Domain Generalization and Feature Fusion for Cross-domain Imperceptible Adversarial Attack Detection

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*—Deep learning-based imperceptible adversarial attack detection methods have recently seen significant progress. However, the accuracy, latency, and computational cost of previous methods remain insufficient. Particularly, trained attack detection models can potentially be applied in previously unseen conditions, such as new datasets or attacks for real-world applications. Therefore, to improve domain generalization performance, we propose a new method for cross-domain imperceptible adversarial attack detection by leveraging domain generalization, where we train the model's feature extractor or detector with a partner well-tuned for different domains. Different from conventional domain generalization methods, we use the global loss and local loss to train each feature extractor or detector. Moreover, to efficiently re-use high-resolution feature maps from the feature extractor, we propose a feature fusion network, which exploits feature maps from images that are attacked with different error rates and helps extract rich features to further improve the attack detection accuracy. Extensive experiments on four public datasets are used to demonstrate the efficacy of the proposed method. The source code of the proposed method is available at https://github.com/Yukino-3/DTAD.**

***Index Terms*—imperceptible adversarial attack, domain generalization, cross-domain, local loss, feature fusion**

I. Introduction

Machine learning techniques play an important part in real-world applications. However, the vulnerability to adversarial training samples becomes one of the major risks for applying neural networks in safety-critical environments because the performance extremely relies on the accuracy of training samples or labels [1]. Due to the importance of the training samples for machine learning technologies, numerous attack detection methods have been proposed to solve this problem. However, there are two key challenges.

Well-trained networks often suffer a performance degradation if deployed to unseen domains with very different statistics to the training data [2]. This so-called domain shift problem is a common challenge in real-world scenarios [3]. The conventional methods to solve this problem can be categorized into domain adaptation and domain generalization methods. Domain adaptation is exploited to solve in the case where some labelled or unlabelled data from the target domain is available for adaptation [4]–[6], while domain generalization aims to solve in the case where no adaptation to the target problem is available due to lack of data or computation [7]. Certainly, domain generalization is a more challenging and practical problem setting than domain adaptation because explicit training on the target is disallowed; yet it is particularly valuable due to its lack of assumptions [8]. However, the domain generalization study in imperceptible adversarial attack detection is limited.

The second challenge is model scaling. In multi-resolution based encoder and decoder architectures, previous approaches mainly rely on larger backbone networks to achieve higher performance [9], [10]. However, in recent studies [11], scaling up feature extractor and classifier is proved to play a more important role in the trade-off between accuracy and efficiency. Tan et al. propose a compound scaling method to jointly scale up the depth, width, and resolution for all backbones, feature networks, and prediction networks [11]. The EfficientDet method consistently achieves high accuracy and reduces the model size. However, the EfficientDet method is applied for the object detection task in the original implementation [11]. The performance of the EfficientDet in attack detection is limited, as shown in Section IV. C.

In this work, we propose a new method to mitigate these two challenges. Firstly, we propose a global loss to train the feature extractor or detector of the neural network with a partner who is well tuned for different domains. The partner is a detector when training the feature extractor, and vice versa. This allows each component, i.e., the feature extractor or detector performs robust in unseen domains. Then, to guarantee the attack detection performance, we propose a local loss for each pair of feature extractor and detector. Furthermore, after extracting feature maps from images that are attacked with different error rates, a feature fusion network (FFN) is proposed to exploit these feature maps and efficiently recover lost information from the high-level representation to further improve the attack detection accuracy. Finally, the well-trained network components, i.e., the feature extractor, feature fusion network, and detector, are combined and evaluated with unseen domains.

E-mail for correspondence: y.li154@lancaster.ac.uk.

The rest of the paper is organized as follows. The related work to attack detection and domain generalization is introduced in Section II. Then, we describe the proposed method in Section III. The experimental settings and results are presented in Section IV. Finally, Section V concludes the work.

II. Related Works

In this section, a literature review is provided for basic attack and detection techniques related to machine learning. Then, we introduce conventional domain generalization methods.

