting suspicious instances with large distortion, and (2) purifying suspicious instances with small distortion by reconstruction. MagNet randomly picks two autoencoders from a repository: one for the detection layer and the other for the reformer layer. Even applying randomness for selecting both autoencoders, [17] has demonstrated that adversarial images can evade MagNet. The reconstruction error-based detection could lose effectiveness when the perturbed images are derived from oblivious attacks (such as the CW attack) that are very likely similar to normal images.

2.3.4 Detection by Feature Manipulation

Generally, the vulnerability of DNNs is because that they learn low-level abstractive features, such as the small edge or gray-scale value, easily influenced by the pixel-wise perturbation. On the other hand, humans generally learn high-level and semantic features, such as the shape of an object, which are robust to tiny perturbation [9], [22], and are therefore not as easily fooled. It is feasible to identify high-level semantic features from the clean input and perform detection based solely on these semantic features. Therefore, high-level feature-based detection has an advantage over other defenses in nature. FBGAN [9] extracts the input images’ semantic features and reconstructs the denoised images from these features. It uses Bidirectional GAN’s generative capability and the mutual information (MI) regularization between all latent codes and the generated images for distinguishment. Image Super-Resolution [23] is used as an image enhancement approach to provide a defense mechanism to mitigate the effect of such adversarial perturbations. Image Transformation-based detection [5] applies certain transformation
features using a manifold that consists of instances that act naturally w.r.t. a certain classification task.

\( X = \{ x_1, x_2, \ldots, x_n \} \) inputs in \( U \)

\( Y = \{ y_1, y_2, \ldots, y_m \} \) true labels of \( x \)

\( C \) classification function

\( x^* \) adversarial example of \( x \)

\( E, q_\theta(z|x) \) encoder \( E : X \rightarrow Z \)

\( D, p_\psi(x|z) \) decoder \( D : Z \rightarrow X' \)

\( z_e = \{ f_1, f_2, \ldots, f_n \} \) latent codes for \( x \)

\( r^{(i)} \in \mathbb{R} \) resistance indicator for latent code \( f_i \)

\( \theta^{(i)} \in \mathbb{R} \) threshold for \( r^{(i)} \)

\( \text{Dis}(\vec{r}, \vec{\theta}) \) distance between the resistance vector of an instance and the resistance threshold vector

\( \theta_{TC} \) Total Correlation to encourage independence in the latent codes \( z \)

\( z_u, z_s \) class-unique and commonly shared across all classes representations

\( L_{KL}, L_R, L_{TC} \) KL regularization term, the reconstruction error and TC term

\( \rho, \eta \) false-positive rate indicator, the distance selection indicator

\( \| a - b \|^2 \) the euclidean distance for \( x \) and \( y \)

operations on an image to generate several transformed images. Then the classification results of these transformed images are used to distinguish between normal and the adversarial. However, a mixture of normal and adversarial examples is required in the training set to train these detectors. This causes the high computational cost to generate adversarial examples and adversaries classifiers, particularly for more complex datasets and stronger attacks. Furthermore, the impact of external feature transformation, such as image rotation and shifting, is not consistent and general for different instances. Besides, the background of an image adds a large number of extra features to the object, which is also sensitive to the external transformation. [24] extensively investigates the detection of adversarial examples via image transformation. They demonstrate that image transformation based detection might not have a good detection performance (also due to the accuracy degradation), it is still possible for the attackers to construct AEs that are robust to input transformations.

### 3 Feature Manipulation as Defense

#### 3.1 Research Questions

An image may be difficult to recognize from the static perspective (such as a human portrait with odd expressions). Its readability could be enhanced from the dynamic perspective via implementing specific image transformation operations (such as a video clip for the same person with different expressions). Our intuition is that the essence of a given image is generally resistant to non-significant intrinsic feature transformations, e.g., varying the thickness of handwritten digits or the facial expression. This paper aims to demonstrate this intuition works for deep learning models when faced with adversarial examples. Adversarial examples fool the image classification model by adding tiny perturbations at some specific positions to push the image away from the right decision boundary and impact the prediction results negatively. Such unstructured perturbation designed for a single image may cause different impacts after different feature transformations. Namely, feature transformations are likely to cause a shift of classification results for adversarial examples while leaving no significant impact on the normal. Consequently, we propose Feature Manipulation based defense (FM-Defense), a framework for defending adversarial examples based on our intuition. This framework aims to address the following research questions:

- **RQ1**: How to make the feature transformations feasible in an interpretable manner?
- **RQ2**: How to recognize adversarial examples by feature transformations efficiently?
- **RQ3**: Is it possible to use the feature transformations of malicious as the purified version?

#### 3.2 Methodology Overview

To address the feasibility of feature transformations (RQ1), we conduct feature transformations via the disentangled latent codes manipulation, and apply the S1: Representation with improved disentanglement strategy using Combo-VAE (detailed in Section 3.3). To recognize adversarial examples efficiently based on feature transformations (RQ2), we propose S2: Detector with fine discrimination ability strategy using resistance evaluation (detailed in Section 3.4). To answer the RQ3, we apply the S3: Purifier with comprehensive reconstruction ability strategy to exploit the possibility of using the feature transformations of malicious as the purified version (detailed in Section 3.5). Next, we summarize the scheme of our defense in terms of these strategies. To formalize the process of our defense, we summarize some notations in Table 1.

