Rethinking Self-supervised Learning for Cross-domain Adversarial Sample Recovery

Yi Li, Plamen Angelov, Neeraj Suri
School of Computing and Communications, Lancaster University
Lancaster, UK
{y.li154, p.angelov, neeraj.suri}@lancaster.ac.uk

***Abstract*—Adversarial attacks can cause misclassification in machine learning pipelines, posing a significant safety risk in critical applications such as autonomous systems or medical applications. Supervised learning-based methods for adversarial sample recovery rely heavily on large volumes of labeled data, which often results in substantial performance degradation when applying the trained model to new domains. In this paper, differing from conventional self-supervised learning techniques such as data augmentation, we present a novel two-stage self-supervised representation learning framework for the task of adversarial sample recovery, aimed at overcoming these limitations. In the first stage, we employ a clean image autoencoder (CAE) to learn representations of clean images. Subsequently, the second stage utilizes an adversarial image autoencoder (AAE) to learn a shared latent space that captures the relationships between the representations acquired by CAE and AAE. It is noteworthy that the input clean images in the first stage and adversarial images in the second stage are cross-domain and not paired. To the best of our knowledge, this marks the first instance of self-supervised adversarial sample recovery work that operates without the need for labeled data. Our experimental evaluations, spanning a diverse range of images, consistently demonstrate the superior performance of the proposed method compared to conventional adversarial sample recovery methods.**

***Index Terms*—Self-supervised learning, adversarial sample recovery, autoencoder, representation learning, cross-domain**

I. Introduction

Deep learning techniques are seeing proliferating usage in diverse applications such as image segmentation [1], audio signal processing [2], and natural language processing (NLP) [3] among many others [4]. However, adversarial attacks maliciously attempt to manipulate data in a way that may appear normal to the human eye but can lead to misclassification in a machine learning pipeline. Particularly, adversarial attacks in neural network training encounter several characteristic challenges. Firstly, adversarial attacks expose the vulnerability of neural networks to small, carefully crafted perturbations in input data [5]. This means that models trained conventionally may not be robust and could fail in real-world scenarios where data may be slightly altered or noisy. Secondly, adversarial attacks are often transferable, meaning an attack developed against one model can be successful against another model with similar architecture or even different datasets. This makes it challenging to create robust defenses. Thirdly, when neural networks are trained to defend against specific adversarial attacks, they may become less effective at generalizing to real-world, non-adversarial data. This trade-off between robustness to attacks and generalization to normal data is a significant drawback.

Adversarial sample recovery is a critical subfield within machine learning security, aiming to enhance neural network security by discerning attacks through differences between adversarial and clean image samples. It is applied in various real-world scenarios, such as autonomous driving systems, object detection, medical image processing, landing runway detection, and robotics [4]. Recent advances in deep learning have spurred the development of various approaches [6]–[10] for detecting adversarial attacks. These approaches are predominantly trained using a supervised methodology, involving the provision of a large number of labeled adversarial and normal samples as input to the neural network. The model is then trained to reconstruct the corresponding clean sample and compare it with the input sample to provide a detection or recovery result. However, supervised learning-based adversarial sample recovery approaches exhibit three primary drawbacks.

Firstly, labeling human-imperceptible adversarial attacks on images can be challenging and time-consuming, potentially introducing errors, especially when annotators lack domain familiarity with the task. Secondly, when trained adversarial sample recovery models are deployed in previously unseen conditions, which may include encountering novel attack algorithms and datasets, there is a strong likelihood of a mismatch between the training and test conditions. In such cases, we lack the ability to leverage recorded test adversarial images to improve the model's performance in the unseen test setting. Thirdly, supervised learning in machine learning involves learning a function that maps inputs to outputs based on sample input-output pairs. Therefore, supervised learning techniques are typically task-specific, making it challenging to adapt the trained model to other tasks during the test stage.

Contributions: To address these limitations, we propose an approach based on self-supervised learning (SSL) for adversarial sample recovery. Our method assumes access to a training set comprising clean image examples. Initially, we employ these examples to learn a suitable representation for clean images in an unsupervised manner. Subsequently, we use this learned representation in conjunction with adversarial

images to establish a mapping from the domain of adversarial samples to that of clean samples. These advancements enable the development of adversarial sample recovery systems that can autonomously learn without requiring human intervention, thereby mitigating the various constraints and drawbacks associated with supervised adversarial sample recovery networks.

II. Related Works

In this section, a literature review is provided for basic attack and recovery techniques related to machine learning prior to introducing contemporary supervised and SSL-based adversarial sample recovery methods.