A. Attacks

Recent studies demonstrate that trained neural networks can be compromised by adversarial samples or attacks with small imperceptible perturbations [12]. This raises safety concerns about the deployment of these networks in real-world applications, including autonomous driving, and clinical settings [1].

1) Fast Gradient Sign Method (FGSM): : As one of single-step adversarial attacks, FGSM exploits backpropagation to efficiently calculate the required gradient [13]. The adversarial image is calculated as $y + \epsilon * \text{sign}(\Delta y J(\lambda, x, y))$ where x and y are the attacked image and the original image, respectively. The scale of the distortion is presented as ϵ. Moreover, $J(\lambda, x, y)$ is the cost function with the network parameter λ.

2) Projected Gradient Descent (PGD): : This attack introduces an iterative version of the FGSM attack. The authors add a perturbation step in each step of the training stage to make the algorithm more robust [14]. The experiments in [14] prove that the PGD attack is one of the strongest attacks to different detection and defense techniques.

3) Semantic similarity attack on high-frequency components (SSAH): : As introduced by [15], 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. By using the constraint, the perceptual similarity between adversarial examples and originals is ensured.

B. Detection

Developing attack detection against adversarial examples plays an important role in guaranteeing the performance robustness of trained networks. Several detection techniques are introduced in the literature.

1) Adversarial Training: The adversarial training aims to augment the training with FGSM samples [13]. For example, in [16], the reverse cross-entropy (RCE) is minimized to encourage the neural network to learn the latent representation that better detects attacked samples, e.g., FGSM samples or labels from normal ones. However, the recent study [17] shows that adversarial training is not always robust, particularly against iterative white-box attacks and black-box single-step attacks.

2) L-RED: Backdoor adversarial attacks aim to force the neural network to classify to a target class when test samples from multiple source classes contain a backdoor pattern while maintaining high accuracy on all clean test samples [18]. Different from conventional defending methods, access to the training dataset is not required in Reverse-Engineering-based Defenses (REDs) against backdoor adversarial attacks [19]. Moreover, the number of source classes can be unavailable but only requires very few clean images to detect the backdoor attacks and reduce the computational complexity [20].

3) Sim-DNN: As one of the deep learning-based detection techniques, Soares et al. propose a similarity-based deep neural network (sim-DNN) to detect adversarial attacks [21]. The degree of similarity between training samples and their prototypes is considered. By minimizing the similarity score, the concept changes are detected from the attacked data when comparing their similarities against the set of prototypes. Compared to the above-mentioned two detection methods, Sim-DNN efficiently improves the detection performance against various attacks, i.e., FGSM, PGD, and Decoupling Direction and Norm (DDN) [22].

C. Domain Generalization

Recently, domain generalization methods can be divided into several categories by motivating intuition [3].

1) Domain Invariant Features: Assuming that the invariant features from the source domains works well in the target domains, these features minimize the discrepancy between source domains to learn the domain-invariant representation [23]. In generative adversarial networks (GAN) based adversarial training, explicit features are generated to fool the decoder or domain discriminator by detecting the source features from target features [24].

2) Data Augmentation: In recent studies [25], [26], data augmentation-based techniques are exploited to synthesise additional training data to further boost the robustness of neural networks to unseen target domains. For example, data augmentation and domain distance minimisation are combined to provide a guarantee on the domain generalization performance [25]. To achieve that, the authors hypothesise a high dimensional space in which each axis corresponds to an independent augmentation function.

3) Optimisation Algorithms: These develop optimisation algorithms such as episodic training and meta-learning to solve domain shift problems [2], [27]. Li et al. propose an episodic training to solve domain generalization problems [2]. A classifier with random weights is exploited to train the target feature extractor. Then, the target classifier is trained with a domain-specific feature extractor.