**S1: Representation with improved disentanglement.** We initially train a disentangle representation model, Combo-VAE, enhanced from \( \beta \)-VAE [25], on a clean dataset. The objective here is to make disentangled latent codes with semantic meaning so that they are easy to control. Such models consist of two components: the encoder \( E : X \rightarrow Z \) takes the input \( x \in \mathbb{R}^d \) and maps it to \( z \) (the latent variable of the bottleneck vector). The decoder \( D : Z \rightarrow X' \) tries to reconstruct the input data from \( z \). The encoder can map a high-dimensional input instance \( x \) to disentangled low-dimensional latent codes \( z \), i.e., one latent code can only control one certain feature. The decoder is used to reconstruct the input from the low-dimensional latent code \( z \). For simple images, e.g., \( 28 \times 28 \) handwritten digits, the latent code can be a \( m \)-dimensional vector. For complex images, e.g., \( 128 \times 128 \) face images, the latent codes are \( ch \) channels of \( w \times h \) feature maps. We find that some \( w \times h \)-dimensions feature maps can reveal some disentangled semantic features as well. Therefore, each channel of \( w \times h \) feature map can be considered as one latent factor used for manipulation for simplicity. Namely, each element of the \( w \times h \) feature map is simultaneously changed at the same scale.

---

**Table 1**

| Notation | Explanation |
|----------|-------------|
| \( U \) | the set of all instances in the sample space |
| \( N \) | a manifold that consists of instances that act naturally w.r.t. a certain classification task |
| \( X = \{ x_1, x_2, \ldots, x_n \} \) | inputs in \( U \) |
| \( Y = \{ y_1, y_2, \ldots, y_m \} \) | true labels of \( x \) |
| \( C \) | classification function |
| \( x^* \) | adversarial example of \( x \) |
| \( E, q_\theta(z|x) \) | encoder \( E : X \rightarrow Z \) |
| \( D, p_\psi(x|z) \) | decoder \( D : Z \rightarrow X' \) |
| \( z_e = \{ f_1, f_2, \ldots, f_n \} \) | latent codes for \( x \) |
| \( r^{(i)} \in \mathbb{R} \) | resistance indicator for latent code \( f_i \) |
| \( \theta^{(i)} \in \mathbb{R} \) | threshold for \( r^{(i)} \) |
| \( \text{Dis}(\vec{r}, \vec{\theta}) \) | distance between the resistance vector of an instance and the resistance threshold vector |
| \( \theta_{TC} \) | Total Correlation to encourage independence in the latent codes \( z \) |
| \( z_u, z_s \) | class-unique and commonly shared across all classes representations |
| \( L_{KL}, L_R, L_{TC} \) | KL regularization term, the reconstruction error and TC term |
| \( \rho, \eta \) | false-positive rate indicator, the distance selection indicator |
| \( \| a - b \|^2 \) | the euclidean distance for \( x \) and \( y \) |
The Combo-VAE is used to extract the disentangled latent codes for each instance so that it is feasible to select and manipulate a number of codes that reveal the desired semantic features, e.g., thickness of the digit. The feasibility of feature manipulation is related to the disentanglement level. Therefore, strategies are used to improve the disentanglement, as described in the following sections.

S2: Detector with fine discrimination ability. Given an instance, we first vary a latent code \( f \) for \( n \) times to obtain \( n \) morphs reconstructed by the decoder. We then record the ratio of unchanged classification prediction by applying a certain classifier (to be protected) on these \( n \) morphs compared with the original prediction. The ratio is used as a resistance indicator \( r^{(i)} \) for code \( f \). At one time, we change one of \( m \) randomly selected latent codes in turn and obtain a \( m \)-dimensional resistance vector \( \overrightarrow{r} \) for each instance. We find that normal instances' resistance ability is significantly better than that of adversarial ones, as shown in the histogram of Fig. 1. Therefore, a \( m \)-dimensional threshold configuration \( \theta_r \) for all selected \( m \) latent codes can be decided on the normal instance to distinguish normal and adversarial instances. An instance that meets \( r^{(i)} > \theta_r^{(i)} \), \( \forall r^{(i)} \in \overrightarrow{r} \) will be recognized as normal. Otherwise, it is recognized as suspicious.

S3: Purifier with comprehensive reconstruction ability. It is feasible to decide a threshold that can achieve nearly 100 percent adversarial detection accuracy (True Positive). However, this will cause a large number of normal instances to be improperly recognized as adversarial, i.e., high False Positive ratio. We assume there exists a manifold of resistance on clean instances. Therefore, we use another threshold \( \theta_d \) over the distance \( \text{Dis}(.) \) between the resistance vector of an instance \( \overrightarrow{r} \) and the resistance threshold vector \( \overrightarrow{\theta_r} \), to build a salvageable set that consists of suspicious instances close to the manifold of normal instances

\[
\text{Dis}(\overrightarrow{r}, \overrightarrow{\theta_r}) = \sum |r^{(i)} - \theta_r^{(i)}|, \quad \forall r^{(i)} < \theta_r^{(i)}.
\]

As illustrated in Fig. 3, if a suspicious instance from the detection has a distance of more than \( \theta_d \), then it will be rejected as adversarial. Otherwise, it is incorporated in a salvageable set that will be reconstructed by a Combo-VAE based purifier. The purifier moves suspicious examples in the set towards the normal manifold before feeding to the target classifier. To improve the reconstructed instance’s quality, we enhance the Combo-VAE to absorb both significant class-unique features and class-shared features. Details of these components are given in the following sections.

3.3 Representation With Improved Disentanglement

VAE-based autoencoders and their variations are commonly applied for disentanglement learning. Specifically, the encoder \( E \), parameterized by \( q_\phi(z|x) \), is trained to convert high-dimensional data \( x \) into the latent representation bottleneck vector \( z \) in the latent space that follows a specific Gaussian distribution \( p(z) \sim N(0, 1) \). The decoder \( p_\theta(x|z) \) is trained to reconstruct the latent vector \( z \) to \( x \) and decoder are trained simultaneously based on the negative reconstruction error and the regularization term, i.e., Kullback-Leibler (KL) divergence between \( q_\phi(z|x) \) and \( p(z) \). The regularization term is used to regularize the distribution \( q_\phi(z|x) \) to be Gaussian distribution whose mean \( \mu \) and diagonal covariance \( \Sigma \) are the output of the encoder. We apply a Combo-VAE to get good disentanglement in \( z \) by improving the inner-independence of latent codes. Specifically, Total Correlation (TC) [26] is used to encourage independence in the latent vector \( z \), as given in Equation (2).