A. Attacks

Recent studies demonstrate that trained neural networks can be compromised by adversarial samples or attacks with human-imperceptible perturbations [5], raising safety concerns about the deployment of these networks in safety-critical applications, including autonomous driving, medical image processing, and clinical settings [11]. According to the threat model, existing adversarial attacks can be categorized into white-box, gray-box, and black-box attacks. The difference between the three models lies in the knowledge of the adversaries. In the frameworks of these threat models, a number of attack algorithms for adversarial sample generation have been proposed, such as Fast Gradient Sign Method (FGSM) [12], Projected Gradient Descent (PGD) [13], Semantic similarity attack on high-frequency components (SSAH) [14], Carlini & Wagner (CW) [15], DeepFool [16], basic iterative method (BIM) [17], and Jacobian-based Saliency Map Attack (JSMA) [18]. For example, as introduced by [14], the SSAH attack concentrates in semantic similarity on feature representations. The high-frequency components of an image contain trivial details and noise, whereas the low-frequency components represent basic information. Therefore, the low-frequency constraint is introduced to limit perturbations within high-frequency components, thus ensuring perceptual similarity between adversarial examples and the original images.

B. Supervised Adversarial Sample Recovery

Developing methods to detect and recover attacks against adversarial examples plays a crucial role in ensuring the robust performance of trained networks. These methods are generally categorized into two main approaches: supervised techniques and SSL-based techniques.

As one of supervised detection and recovery techniques [10], [19]–[23], Qi et al. propose a local neural network as a substitution of the remote target network is trained [20]. The local model estimates the misclassification probability of the perturbed examples in advance and deletes those invalid adversarial examples. Li et al. firstly implement fuzzy logic on the decoder to detect adversarial attacks [10]. The loss between the prediction and label is converted in a fuzzy value to better describe the similarity. Bana et al. introduce a robust recovery (RR) algorithm to recover adversarial examples by generating negative noise and reconstructing the entire image [23]. While these supervised methods achieve high benchmarks, their heavy dependence on labels and datasets constrains their use in real-world applications.

C. Self-supervised Learning

Differing from supervised learning, self-supervised adversarial sample recovery algorithms do not require the access to any paired training data, i.e., adversarial and normal images. To achieve this, researchers learn features from a large number of clean images and fine-tune with labels for the downstream task, i.e., adversarial attack detection or recovery. Zhang et al. train an autoencoder with Disentangled Representation-based Reconstruction (DRR) over both correctly paired labels and incorrectly paired labels to reconstruct benign and counterexamples [24]. This mimics the behavior of adversarial examples and can reduce the unnecessary generalization ability of autoencoder. Moayeri et al. classify adversarial attacks to their respective threat models, based on a linear model operating on the embeddings from a pre-trained self-supervised encoder, SimCat [25]. These SSL-based adversarial attack detection methods, as mentioned earlier, achieve promising accuracy across commonly used datasets. However, these methods have two limitations. Firstly, these models rely on labels in the training stage. For example, DRR extracts labels and features from both clean and adversarial images, calculating the loss between the predicted label and the true label to train the autoencoder. Secondly, these methods lack consideration of the possibility that the trained model may be employed across different domains, while it is a common scenario in the real world. These limitations are addressed by the proposed SSL pipeline in this work.

III. Proposed Method

A. Preliminaries

In this section, we discuss the training of each network component of our scheme. In the proposed method, two variational autoencoders (VAEs) [26], [27], namely, the CAE and the AAE, are progressively utilized to reconstruct clean images and adversarial images, respectively. In CAE, the encoder and decoder are denoted as E_{CAE} and D_{CAE}, while in AAE, they are denoted as E_{AAE} and D_{AAE}, respectively. In the test stage, trained E_{AAE} and D_{CAE} are combined to form the final model, which recover the image with potential attacks. It is worth noting that the clean images used in the CAE are not used for generating the adversarial images in the AAE to learn the shared feature space. Besides, the adversarial images used in the AAE are unseen in the test stage for the fair evaluation.

B. Stage 1: Clean Image Autoencoder

The training pipeline of the CAE is presented in Fig. 1.

Initially, as the input of the CAE, the clean images are fed into the E_{CAE}. Then, the features are extracted from the clean images. The decoder D_{CAE} obtains the latent representation of clean images to reconstruct the images. To achieve that, the CAE is trained by minimizing an appropriate measure

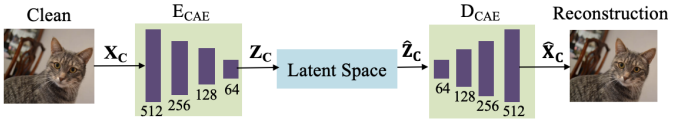

Fig. 1. Proposed pipeline of the clean image autoencoder (CAE). The clean images are fed into the CAE to produce the reconstruction.

of discrepancy between the input clean image $\mathbf{X_S}$ and its reconstruction $\hat{\mathbf{X}}_\mathbf{S}$ from $\mathrm{D_{CAE}}$ with a hyper-parameter λ_1:

$$\mathcal{L}_{\mathrm{CAE}} = \|\mathbf{X_S} - \hat{\mathbf{X}}_\mathbf{S}\|_2^2 + \lambda_1 \cdot \mathcal{L}_{\mathrm{KL-CAE}} \tag{1}$$

where $\mathcal{L}_{\mathrm{KL-CAE}}$ is Kullback–Leibler (KL) loss [28] to learn a latent representation that is close to a zero-mean normal distribution. The L2 norm of the loss is presented as $\|\cdot\|_2^2$. The hyper-parameter λ_1 is empirically set to 0.001, the supportive diagnostic experiment will be presented later. By reconstructing the images, the CAE is trained to learn a latent representation of clean images.