III. Proposed Methods

In the domain generalization problem, we assume that we are provided with n source domains $\mathcal{D} = [\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n]$, where $\mathcal{D}_i$ is the ith source domain containing data-label pairs $\left(\mathbf{x}_i^j, y_i^j\right)$. Particularly, j denotes the layer number within the ith domain. The neural network is required to learn $f : \mathbf{x} \rightarrow y$

and performs well to an unseen test domain with different statistics to the training domains. However, knowledge of the test domain is not available during the training stage. To achieve that, in this work, we aim to train a robust set of a feature extractor F_T and a detector D_T. We assume that the parameters of F_T and D_T are ϕ and θ, respectively. Thus, after the training, the model is expected as $f(\mathbf{x}) = \theta(\phi(\mathbf{x}))$. The overall proposed domain generalization method is illustrated in Fig. 1.

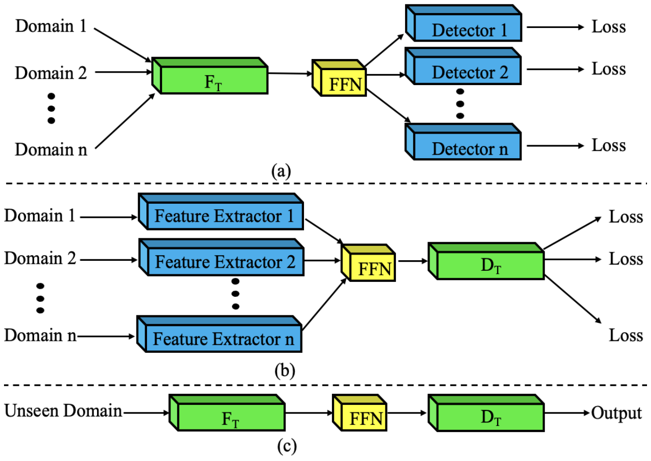

Fig. 1. Proposed training and test strategies of the proposed domain generalization method. (a) The target feature extractor F_T is sequentially trained with n detectors. Losses are calculated from each pair of network components with one specific domain to further boost the domain generalization performance of the parameter ϕ. (b) The target detector F_T is sequentially paired with n feature extractors. The parameter θ is further trained due to rich extracted features from different domains. (c) In the test stage, the trained F_T and D_T with the feature fusion network are combined as the final network and evaluated with unseen domains.

A. Feature Extractor Training

In the first stage, i.e., Fig. 1 (a), we aim to train ϕ to learn robust features that test data from unseen domains can be well processed by a detector that has never processed this domain before. To achieve that, we propose two loss terms to train the target feature extractor.

The first loss term is domain-specific loss of the target feature extractor $\mathcal{L}_{\text{ds}}^{\phi}$ that describes the global ϕ training to improve domain generalization performance. The optimisation task for $\mathcal{L}_{\text{ds}}^{\phi}$ can be presented as:

$$\underset{\phi,[\theta_1,\ldots,\theta_n]}{\operatorname{argmin}} \mathbb{E}_{\mathcal{D}_i\sim\mathcal{D}}\left[\mathbb{E}_{(\mathbf{x}_i,y_i)\sim\mathcal{D}_i}\left[\mathcal{L}_{\text{at}}^{\phi}\left(y_i,\theta_i\left(\phi\left(\mathbf{x}_i\right)\right)\right)\right]\right] \quad (1)$$

where $\theta_1,\ldots,\theta_n$ indicates parameters of different detectors that are trained with ϕ. The attack loss of ϕ is presented as $\mathcal{L}_{\text{at}}^{\phi}$. During the training ϕ, input data $\mathbf{x}_i$ is attacked with different error rates. The FFN obtains the feature maps and maps them into a fused latent space with rich information from the high- and low-level representations. Moreover, each paired detector is required to make correct predictions with the specific domain by using the fused latent space. Particularly, the parameter ϕ is penalized whenever different detectors output the wrong predictions with various source domains. Therefore, by optimising the $\mathcal{L}_{\text{ds}}^{\phi}$, ϕ is trained to extract more generalized features to help a detector recognize data that is from unseen domains.

The second loss term is attack loss $\mathcal{L}_{\text{at}}^{\phi}$ to train each pair of ϕ and θ_i, which improves the attack detection performance. To explicitly introduce $\mathcal{L}_{\text{at}}^{\phi}$, the model architecture is shown in Fig. 2.