\[
TC(z) = KL(q(z)||\overline{q}(z)) = EQ(z) \left[ \log \frac{q(z)}{\overline{q}(z)} \right].
\]

As TC is hard to obtain, the approximate tricks used in [27] is applied to estimate TC. Specifically, a discriminator \( D_c \) is applied to classify between samples from \( q(z) \) and \( \overline{q}(z) \). Thus, learning to approximate the density ratio is needed for estimating TC [27], \( D_{ov} \), parameterized by \( \nu \), is jointly trained with other components. The TC term is replaced by the discriminator-based approximation as follows:

\[
TC(z) \approx EQ(z) \left[ \log \frac{D(z)}{1 - D(z)} \right].
\]

The objective of Combo-VAE is augmented with a TC [26] term to encourage independence in the latent factor distribution as follows:

\[
E_{q_\phi(x|z)} \left[ \log p_\theta(x|z) \right] - KL(q_\phi(z|x)||p(z)) - \gamma L_{TC}.
\]

Note that this is also a lower bound on the marginal log-likelihood \( E_{q_\phi(x|z)} \left[ \log p_\theta(x|z) \right] \). The first part reveals the reconstruction error, denoted by \( L_R \), evaluating whether the latent bottleneck vector \( z \) is informative enough to recover the original instance. \( L_R \) can be defined as the \( l_1 \) loss between the original instance and the reconstructed instance. The second part is a regularization term, denoted by \( L_{KL} \), to push \( q_\phi(z|x) \) to match the prior distribution \( p(z) \). The third part is the TC term, denoted by \( L_{TC} \), to measure the dependence of multiple random variables.

As shown in Fig. 2, the parameter \( \phi \) of encoder \( q_\phi(z|x) \) is then trained by \( L_{KL} + L_R + \gamma L_{TC} \). The parameter \( \theta \) of decoder is updated in terms of \( -\nabla_\theta L_R \). The parameter \( \nu \) of TC-discriminator is updated in terms of \( -\nabla_\nu \log(1 - D_{ov}(\text{permutated}(z^{(i)}))) \). Here, the permuted channel function is to random permute on a sample in the batch for each dimension of its \( z \), similar to [27].

3.4 Detector With Fine Discrimination

The indicator for adversarial detection should easily differentiate normal and adversarial instances, be feasible and stable to conduct, and attack-agnostic. The classification accuracy resistance over the morphs, derived from feature manipulation by changing a certain latent code, can meet these criteria. The detector’s discrimination ability depends on the ability to reduce the false-positive ratio (normal instances to be recognized as adversarial) and the naturality
of the morphs. Consequently, we apply two strategies: normal value range selection and natural morph generation.

**Natural Morph Generation.** The initial step is to find the normal value range of each code on the clean validation set. Then the morphs are produced via manipulating each code within its normal value range. As the latent codes are disentangled, independent (all from $N(0, 1)$) and have semantic meaning, some latent codes that reveal nonsignificant intrinsic features (e.g., thickness for handwritten digits) will be selected and their normal range can be decided empirically in an interpretable manner for human on a validation set. We can incrementally add/reduce a fixed value to the original learned latent codes within the normal range to obtain the morphs by feature manipulation. However, the modified latent vector may not be on the manifold of normal instances. If that happens, an unnatural instance will be reconstructed by the decoder. Hence, we conduct an iterative stochastic search to make the morphs on the manifold by adding natural noise. Specifically, we increase the search range by $\Delta r$ within which the perturbation for a certain latent code $\Delta z_i$ is randomly sampled ($B$ samples for each iteration) until we produce $N$ natural latent code with the value in the normal value range to reconstruct $N$ natural morphs. We then evaluate the resistance on classification for this latent code using the targeted classifier. Iterative, we can get a $m$-dimensional resistance evaluation vector for each instance.

**Automatic Threshold Selection Strategies.** Given a pre-trained classifier for detection, we decide a resistance threshold for each latent code on a validation set containing clean instances only, without requiring the prior knowledge of attacks. The threshold of resistance could be simply decided for each factor so that the false-positive rate on the validation set is below a pre-defined value $1 - \rho$ (i.e., more than $\rho\%$ clean instance are correctly recognized). Therefore, we can decide a unified and fixed resistance threshold for all latent codes or formulate a specific threshold for each latent code in terms of $\rho = 1 - s^*$, i.e., more than $\rho\%$ clean instance are correctly recognized.”

**3.5 Purifier With Comprehensive Reconstruction**

We assume that suspicious instances with a small distance between their resistance vector and the resistance threshold vector can be considered as close to the manifold of the normal instances. Namely, the distance is under the $\theta_d$, which is set as the $\eta\%$ fractile on clean validation data. Therefore, a VAE-based purifier is used to reconstruct and move them towards the manifold of normal examples. In regular VAEs, the prior over the latent variables is commonly isotropic Gaussian, resulting in limited representation because the learned representation can only be unimodal and does not allow for more complex representations [28]. Consequently, regular VAE-based disentangled learning generally learns some class-shared information and some essential class-unique information lost. This limitation causes only the suspicious instances with tiny perturbations, while suspicious instances with larger perturbations, e.g., derived from FGSM attack in Fig. 6, can not be reconstructed correctly. To address the completeness and accuracy of the VAE-based purifier, we enhance the Combo-VAE by incorporating class-relevant conditional information to guide the reconstruction. Our latent codes consist of class-unique representation, $z_u$ (e.g., essential features unique to each digit), and commonly shared across all classes, $z_s$ (e.g., the thickness of handwriting digits). As shown in Fig. 2, the enhanced Combo-VAE has a similar scheme to a VAE, but instead of using exclusively the same data for the input and output of the network, we use class-unique additional information as an extra input to the decoder.