As presented in Fig.1, in CAE, both $\mathrm{E_{CAE}}$ and $\mathrm{D_{CAE}}$ consist of four 1-D convolutional layers. In $\mathrm{E_{CAE}}$, the size of the hidden dimension decreases sequentially from 512 $\rightarrow$ 256 $\rightarrow$ 128 $\rightarrow$ 64. Accordingly, the dimension of the latent space is set to 64, with the stride of 1 and the kernel size of 7 used for the convolutions. Different from $\mathrm{E_{CAE}}$, the decoder $\mathrm{D_{CAE}}$ scale up the latent dimensions sequentially. We will show the experimental results to confirm the configuration in Section IV-D.

C. Stage 2: Adversarial Image Autoencoder

The training process of the AAE is marked from ① to ⑤ in Fig. 2.

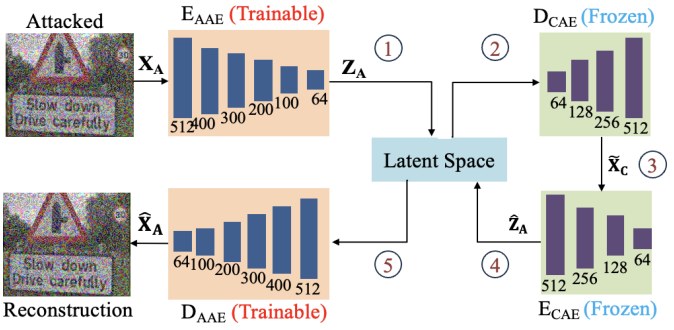

Fig. 2. Proposed pipeline of the adversarial image autoencoder (AAE). In this period, the AAE is trained to produce the estimated adversarial images, while the weights of the CAE are *frozen*.

Once the CAE is trained with the clean images, we now fix the weights of the trained CAE and use it in the AAE training with the adversarial images. Different from the CAE, the AAE only needs access to unseen adversarial images. Note that the adversarial images in the AAE are not generated from the clean images in the CAE. The feature is extracted from the adversarial images and fed to $\mathrm{E_{AAE}}$. Consequently, the latent representation of the adversarial images is obtained as the output of $\mathrm{E_{AAE}}$ and exploited to modify the loss functions and learn a shared latent space between the clean image feature representation $\mathbf{Z_S}$ and adversarial images representation $\mathbf{Z_A}$ as the core idea of this work.

To achieve this, the adversarial image representation is passed through the decoder of the CAE to get the enhanced version of the adversarial image representation as the process ③ in Fig. 2:

$$\tilde{\mathbf{X}}_\mathbf{C} = \mathrm{D_{CAE}}(\mathbf{Z_A}) \tag{2}$$

Benefiting from the learned clean image representation, a mapping from the adversarial image to the corresponding clean version is learned with the latent representation of the clean image feature. Then, the estimated adversarial image representation is obtained from the trained $\mathrm{E_{CAE}}$ as the process ④ in Fig. 2:

$$\hat{\mathbf{Z}}_\mathbf{A} = \mathrm{E_{CAE}}(\tilde{\mathbf{X}}_\mathbf{C}) \tag{3}$$

Furthermore, $\mathrm{D_{AAE}}$ is trained to produce the estimated the adversarial image as:

$$\hat{\mathbf{X}}_\mathbf{A} = \mathrm{D_{AAE}}(\hat{\mathbf{Z}}_\mathbf{A}) \tag{4}$$

With these relationships, we now enforce that the cycle reconstruction of the adversarial image $\hat{\mathbf{X}}_\mathbf{A}$ resembles the input adversarial image $\mathbf{X_A}$. Likewise, we also enforce that the two latent representations before and after the cycle loop through the CAE are close.

The overall loss to train the AAE is a combination of three loss terms with hyper-parameter λ_2:

$$\mathcal{L}_{\mathrm{AAE}} = \lambda_2 \cdot \mathcal{L}_{\text{KL-AAE}} + \mathcal{L}_{\mathrm{cycle}} \tag{5}$$

Similarly, the hyper-parameter λ_2 is empirically set to 0.001. Moreover, $\mathcal{L}_{\text{KL-AAE}}$ denotes the KL loss and is applied to train the latent representation closed to a normal distribution. Besides, the cycle loss $\mathcal{L}_{\mathrm{cyc}}$ consists of two loss terms, $\mathcal{L}_{\mathrm{X_A}}$ and $\mathcal{L}_{\mathrm{Z_A}}$:

$$\mathcal{L}_{\mathrm{cycle}} = \|\mathbf{X_A} - \hat{\mathbf{X}}_\mathbf{A}\|_2^2 + \left\|\mathbf{Z}_A - \hat{\mathbf{Z}}_\mathbf{A}\right\|_2^2 \tag{6}$$

where $\mathcal{L}_{\mathrm{A}}$ refers to the loss between the input adversarial image and the corresponding reconstruction from $\mathrm{D_{AAE}}$. Moreover, the loss between the latent representation and the corresponding reconstruction is presented as $\mathcal{L}_{\mathrm{Z_A}}$. Finally, the overall loss $\mathcal{L}_{\mathrm{AAE}}$ is utilized in AAE to improve the adversarial attack revocery performance.