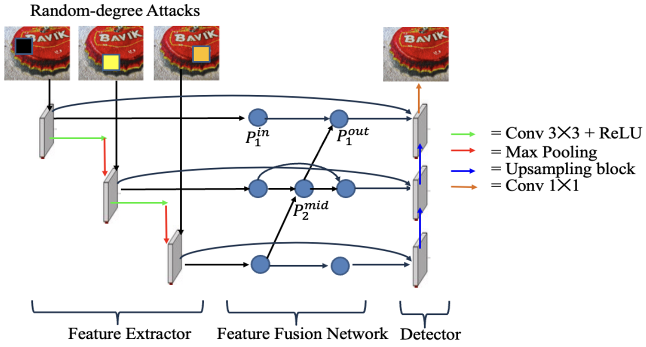

Fig. 2. Model architecture. The proposed MM-BiFPN network consists of a feature extractor, a FFN, and a detector. Each feature extractor's respective layers are concatenated and act as input for FFN layers for feature fusing. Then, fused features obtained from the FFN are transmitted into the detector to detect the attacks and produce a reconstructed image.

Different from conventional attack detection methods [16], [21], in the data pre-processing stage, each image sample are attacked with random error rates and individually fed to each layer. Then, feature maps are extracted from the attacked and clean images. The detector obtains the fused feature from the FFN and produces the reconstructed image. The loss $\mathcal{L}_{\text{at}}^{\phi}$ can be calculated between reconstructed and clean images as:

$$\mathcal{L}_{at}^{\phi} = \sum_{j=1}^{N} \ell\left(y_i, \theta_i\left(\phi\left(\mathbf{x}_i^j\right)\right)\right) \quad (2)$$

where $\ell(\cdot)$ denotes the cross-entropy loss. Therefore, we exploit the global loss, i.e., $\mathcal{L}_{\text{ds}}^{\phi}$ to train the parameter ϕ with multiple θ_is, while we use the local loss, i.e., $\mathcal{L}_{\text{at}}^{\phi}$ to train each pair of ϕ and θ_i.

B. Feature Fusion Network

In this work, inspired from the BiFPN [11], we propose a feature fusion network (FFN) between the feature extractor and detector for an efficient feature fusion.

As shown in Fig. 2, the images are attacked with different random error rates and fed into layers of the feature extractor. After extracting feature maps from attacked and clean images, we aim to efficiently recover lost information from the high-level representations with a simple network. Conventional feature fusion methods commonly focus on fusing with different resolutions [11], [21]. However, we observe that the attack detection performance is still limited, which will be further confirmed in Section IV.C. To address this issue, we firstly attack a image with different error rates. To the best of our knowledge, this is the first feature fusion of different error rates work in imperceptible adversarial attack detection.

To achieve that, we aim to exploit these feature maps for a feature fusion with the proposed FFN. In the proposed method, each layer of the feature extractor is concatenated and provides the input for each FFN layer. In the FFN training, the features maps are fused in both bottom-up and top-down directions, thus the FFN can effectively learn the features from clean and attacked images.

In the basic architecture i.e., FFN containing only three layers. We define the input, mid and output units as P_m^{in}, P_m^{mid} and P_m^{out} , respectively. As shown in Fig. 2, the output of each FFN layer can be calculated as follows:

$$\begin{aligned} P_1^{\text{out}} &= \operatorname{Conv}\left(P_1^{in} \oplus \operatorname{Resize}\left(P_2^{\text{mid}}\right)\right) \\ P_2^{\text{mid}} &= \operatorname{Conv}\left(P_2^{\text{in}} \oplus \operatorname{Resize}\left(P_3^{\text{in}}\right)\right) \\ P_2^{\text{out}} &= \operatorname{Conv}\left(P_2^{\text{mid}} \oplus \operatorname{Resize}\left(P_2^{\text{in}}\right)\right) \end{aligned} \tag{3}$$

where Conv and $\oplus$ are the convolution layer and concatenation operation, respectively. Moreover, the resize function is exploited to match the resolution size to combine the inputs.