Expressly, we assume the observed instances are derived from a mixture of Gaussians, i.e., the inference of the class of an instance is equivalent to inferring which mode of the latent codes $z_u$ of the data point was generated from. Namely, we use a mixture of Gaussians as our prior for $z_u$, used as conditional information for training decoder of VAE. For each class label $c$, we assume it has $K$ features such as different writing styles for handwritten digits, namely $K$-dimensional $z_u$. Therefore, we first train a DNN using Gaussian Mixture loss (DGMM) that maps the input $x$ to $z_u$ that is learned with the supervision of the categorical class label $c$. $z_u$ reveals $K$
features for the label of a given input instance. The DGMM is solely trained using the clean labeled instances, and the output is a logistic regression on the latent representation of \( K \) features as a classification task.

Each feature \( z^{(k)}_u \) follows a mixture of \( K \) Gaussian distribution with learned mean \( \mu_c \) and covariance \( \Sigma_c \) for each class \( c \) given by neural networks of DGMM with parameters \( \kappa_\mu \) and \( \kappa_\Sigma \) respectively

\[
p(z^{(k)}_u) = \sum_c N(c; \mu_c, \Sigma_c)p(c). \tag{9}
\]

Here, \( p(c) \) is the prior probability of class \( c \). The loss of DGMM is calculated as the cross-entropy between the posterior probability \( q(c|z_u) \) and the corresponding one-hot class labels, denoted \( L_{\text{cls}} \), combined with a likelihood regularization term to force the training samples to obey the assumed GM distribution, denoted \( L_{\text{ld}} \) \cite{28}, \cite{29}. \( L_{\text{cls}} \) can let \( z_u \) contain as much label information as possible, as the MI between \( z_u \) and class \( c \) are added to the maximization objective function

\[
L_{\text{cls}} = -E_{q_k}(z_u|x) \sum_c I(c = y) \log q(c|z_u) = -\log \frac{N(z_u; \mu_y, \Sigma_y)p(y)}{\sum_k N(z_u; \mu_k, \Sigma_k)p(k)}. \tag{10}
\]

\( L_{\text{ld}} \) is applied to measure how trained the model is to classify the data. In the setting of Equation (7) when \( p(c) \) is simply set to \( 1/C \) for all classes. The \( L_{\text{ld}} \) for a given class \( c \) is given as follows:

\[
L_{\text{ld}} = -\log N(z_u; \mu_c, \Sigma_c). \tag{11}
\]

Consequently, the loss for DGMM is

\[
L_{\text{GM}} = L_{\text{cls}} + \lambda_{\text{ld}} L_{\text{ld}}. \tag{12}
\]

where \( \lambda \) is a non-negative weighting coefficient.

In addition, an encoder \( E_x \) is trained to map input \( x \) to \( z_u \) where each code is forced to follow the standard Gaussian \( N(0, I) \), implemented completely by the \( q_k(z|x) \) in Section Representation with improved disentanglement. The input of a pre-trained DGMM can be any instance \( x \) without a label, since DGMM will output the \( K \)-dimensional feature vector of the most likely class of \( x \). The latent codes \( z_u \) and \( K \)-dimensional \( z_u \) derived from the pre-trained DGMM are then simply concatenated together to a decoder to reconstruct the input \( x \). The loss of decoder is used to measure how probable it is to generate \( x \) by using the distribution \( p(x|z_u, z_s) \), that is, is a distance between \( x \) and reconstructed \( x' \).

The training of Combo-VAE is two stages. Initially, the DGMM (modeled by \( q_k \)) is updated using \( L_{\text{GM}} \) to learn mean \( \mu_c \) and covariance \( \Sigma_c \) of the prior \( p(z_u|c) \), encouraging \( z_u \) to be label dependent and follows a learned Gaussian mixture distribution. In the second step, the encoder \( E_x \) and the decoder are trained jointly to reconstruct images with concatenated \( z_u \) and \( z_s \). The Encoder \( E_x \) (modeled by \( q_k \)) is intended to extract class-shared code \( z_s \). It is trained by \( L_{\text{ld}} \) and \( L_R \) to make \( z_s \) be close to \( N(0, 1) \). Then the decoder \( p_\theta \) generates a reconstruction image using the combined feature of \( z_s \) and \( z_u \) with the loss \( L_R \).

In case that the training dataset is contaminated, we add a density estimation \cite{30}, \cite{31} regularizer \( L_A \) into the objective function in Equation (8), to the estimated anomaly score of the learned low-dimensional embeddings. \( L_A \) is given as follows:

\[
L_A = \frac{1}{N} \sum_{i=1}^{N} E(z'_i), \quad z'_i = \text{concatenate}(z, L_R)
\]

\[
E(z'_i) = -\log \left( \sum_{k=1}^{K} \alpha_k \frac{\exp(-0.5(z'_i - \mu_k)^T \Sigma_k^{-1}(z'_i - \mu_k))}{\sqrt{2\pi \Sigma_k}} \right).
\tag{13}
\]

Here, \( \alpha_k, \mu_k \) and \( \Sigma_k \) are mixture probability, mean, covariance for component \( k \) in GMM, respectively.

4 Experiments

The performance of FM-Defense is evaluated against the state-of-the-art adversarial attacks on two datasets: MNIST \cite{32}, FMNIST (2D shapes) \cite{33}, CelebA \cite{34} (Face) and ImageNet-1000 \cite{35}. The adversaries are assumed to have no knowledge about the detector and purifier (black-box setting). They only focus on generating adversarial examples that aim to maximize the prediction errors on a target classifier and do not care which class the victim classifier outputs to as long as it is different from the ground truth.