The AAE network follows an architecture similar to CAE. E_M consists of six 1-D convolutional layers where the hidden layer sizes are decreased from 512 $\rightarrow$ 400 $\rightarrow$ 300 $\rightarrow$ 200 $\rightarrow$ 100 $\rightarrow$ 64, while those of the decoders increase inversely. Again, the experimental results will be provided in Section IV-D to confirm the configuration.

D. Test Stage

In the test stage, the feature of the input images is extracted and fed to the trained $\mathrm{E_{AAE}}$. Then, the latent representation of the input images is obtained. Because the decoder $\mathrm{D_{CAE}}$ are trained to produce the clean image, the input image representation is fed to the decoder to recover the original image. Finally,

the recovery ratio is calculated between the input image and reconstructed image to evaluate the adversarial image recovery performance of the proposed model. The pipeline of the test stage is presented in Fig. 3.

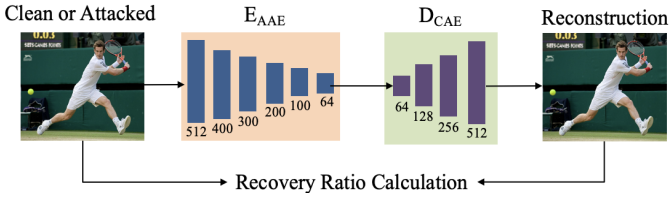

Fig. 3. Test stage pipeline. The trained E_{AAE} and D_{CAE} are combined as the final model.

IV. Experimental Results

A. Datasets and Attacks

We extensively perform experiments on several public datasets, including ImageNet-R [29], Canadian Institute For Advanced Research-10 (CIFAR-10) [30], and Microsoft Common Objects in Context (COCO) [31]. We randomly select 10,000 images from COCO to train the CAE. Then, 40,000 images from CIFAR-10 are randomly selected to train the AAE. Furthermore, we randomly select 10,000 images from ImageNet-R to evaluate the adversarial sample recovery accuracy of competitors and proposed model.

We select aforementioned attacks in Section II because they are robust to novel adversarial attack detection and recovery techniques [14], [19]. We randomly select two attack algorithms to generate the adversarial images for the AAE training and final model evaluation, respectively. Parameters of all the seven attacks exploited in the experiments are shown in Table I.

TABLE I
PARAMETERS OF SEVEN ADVERSARIAL ATTACKS

Attack	Parameters
FGSM	ϵ=0.008
PGD	ϵ=0.01, α=0.02, Steps=40
SSAH	α=0.01
DeepFool	Steps=20
BIM	ϵ=0.03, α=0.01, Steps=10
CW	C=2, Kappa=2, Steps=500, learning rate=0.01
JSMA	γ=0.02

B. Competitors and Performance Measure

In addition, the proposed method is evaluated and compared to state-of-the-art competitor models. Firstly, we select four supervised adversarial attack detection and recovery techniques [19]–[23] as the original implementations in the literature but with same data as the proposed method. Secondly, we use five pre-trained self-supervised models [32]–[36] which are state-of-the-art in image processing tasks. Thirdly, two self-supervised adversarial attack detection and recovery techniques [24], [25] are adopted. For a fair comparison, we reproduce these models with same data as the proposed method.

In the experiment, the recovery rate (RR) is used as the performance measure, calculated as the ratio of correctly recovered pixels to all pixels.

C. Model Configuration

We propose a self-supervised training pipeline to address the limitations of supervised pipelines, therefore, the backbone of autoencoders is out of scope of this paper. We simply conduct experiments to demonstrate the trade-off between performance improvement and network depth, specifically, varying the number of convolutional layers.

The proposed model is trained by using the SGD optimizer with a weight decay of 0.0001, a momentum of 0.9, and a batch size of 256. We train the CAE for 700 epochs. Then, we train the AAE for 1500 epochs, where we warm-up the network in the first 300 epochs by only using the adversarial image reconstruction loss $\mathcal{L}_{\mathrm{A}}$. The initial learning rate is 0.03, and is multiplied by 0.1 at 500 and 1000 epochs. All experiments are run on the High End Computing (HEC) Cluster with Tesla V100 GPUs.

D. Diagnostic Experiment

An ablation study of hyper-parameters λ_1 and λ_2 is conducted on the ImageNet-R dataset. Fig. 4 shows downstream attack recovery accuracy when they varies from 0.0001 to 0.1. Each data point being an average of 70,000 experiments (10,000 images $\times$ 7 attacks).

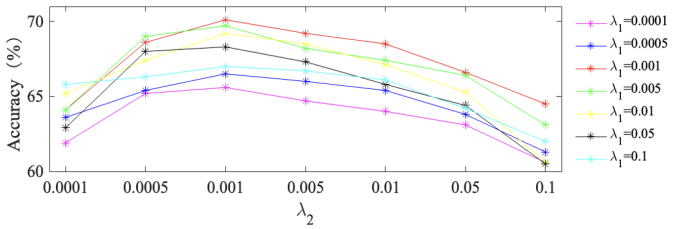

Fig. 4. Ablation study for hyper-parameters λ_1 and λ_2.

According to Fig. 4, the value of 0.001 is optimal for both λ_1 and λ_2. The recovery accuracy achieves 70.1% at the peak. We conduct experiments to demonstrate the trade-off between performance improvement and network depth, specifically, varying the number of convolutional layers in two autoencoders, i.e., CAE and AAE.

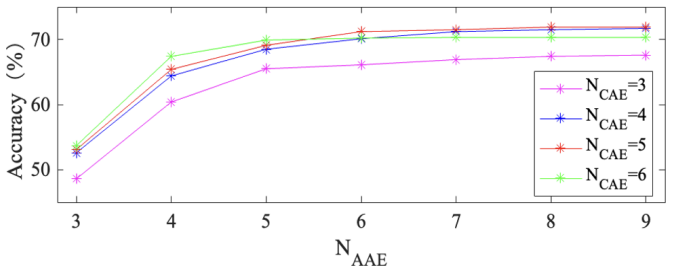

Fig. 5. Ablation study for number of convolutional layers in the CAE and AAE.

TABLE II
COMPARISON WITH **SEEN** ATTACK ALGORITHMS ON THE IMAGENET-R DATASET. AVR DENOTES THE AVERAGE RECOVERY ACCURACY ((%)).

	Algorithm & Network			DR (%)								
Method	Supervised	Pre-training	Backbone	Clean	FGSM	PGD	SSAH	DeepFool	BIM	CW	JSMA	Avr
ESMAF [22]	✓	✗	ResNet-101V2	70.8	52.7	67.5	62.9	39.7	35.9	37.2	41.0	51.0
SSAE [21]	✓	✗	ResNet-34	74.0	58.5	67.0	69.2	41.5	39.4	41.6	41.3	54.1
sim-DNN [19]	✓	✓	VGG16+DNN	76.2	60.7	72.3	71.0	44.8	46.7	49.9	50.2	59.0
DTBA [20]	✓	✓	VTCNN-2	79.2	59.2	75.5	74.9	51.4	53.8	56.0	59.9	63.8
RR [23]	✓	✓	VGG-13	86.5	62.7	79.0	76.2	67.1	58.7	60.9	71.3	70.3
TiCo [32]	✗	✓	Mask R-CNN+FPN	74.5	53.6	68.6	65.2	45.1	44.5	43.1	57.9	56.6
MAE [33]	✗	✓	ViT-L/16	82.2	59.6	75.5	74.4	54.2	50.3	51.4	62.8	63.8
Mugs [34]	✗	✓	ViT L/14	83.4	57.2	75.9	76.7	56.0	51.1	50.8	64.3	64.4
Unicom [35]	✗	✓	ViT L/14	86.4	59.8	76.2	79.3	61.0	55.5	58.4	68.2	68.1
DINOV2 [36]	✗	✗	ViT g/14	87.5	61.6	79.4	78.3	64.5	57.1	57.9	71.6	69.7
SimCat [25]	✗	✓	ResNet-50	85.1	58.0	75.2	77.0	56.4	56.5	55.3	69.6	66.6
DRR [24]	✗	✓	VGG-16	87.2	64.8	79.6	78.2	66.9	60.7	60.1	70.3	71.0
Ours	✗	✗	CNN	**87.9**	**65.9**	**80.0**	**79.7**	**69.1**	**61.5**	**61.8**	**72.4**	**72.3**

TABLE III
COMPARISON WITH **UNSEEN** ATTACK ALGORITHMS ON THE IMAGENET-R DATASET.

	DR (%)								
Method	Clean	FGSM	PGD	SSAH	DeepFool	BIM	CW	JSMA	Average
ESMAF [22]	61.4	40.6	58.0	48.2	22.1	20.6	25.6	35.8	39.0
SSAE [21]	63.0	42.9	59.1	54.7	29.2	26.5	33.0	36.0	43.1
sim-DNN [19]	64.5	42.5	60.4	57.2	34.9	30.5	34.8	38.1	45.4
DTBA [20]	68.9	42.0	65.7	65.8	35.3	31.0	37.2	40.1	48.3
RR [23]	74.6	50.8	70.2	72.7	43.8	34.3	49.6	58.9	56.9
TiCo [32]	74.4	53.9	68.2	64.8	45.8	45.1	44.0	57.2	56.7
MAE [33]	82.6	59.4	75.5	73.9	54.6	50.5	51.2	62.9	63.8
Mugs [34]	83.3	57.0	75.2	76.9	55.7	50.1	51.4	63.9	64.2
Unicom [35]	86.0	59.6	76.2	78.8	61.2	55.4	58.5	68.0	68.0
DINOV2 [36]	87.6	62.0	79.2	78.4	64.8	57.2	57.5	**71.5**	70.0
SimCat [25]	83.0	56.7	73.1	73.8	55.6	55.2	53.9	69.0	65.0
DRR [24]	87.0	64.0	**80.1**	78.1	65.7	**60.3**	59.4	70.9	70.7
Ours	**87.9**	**64.1**	79.0	**78.6**	**67.5**	59.8	**59.9**	69.8	**70.8**