4.1 Setups

We represent the amount of data needed to train the combo-VAE correctly in terms of different data types. For end-to-end training, we train the one-off Combo-VAE on 20000 clean examples for MNIST, FMNIST, and 30000 for CelebA. We use a batch size of 64 for all data sets. After 50000 iterations, it is feasible to obtain apparent disentangled features for MNIST and FMNIST, such as shape, thickness, azimuth and stretching, etc. For CelebA, we can capture the apparent disentangled features such as facial expression after 100000 iterations. We randomly select 5,000 clean images (named CLE, labeled 0) and the corresponding 5,000 adversarial examples (named ADV, labeled 1), respectively. These datasets are used to test the efficiency of the FM-Defense. Besides, another 2,000 clean instances are chosen as the validation data (named VAL) to decide the thresholds. We trained a classifier for MNIST using the setting in \cite{17} with an accuracy of 99.4 percent. We train a classifier using the setting in \cite{36} for FMNIST. For the CelebA, we train an identification classifier using the setting in \cite{37} with an accuracy of 94.7 percent. We normalized the data between 0 and 1 instead of [0, 255] for simplicity. Table 1 shows the architectures of the Combo-VAE for MNIST, FMNIST, and CelebA.

We use the fast gradient sign method (e.g., FGSM \( \epsilon = 0.3, L^\infty \) \cite{38}), DeepFool \( L^\infty \) \cite{17}, \cite{39}, and CW \( L^2 \) attack \cite{17}, \cite{39} for the experiments, implemented using Fooling box \cite{40}. We select 20-dimensional latent codes to manipulate MNIST and 10-dimensional for FMNIST. The face image will be mapped into 256 channels of 64*64 feature maps. Each channel is considered as one latent factor. We generate 100 morphs for each latent factor of the
instance by varying each latent code 100 times within the normal value range. The morphs were fed into the target classifier to calculate the resistance vector for each instance.

For MNIST and FMNIST, we respectively, decide the resistance threshold vector, such that the false-positive rate of the detector on the validation set VAL is at most 0.001 for all selected latent codes. This means that each detector mistakenly rejects no more than 0.1 percent clean instances in the validation set, i.e., \( \rho = 99.9\% \). Besides, distance threshold values are decided for MNIST and FMNIST, respectively, set as the \( \eta = 90\% \) fractile on clean validation data in our experiments. Other default hyper-parameters are given as follows: \( \gamma = 40 \), \( \lambda_{\text{det}} = 0.1 \). For the face images of CelebA, we find that only one randomly selected 64*64 feature map as a latent factor can achieve well-detecting performance, approximately 100 percent, even without the TC-optimization mentioned in Section 2.3. Therefore, a basic Combo-VAE without optimization strategies is used for face images and only one latent factor (64*64 feature map) is used for detection.

### 4.2 Classification-Based Detector for Comparison

The threshold-based detection could be considered a semisupervised method, where only a clean validation dataset is necessary to empirically decide a threshold for detection without adversarial examples and extra training. Besides, we also compare the threshold-based detection with a supervised classification-based detector. Specifically, we select a set of clean images that are correctly recognized (CleanC), and the corresponding adversarial examples (AdvC) derived from the different attacks algorithm that can successfully attack the images in the training set of CleanC. The clean images are labeled 0, and their adversarial examples are labeled 1. Next, we apply our feature transformation to each image of both datasets. Specifically, we randomly select ten latent codes to modify 100 times, resulting in a total of \( 10 \times 100 = 1000 \) morphs of each image. All 100 morphs are then fed into a pre-trained classifier and record the logits (output of the DNN before the output layer). The concatenated logits of all 1000 morphs represent each image and the training instances for the classification-based detector. Multi-layer perceptrons are used as the detector network (two layers for MNIST, three for CelebA, and five for high-fidelity images).

### 4.3 Defense Evaluation Against Adversarial Attacks

#### 4.3.1 Detection Accuracy Evaluation

We first evaluate the performance of the detector for varying resistance thresholds. The overall results on MNIST are shown in the first row of Fig. 4 in terms of adversarial, clean and overall detection accuracy. Here, we test on both CLE and ADV datasets, respectively. The correct decision is that adversarial examples are recognized as 1 while normal ones as 0. Initially, we use a unified resistance threshold for all selected latent codes. The adversarial detection accuracy (legend ADV) is the proportion of adversarial instances in ADV to be recognized as adversarial, i.e., True-Positive. The clean detection accuracy (legend Clean) is the proportion of clean instances in CLE to be recognized as clean, i.e., True-Negative. The overall detection accuracy (legend Overall) is the proportion of all correctly detected instances in both ADV and CLE. We observe that even for a small resistance threshold, e.g., 5, it can detect more than 75 percent adversarial examples and retaining 81 percent clean instances correctly labeled. The adversarial detection accuracy of FM-Defense on ADV is above 99.9 percent for both MNIST and FMNIST on all the potential attacks, including CW attack (99.9 percent adversarial detection accuracy at the united resistance threshold 85 with 13 percent correctly recognized clean instances). Note that we achieved such high accuracy without any adversarial examples required and only based on threshold vectors that are easy to be decided experimentally. However, the False-Positive (FP) rate of normal instances increases as the resistance threshold and adversarial detection accuracy rise. The major concern here is how to balance the adversarial detection accuracy and the FP rate. Combining the correctly recognized normal instances and adversarial instances together, the overall detection accuracy reaches the best at nearly 80 percent on the threshold at 30, which can be used as a balanced threshold, with 85-93 percent adversarial detection accuracy and 75 percent clean detection accuracy. These results demonstrate that our method can efficiently thwart adversarial examples while achieving an acceptable FP rate. The PGD attack evaluation, stronger than FGSM while having lower variance than DeepFool and CW, also confirms the previously mentioned findings.