Fig. 5 compares the number of convolutional layers in two autoencoders against recovery accuracy on ImageNet-R. As Fig. 5 shows, recovery accuracy starts to increase with N_{CAE}&N_{AAE} = 3 and reaches the peak around $N_{CAE} = 4$ and $N_{AAE} = 6$, but performance is fairly stable for $4 \leq N_{CAE} \leq 6$ and $6 \leq N_{AAE} \leq 9$. Therefore, the results indicate that $N_{CAE} = 4$ and $N_{AAE} = 6$ offer the best trade-off, validating the chosen implementation setting.

E. Comparison to State-of-the-Arts

We conduct two experiments in this section. In the first experiment, we use the same attack but different datasets between the training and test stages to evaluate and compare the recovery ratio of competitors and proposed methods. Table II shows the results, each of them is the average of 10,000 images.

Table II shows the averaged adversarial sample recovery performance of the proposed method as compared to [19]–[25], [32]–[36] on the ImageNet-R dataset. From Table II, it can be observed that in all the evaluated models, the proposed model achieves 87.9% and 70.1% for clean and adversarial images recovery, respectively, which offers the best effectiveness.

In the second experiment, we evaluate both different attack algorithms and datasets between the training and test stages as a more challenging scenario. In the data pre-processing, we select one attack algorithm to generate the test data and randomly select the other attack algorithm from the rest of Table I to generate the training data. It is highlighted that we use the same generated data for all competitors and the proposed model for a fair comparison. Table III shows the results, each of them is the average of 10,000 images.

Table III shows that our pre-training significantly improves results over self-supervised vision learning models and supervised pre-training. Our model is 0.8 points higher than DINOV2 (70.8% vs. 70.0%). Our self-supervised learning models also outperforms the supervised models. These observations are consistent with those in ImageNet-R. Moreover, comparing the results to Table II, supervised models suffer a significant accuracy degradation with unseen attacks, while self-supervised models perform more consistent. We conduct more experiments to confirm this point in the next section.

F. Robustness Evaluation

We perform experiments on ImageNet-R to evaluate the robustness of the supervised and self-supervised models on adversarial attack detection. Fig. 6 shows the detection accuracy results (in %), each of them is the average of 80,000 experiments (10,000 images $\times$ 8 (1 clean + 7 attack algorithms)).

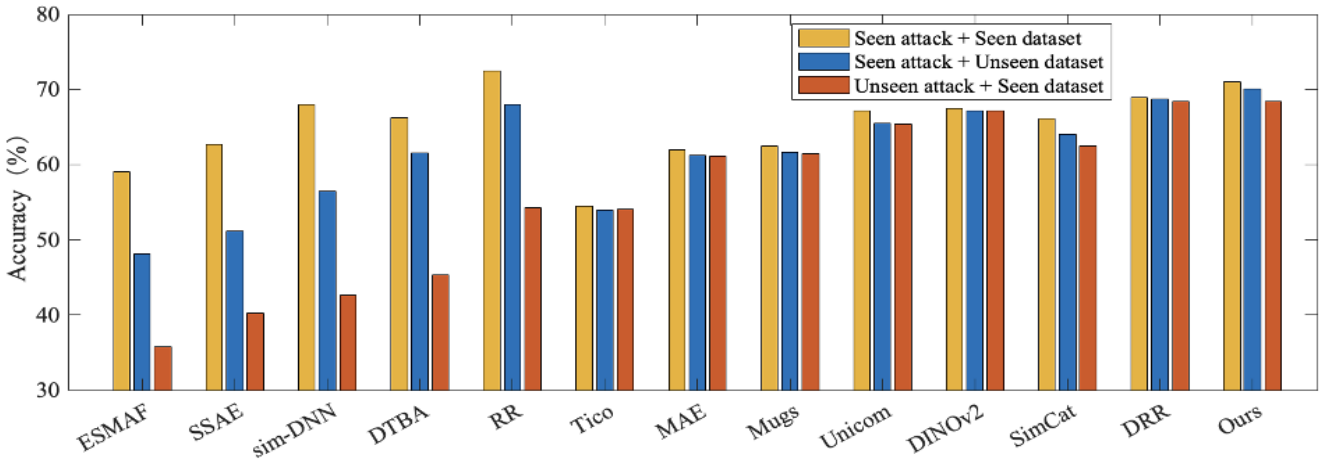

Fig. 6. Robust study of supervised and self-supervised adversrial image recovery.

It can be observed that: (1) The adversarial sample recovery accuracy on unseen attack algorithms and datasets is lower compared to the seen attack utilized in both the training and test stages, primarily due to differences in distributions of datasets. (2) When compared to supervised learning-based methods [19]–[23], the proposed SSL representation learning method experiences relatively less performance degradation. The reason is that unlike supervised learning models, SSL models extract the underlying sub-class structure from the augmented data distribution and encodes it into the embeddings. This guarantees the robustness of the downstream task. (3) The proposed model shows the best performance in challenging scenarios because it learns mapping cross-domain adversarial samples to the domain of clean image representation via the shared latent space between two autoencoders.

G. Visualizations

In this section, we present qualitative results demonstrating the recovery of attacked image samples on ImageNet-R in Fig. 7. Particularly, we compare the proposed model to a baseline autoencoder with the same configuration as the AAE.

After comparing the recovery results of the proposed model and the baseline, the reconstructions obtained via the proposed method, are closer to original images, which confirms the efficacy of the proposed method.

H. Discussion

The above detailed experimental results confirm that the proposed self-supervised method with a shared latent space can further improve adversarial sample recovery performance in different scenarios, i.e., seen or unseen datasets and attack algorithms, compared to the state-of-the-art methods.

Moreover, a significant advantage of the proposed pipeline is its independence from labels. The two autoencoders are dedicated to different tasks; the CAE learns the clean image representation, and the AAE learns the adversarial image representation. We ensure that the encoder learns the cross-domain representation even though the input to the two autoencoders consists of unpaired images. This ensures the robustness of the downstream task.

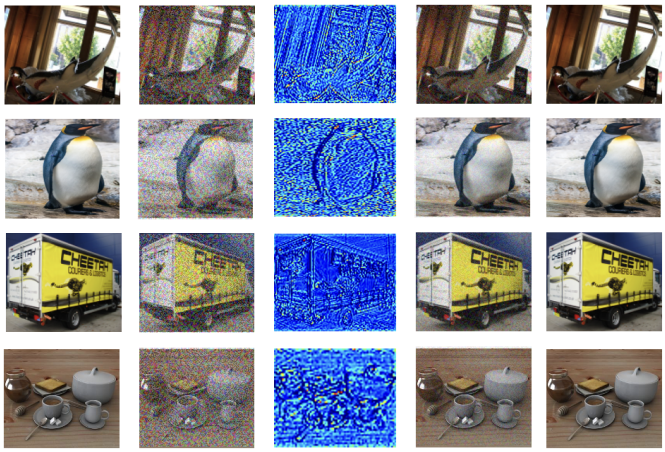

Fig. 7. Qualitative adversarial sample recovery results on ImageNet-R. From left to right: input images, images attacked by FGSM, feature maps of the proposed model, recovery results of the baseline, recovery results of the proposed model.

Furthermore, an additional advantage is the reusability of the pre-trained CAE. It can be employed to recover adversarial samples with different distributions, irrespective of the nature of the adversarial attack algorithms.

V. Conclusion

In this paper, we have proposed a self-supervised adversarial sample recovery method, a simple yet effective replacement to the conventional supervised pipelines. We firstly trained the CAE using clean images, learning an appropriate latent representation. This latent representation was then used in a downstream adversarial sample recovery task to train the AAE for adversarial images so that the two autoencoders shared their latent spaces. This allowed us to map the domain of adversarial images to the domain of clean images. Differing from conventional SSL-based adversarial attack recovery

works, the proposed pipeline does not require labels for the mode training. Our evaluation on cross-domain experiments over different datasets and attacks has demonstrated the high effectiveness of the proposed method.

ACKNOWLEDGMENT

Research supported by the UKRI Trustworthy Autonomous Systems Node in Security/EPSRC Grant EP/V026763/1.

REFERENCES

[1] C. Li, L. Wang, and Y. Li, "Transformer and group parallel axial attention co-encoder for medical image segmentation," *Scientific Reports*, vol. 12, p. 16117, 2022.

[2] Y. Li, Y. Sun, W. Wang, and S. M. Naqvi, "U-shaped Transformer with frequency-band aware attention for speech enhancement," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, p. 1511–1521, 2023.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *International Conference on Neural Information Processing Systems (NeurIPS)*, 2017.

[4] K. Ren, T. Zheng, Z. Qin, and X. Liu, "Adversarial attacks and defenses in deep learning," *Engineering*, vol. 6, no. 3, pp. 346–360, 2020.

[5] X. Ma, Y. Niu, L. Gu, Y. Wang, Y. Zhao, J. Bailey, and F. Lu, "Understanding adversarial attacks on deep learning based medical image analysis systems," *Pattern Recognition*, vol. 110, p. 107332, 2021.

[6] Y. Li, P. Angelov, and N. Suri, "Domain generalization and feature fusion for cross-domain imperceptible adversarial attack detection," *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2023.

[7] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. Mc-Daniel, "Ensemble adversarial training: attacks and defenses," *International Conference on Learning Representations (ICLR)*, 2018.

[8] A. L. Pellcier, K. Giatgong, Y. Li, N. Suri, and P. Angelov, "UNICAD: A unified approach for attack detection, noise reduction and novel class identification," *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2024.