We also demonstrate the impact of choosing different threshold selection strategies, i.e., varying the \( \rho \) to decide a specific resistance threshold for each latent code and selecting different codes. As shown in the second row of Fig. 4, both adversarial detection accuracy and FP rate (1-clean detection accuracy) increase as \( \rho \) rises. Hence, it is feasible to decide a \( \rho \) that can balance the adversarial detection accuracy and FP rate. Overall, the specific resistance threshold strategy performs better than the unified threshold, with relatively high adversarial detection accuracy. Besides, we find that selecting different latent codes affects the detection accuracy. As shown in the third row of Fig. 4, selecting latent codes for FM-Defense against the CW attack over MNIST has various discriminatory abilities. It is feasible to decide an efficient code selection configuration experimentally. The third row of Fig. 4 shows results on the FMNIST(2D) dataset, which confirms the findings mentioned above. The last row shows the resistance statistics on face images. For intricate images, it is interesting to find that the clean images are more resistant to the latent factor change within a certain value range, reaching 100 percent. In contrast, the adversarial ones are more sensitive to such change, totally below 80 percent. Therefore, it is feasible to use a higher resistance, e.g., 90 percent, to detect all adversarial examples (detection accuracy = 100% and FP rate = 0).

We further compare the threshold-based detection with a supervised classification-based detector. We also evaluate the correlation between the number of available adversarial images with the quality of detection. We implement a MLP based binary classification to automatically detect malicious inputs with various size of adversarial images (0, 2000, 3000, 5000), as described in Section 4.2. As shown in Fig. 5, the detection accuracy increases as the size of labeled adversarial examples in the training set. Compared to binary classification, the threshold parameters that are empirically set also obtain good accuracy, without extra training. Since only a clean validation
A dataset is necessary to empirically decide a threshold for detection without adversarial examples and extra training. Consequently, threshold-based detection, as a semisupervised method, is highly recommended to use.

We also evaluate the detection performance on the Imagenet dataset, aiming at investigating multi-domain and more complicated images. As shown in the last subfigure in Fig. 4c, the recognition accuracy for both adversarial and clean samples are both above 90 percent for FGSM attack, above 85 percent for PGD attack and around 75 percent for CW attack. For the malicious samples in the salvageable set, almost 100 percent of them could be purified after reconstruction. However, we also find that the defense performance is limited to the autoencoder’s generative ability, especially when the category of data is large and the fidelity.

4.3.2 Purification Accuracy Evaluation

We then evaluate the purification performance of our approach on the detected suspicious instances. First, we
calculate the distance between the resistance vector of a given instance and the resistance threshold vector to establish a salvageable set for purification. The set includes only adversarial examples and misclassified clean instances.

We measure the recall in Fig. 5, i.e., the proportion of detected suspicious instances in the salvageable set that gained the correct labels after reconstruction, when varying the distance threshold \( \theta_d \) (set at the fractile of \( \eta \), i.e., how many salvageable instances are purified) as well as resistant threshold \( \theta_r \) (the unified threshold for simplicity). There are three types of recall: clean recall, adversarial recall, and overall recall to achieve a manifold. Salvageable adversarial examples towards the normal can effectively move the misclassified clean instances and the purifier can efficiently recognize adversarial examples (2D) demonstrate the same conclusions, as shown in the second row of Fig. 5. The last row illustrates the performance of FM-Defense, as DGMM is used to capture more class-unique features on clean instances compared with Factor-VAE. The performance of our Combo-VAE on reconstructing adversarial examples is much better than Factor-VAE, as DGMM is used to capture more class-relevant information that can guide reconstruction. Therefore, our defense has better generalization ability and can purify more plentiful adversarial examples instead of only these small distortions.

4.3.4 Overall Evaluation With Comparisons

Table 2 shows the effects of defense against different adversarial attacks on the MNIST and CelebA, compared with the state-of-the-art defenses, e.g., MagNet, Defense-GAN and FBGAN. On clean MNIST, without FM-Defense, the accuracy of the classifier is 99.4 percent; with FM-Defense, the accuracy is reduced to 98.3 percent. This small reduction is negligible. As illustrated in the table, the defense performance of FM-Defense with maximum defense ability of MagNet, and Defense-GAN. Table 2 shows the performances of defense methods on the MNIST and the FMNIST datasets, respectively. As shown, the performance of FM-Defense exceeds that of MagNet against all oblivious attacks (DeepFool and CW). FM-Defense also outperforms FB-GAN and Defense-GAN against the FGSM attack. Overall, FM-Defense shows the best performance against all evaluated attacks. These evaluations provide empirical evidence that FM-Defense is effective, easy to conduct and generalizes well to different attacks.

4.4 Data Amount and Cost Discussion

Like other generative models, the success of the Combo-VAE comes at the cost of both computation and data. They are generally data-hungry (multimillion images) and rely on large volumes of diverse and high-quality training examples. One promising solution to address this issue is to adopt data augmentation approaches, e.g., the DiffAugment method. DiffAugment showed state-of-the-art performance on popular generative benchmarks using only 100 samples. DiffAugment also equaled the top performance on CIFAR-10 and CIFAR-100 corpora using only 20 percent training data without any pre-training [41]. We examine various limited data scenarios to demonstrate the feasibility of such augmentation, considering 1k, 5k, 10k, and 30k training images available for the CelebA case. As we can see from the Fig. 8, the Frechet Inception Distance (FID) [42] of the generated samples is obviously enhanced after the augmentation under all data percentage settings.

The time complexity of the combo-VAE seems to be increased by these strategies of disentanglement and quality enhancement. However, our method could be trained once and used to continuously detect malicious inputs, including those derived from unknown attacks. Only a clean validation dataset is needed for the threshold-based detector to empirically decide a threshold with good detection accuracy instead of generating various adversarial examples or extra training of a classifier. This will avoid enormous computing complexity, time cost, and extensive labeling efforts. Additionally, one feasible solution to further reduce the time cost is to adopt transfer learning. For example, pre-trained models (pre-trained VAE or CNN) can be used as the starting point to train our combo-VAE. Besides, the knowledge learned from a combo-VAE trained on a specific domain could also be used for the same relevant scenarios via finetuning with a small amount of new data.
This paper proposes FM-Defense, a one-off and attack-agnostic defense that effectively detects and purifies state-of-the-art adversarial attacks. Our defense assumes that the perturbation lacks of transferability. FM-Defense applies a Combo-VAE for both adversarial detection and salvageable instance purification. FM-Defense achieves high accuracy against the state-of-the-art attacks, especially for complex images, delivering empirical evidence that our assumptions are likely correct. The Combo-VAE with two enhanced encoders is used to reconstruct salvageable candidates for purification. However, before we discover stronger justification or proof, it is not appropriate to dismiss the possibility that the good performance of FM-Defense is attributed to

5 CONCLUSION

This paper proposes FM-Defense, a one-off and attack-agnostic defense that effectively detects and purifies state-of-the-art adversarial attacks. Our defense assumes that the perturbation lacks of transferability. FM-Defense applies a Combo-VAE for both adversarial detection and salvageable instance purification. FM-Defense achieves high accuracy against the state-of-the-art attacks, especially for complex images, delivering empirical evidence that our assumptions are likely correct. The Combo-VAE with two enhanced encoders is used to reconstruct salvageable candidates for purification. However, before we discover stronger justification or proof, it is not appropriate to dismiss the possibility that the good performance of FM-Defense is attributed to

![Graphs showing purification performance evaluations.](image)

Fig. 6. Purification performance evaluations against adversarial attacks.

### TABLE 2
The Network Structures

| Encoder/DGMM | Face Encoder | Decoder | Face Encoder |
|--------------|--------------|---------|--------------|
| C.ReLU N32,K4,S2 | C.ReLU N64,K7,S2 | Dense.ReLU 128/512 | RSB. N512,K3,S1 |
| C.ReLU N32,K4,S2 | C.ReLU N128,K3,S2 | Dense.ReLU N64,K4 | RSB. N512,K3,S1 |
| C.ReLU N64,K4,S2 | C.ReLU N256,K3,S2 | C.ReLU N64,K4,S2 | RSB. N512,K3,S1 |
| C.ReLU N64,K4,S2 | RSB. N512,K3,S1 | C.ReLU N32,K4,S2 | DC.C.ReLU N256,K3,S2 |
| Dense 128 | RSB. N512,K3,S1 | C.ReLU N32,K4,S2 | DC.C.ReLU N128,K3,S2 |
| Dense 2*10 | RSB. N512,K3,S1 | C.ReLU N1,K4,S2 | DC.C.ReLU N512,K3,S1 |

N=Neurons, K=Kernel size, S=Stride size. Convolutional layer is denoted by C. or DC. The residual basic block is denoted as RSB.

### TABLE 3
Classification Accuracy (% , MNIST/CelebA) Under Different Attack and Defense Methods

| Attack | No defense | FM-Defense (Ours) | Defense-GAN | MagNet | FBGAN |
|--------|------------|-------------------|-------------|--------|-------|
| FGSM   | 21.2/7.6   | 99.9/99.9         | 83.21/32.17 | 74.6/59.2 | 80.43/33.71 |
| CW     | 0/0        | 98.78/98.9        | 80.11/28.22 | 19.6/40.5 | 90.8/35.5   |
| DeepFool | 9.1/2.1   | 99.7/99.8         | 81.14/30.17 | 49.4/53.4 | -     |
Fig. 8. The feasibility of data augmentation to address the data limitation. x is different data size settings and y is the average FID of generated samples.

the state-of-the-art attacks that are not strong enough. We hope that our findings would provoke further explorations on designing more powerful attacks even more practical anomaly detection.

In this work, the adversaries are assumed to have no knowledge about the detector and purifier. Even if the attacker knows everything else about the defense, such as network structure, training set, and training procedure, the randomness derived from cryptography could be applied to guarantee the attacker’s computational difficulty. Specifically, we can generate a great number and large diversity of AEs candidates and randomly select one of these AEs for each defensive device for every session, every test set, or even every test example.

We test our approach on various fidelity of face images from 64*64 to 128*128, as well as multi-domain images from Imagenet (224*224, with 1000 classes). The defense performance is also limited to the generative ability of the VAE models, an opening problem for the computer vision community, especially when facing high-fidelity images. The future work also includes incorporating the auto-encoder approaches for high-fidelity images, such as Vector Quantized Variational AutoEncoder.

ACKNOWLEDGMENTS
This work was supported by the Cyber Security Research Centre Limited whose activities are partially funded by the 1348 Australian Government’s Cooperative Research Centres Programme.

REFERENCES

[1] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in Proc. IEEE Eur. Symp. Secur. Privacy, 2016, pp. 372–387.

[2] Y. Liu, X. Chen, C. Liu, and D. Song, “Delving into transferable adversarial examples and black-box attacks,” 2016, arXiv:1611.02770.

[3] K. Grosse, P. Manoharan, N. Papernot, M. Backes, and P. McDaniel, “On the (statistical) detection of adversarial examples,” 2017, arXiv:1702.06280.

[4] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, “On detecting adversarial perturbations,” 2017, arXiv:1702.04267.

[5] S. Tian, G. Yang, and Y. Cai, “Detecting adversarial examples through image transformation,” in Proc. 32nd AAAI Conf. Artif. Intell., 2018, pp. 4139–4146.

[6] D. Meng and H. Chen, “Magnet: A two-pronged defense against adversarial examples,” in Proc. ACM SIGSAC Conf. Comput. Commun. Secur., 2017, pp. 135–147.

[7] N. Carlini and D. Wagner, “Adversarial examples are not easily detected: Bypassing ten detection methods,” in Proc. 10th ACM Workshop Artif. Intell. Secur., 2017, pp. 3–14.

[8] P. Samangouei, M. Kabbab, and R. Chellappa, “Defense-GAN: Protecting classifiers against adversarial attacks using generative models,” 2018, arXiv:1805.07095.

[9] R. Bao, S. Liang, and Q. Wang, “Featurized bidirectional GAN: Adversarial defense via adversarially learned semantic inference,” 2018, arXiv:1805.07862.

[10] Y. Bengio, A. Courville, and P. Vincent, “Representation learning: A review and new perspectives,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 35, no. 8, pp. 1798–1828, Aug. 2013.