[9] Z. Ji, B. Yang, P. L. Yeoh, Y. Zhang, Z. He, and Y. Li, "Active attack detection based on interpretable channel fingerprint and adversarial autoencoder," *IEEE International Conference on Communications (ICC)*, 2022.

[10] Y. Li, P. Angelov, and N. Suri, "Adversarial attack detection via fuzzy predictions," *Proceedings of IEEE Symposium Series on Computational Intelligence (SSCI)*, 2023.

[11] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: attacks and defenses for deep learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 9, pp. 2805 – 2824, 2019.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *Proceedings of International Conference on Learning Representations (ICLR)*, 2015.

[13] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *Proceedings of International Conference on Machine Learning (ICML)*, 2017.

[14] C. Luo, Q. Lin, W. Xie, B. Wu, J. Xie, and L. Shen, "Frequency-driven imperceptible adversarial attack on semantic similarity," *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[15] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," *IEEE Symposium on Security and Privacy*, 2017.

[16] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: a simple and accurate method to fool deep neural networks," *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[17] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[18] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The Limitations of Deep Learning in Adversarial Settings," *IEEE Symposium on Security and Privacy*, 2016.

[19] E. Soares, P. Angelov, and N. Suri, "Similarity-based deep neural network to detect imperceptible adversarial attacks," *Proceedings of IEEE Symposium Series on Computational Intelligence (SSCI)*, 2022.

[20] P. Qi, T. Jiang, L. Wang, X. Yuan, and Z. Li, "Detection tolerant black-box adversarial attack against automatic modulation classification with deep learning," *IEEE Transactions on Reliability*, vol. 71, no. 2, pp. 674–686, 2022.

[21] C. Cintas, S. Speakman, V. Akinwande, W. Ogallo, K. Weldemariam, S. Sridharan1, and E. McFowland, "Detecting adversarial attacks via subset scanning of autoencoder activations and reconstruction error," *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2020.

[22] J. Chen, T. Yu, C. Wu, H. Zheng, W. Zhao, L. Pang, and H. Li, "Adversarial attack detection based on example semantics and model activation features," *Proceedings of International Conference on Data Science and Information Technology (DSIT)*, 2022.

[23] T. Bana, J. Loya, and S. R. Kulkarni, "Robust recovery of adversarial examples," *Proceedings of International Conference on Machine Learning (ICML)*, 2021.

[24] Z. Zhang, L. Y. Zhang, X. Zheng, J. Tian, and J. Zhou, "Self-supervised adversarial example detection by disentangled representation," *Proceedings of TrustCom*, 2022.

[25] M. Moayeri and S. Feizi, "Sample efficient detection and classification of adversarial attacks via self-supervised embeddings," *Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[26] Z. Li, Y. Zhao, H. Xu, W. Chen, S. Xu, Y. Li, and D. Pei, "Unsupervised clustering through gaussian Mmxture variational autoencoder with non-reparameterized variational inference and std annealing," *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2020.

[27] Y. Li, Y. Sun, K. Horoshenkov, and S. M. Naqvi, "Domain adaptation and autoencoder based unsupervised speech snhancement," *IEEE Transactions on Artificial Intelligence*, vol. 3, no. 1, pp. 43 – 52, 2021.

[28] I. Csiszár, "i-divergence geometry of probability distributions and minimization problems," *Annals of Probability*, vol. 3, pp. 146–158, 1975.

[29] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, D. Song, J. Steinhardt, and J. Gilmer, "The many faces of robustness: a critical analysis of out-of-distribution generalization," *Proceedings of IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[30] A. Krizhevsky, "Learning multiple layers of features from tiny images," *Master's thesis*, 2009.

[31] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, "Microsoft COCO: common objects in context," *Proceedings of European Conference on Computer Vision (ECCV)*, 2014.

[32] J. Zhu, R. Moraes, S. Karakulak, V. Sobol, A. Canziani, and Y. LeCun, "Tico: transformation invariance and covariance contrast for self-supervised visual representation learning," *arXiv preprint arXiv: 2203.14415*, 2022.

[33] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[34] P. Zhou, Y. Zhou, C. Si, W. Yu, T. K. Ng, and S. Yan, "Mugs: a multi-granular self-supervised learning framework," *arXiv preprint arXiv: 2203.14415*, 2022.

[35] X. An, J. Deng, K. Yang, J. Li, Z. Feng, J. Guo, J. Yang, and T. Liu, "Unicom: universal and compact representation learning for image retrieval," *Proceedings of International Conference on Learning Representations (ICLR)*, 2023.

[36] M. Oquab, T. Darcet, T. Moutakanni, H. Vo, M. Szafraniec, V. Khalidov, P. Fernandez, D. Haziza, F. Massa, A. El-Nouby, M. Assran, N. Ballas, W. Galuba, R. Howes, P.-Y. Huang, S.-W. Li, I. Misra, M. Rabbat, V. Sharma, G. Synnaeve, H. Xu, H. Jegou, J. Mairal, P. Labatut, A. Joulin, and P. Bojanowski, "Dinov2: learning robust visual features without supervision," *arXiv preprint arXiv: 2304.07193*, 2023.