[11] M. Carlini, B. Nielsen, L. A. Doctor, and J. Tygar, “The security of machine intelligence,” Mach. Lear., vol. 81, no. 2, pp. 121–148, 2010.

[12] D. Lowd and C. Meek, “Adversarial learning,” in Proc. 11th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining, 2005, pp. 641–647.

[13] B. Biggio et al., “Evasion attacks against machine learning at test time,” in Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discov. Databases, 2013, pp. 387–402.

[14] C. Szegedy et al., “Intriguing properties of neural networks,” 2013, arXiv:1312.6199.

[15] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, “Distillation as a defense to adversarial perturbations against deep neural networks,” in Proc. IEEE Symp. Secur. Privacy, 2016, pp. 582–597.

[16] G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network,” 2015, arXiv:1503.02531.

[17] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in Proc. IEEE Symp. Secur. Privacy, 2017, pp. 39–57.

[18] U. Shaham, Y. Yamada, and S. Negahban, “Understanding adversarial training: Increasing local stability of neural nets through robust optimization,” 2015, arXiv:1511.05432.

[19] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” 2014, arXiv:1412.6572.

[20] A. Mustafa, S. Khan, M. Hayat, R. Goecke, J. Shyam, and L. Shao, “Adversarial defense by restricting the hidden space of deep neural networks,” in Proc. IEEE/CVF Int. Conf. Comput. Vis., 2019, pp. 3385–3394.

[21] G. Cohen, G. Sapiro, and R. Giry, “Detecting adversarial samples using influence functions and nearest neighbors,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2020, pp. 14453–14462.

[22] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, “There is no free lunch in adversarial robustness (but there are unexpected benefits),” 2018, arXiv:1805.12152.
[23] A. Mustafa, S. H. Khan, M. Hayat, J. Shen, and L. Shao, “Image super-resolution as a defense against adversarial attacks,” IEEE Trans. Image Process., vol. 29, pp. 1711–1724, Sep. 19, 2019.

[24] F. Nesti, A. Biondi, and G. Buitazzzo, “Detecting adversarial examples by input transformations, defense perturbations, and voting,” arXiv:2101.11466.

[25] I. Higgins et al., “beta-VAE: Learning basic visual concepts with a constrained variational framework,” in Proc. Int. Conf. Learn. Representations, 2017. [Online]. Available: https://openreview.net/references/pdf?id=rJi_RxvQg

[26] S. Watanabe, “Information theoretical analysis of multivariate correlation,” IBM J. Res. Develop., vol. 4, no. 1, pp. 66–82, 1960.

[27] H. Kim and A. Mnih, “Disentangling by factorising,” 2018, arXiv:1802.05983.

[28] N. Dilokthanakul et al., “Deep unsupervised clustering with Gaussian mixture variational autoencoders,” 2016, arXiv:1611.02648

[29] Z. Zheng and L. Sun, “Disentangling latent space for VAE by label relevant/irrelevant dimensions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2019, pp. 12192–12201.

[30] A. Y. Ng, “Learning deep energy models,” in Int. Conf. Mach. Learn., 2011, pp. 1105–1112.

[31] Y. Gao et al., “RVAE-ABFA: Robust anomaly detection for high dimensional data using variational autoencoder,” in Proc. IEEE 44th Annu. Comput., Softw., Appl. Conf., 2020, pp. 334–339.

[32] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, “Gradient-based learning applied to document recognition,” in Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[33] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms,” 2017, arXiv:1708.07747.

[34] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep learning face attributes in the wild,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 3730–3738.

[35] O. Russakovsky et al., “ImageNet large scale visual recognition challenge,” Int. J. Comput. Vis., vol. 115, no. 3, pp. 211–252, 2015.

[36] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 770–778.

[37] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556.

[38] A. Kurakin, I. Goodfellow, and S. Bengio, “Adversarial examples in the physical world,” 2016, arXiv:1607.02533.

[39] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “Deepfool: a simple and accurate method to fool deep neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2574–2582.

[40] J. Rauber, W. Brendel, and M. Bethge, “Foolbox: A python toolbox to benchmark the robustness of machine learning models,” 2017, arXiv:1707.04131.

[41] S. Zhao, Z. Liu, J. Lin, J.-Y. Zhu, and S. Han, “Differentiable augmentation for data-efficient GAN training,” 2020, arXiv:2006.10738.

[42] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, “GANs trained by a two time-scale update rule converge to a local Nash equilibrium,” 2017, arXiv:1706.08500.

Shuo Wang (Member, IEEE) received the PhD degree from the School of Computing and Information Systems, University of Melbourne. He is currently a research fellow with Data61 & Cybersecurity CRC, CSIRO. His main research interests include adversarial machine learning and computer security and privacy, including security and privacy issues in systems, networking, and databases.

Surya Nepal (Member, IEEE) is currently a principal research scientist with Data61 & Cybersecurity CRC, CSIRO, working on trust and security aspects of web services. His main research interests include development and implementation of technologies in web services and service-oriented architectures.

Carsten Rudolph (Member, IEEE) is currently an associate professor with the Faculty of IT, Monash University, Melbourne, Australia, and the director of the Oceania Cyber Security Centre.

Mathrie Grobler (Member, IEEE) is currently a senior research scientist with Data61 & Cybersecurity CRC, CSIRO, Melbourne, Australia, where she leads the human-centric cybersecurity team, driving the work on cybersecurity governance, policy, and awareness.

Shangyu Chen is currently working toward the postgraduate degree with the Faculty of Computing and Information Systems, University of Melbourne, Melbourne, Australia. His research interests include machine learning and security.

Tianle Chen is currently working toward the postgraduate degree with the Faculty of Information Technology, Monash University, Melbourne, Australia. His research interests include deep learning and privacy.

Zike An received the graduation degree from the University of British Columbia, Vancouver, BC. His research interests include statistics and computer science.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.