C. Detector Training

Similar to the training for the feature extractor, in the second stage i.e., Fig. 1 (b), we aim to train θ to detect the attacks from unseen domains with extracted features from trained ϕ. It is highlighted that the trained ϕ and $\theta_1, \ldots, \theta_n$ from Fig. 1 (a) are not available in this stage. Therefore, the training of θ and ϕ can be processed in parallel to reduce the training time. To improve domain generalization performance of θ, we propose two loss terms to train the target detectors with different feature extractors.

The first loss term is domain-specific loss of the target detector $\mathcal{L}_{\text{ds}}^{\theta}$ that describes the global θ training to improve domain generalization performance. We optimise:

$$\underset{\theta,[\phi_1,\ldots,\phi_n]}{\operatorname{argmin}} \ \mathbb{E}_{\mathcal{D}_i \sim \mathcal{D}}\left[\mathbb{E}_{(\mathbf{x}_i, y_i) \sim \mathcal{D}_i}\left[\ell\left(y_i, \theta\left(\phi_i\left(\mathbf{x}_i\right)\right)\right)\right]\right] \tag{4}$$

where parameters of different feature extractors that extract features for θ are presented as $\phi_1, \ldots, \phi_n$. In each pair of ϕ_i and θ training, feature maps are obtained from input source domains. Then, feature maps are fed into the FFN for a feature fusion from different levels of representations. The lost information from particular-level representations is efficiently recovered. Then, the fused feature representations from different feature extractors are sequentially fed into the target detector to detect the attacks. The loss is calculated by the ground truth and detection result from each pair of ϕ_i and θ to train θ. During the training, the detector θ is trained to be robust enough to process data $\mathbf{x}_i$ that has been encoded by a paired feature extractor. Therefore, the domain generalization performance of θ is improved.

The second loss term is attack loss of the detector. As the input of the detector, the fused feature is obtained from the FFN. For each detector layer, we concatenate three components, including the previous upsampled layer, the respective FFN layer, and the respective feature extractor layer outputs to produce the input for the next detector layer. Therefore, the loss $\mathcal{L}_{at}^{\theta}$ is calculated as:

$$\mathcal{L}_{at}^{\theta} = \sum_{j=1}^{N} \ell\left(y_i, \theta\left(\phi_i\left(\mathbf{x}_i^j\right)\right)\right) \tag{5}$$

Similarly, the global loss, i.e., $\mathcal{L}_{\text{ds}}^{\theta}$ is exploited to train the parameter θ with multiple ϕ_is, while we use the local loss, i.e., $\mathcal{L}_{\text{at}}^{\theta}$ to train each pair of ϕ_i and θ. The pseudo-code of the training stage is summarized in Algorithm 1.

Algorithm 1 Overall algorithm.

```
Input: Datasets $\mathcal{D} = [\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_n]$, learning rate $\eta$, epoch $E_{\max}$
Initialize target network parameters $\phi$ and $\theta$
Initialize paired network parameters $\phi_1, \ldots, \phi_n$ and $\theta_1, \ldots, \theta_n$
Initialize hyper parameter $\alpha$
for $i \in (1, \ldots, n)$ do
    for $E = 1, 2, ..., E_{\max}$ do
        $\phi = \phi - \alpha \bigtriangledown_{\theta_i} (\mathcal{L}_{\text{at}}^{\phi})$
    end for
    $\phi = \phi - \alpha \bigtriangledown_{\theta_i} (\mathcal{L}_{\text{ds}}^{\phi})$
    for $E = 1, 2, ..., E_{\max}$ do
        $\theta = \theta - \alpha \bigtriangledown_{\phi_i} (\mathcal{L}_{\text{at}}^{\theta})$
    end for
    $\theta = \theta - \alpha \bigtriangledown_{\phi_i} (\mathcal{L}_{\text{ds}}^{\theta})$
end for
Output: $\phi$ and $\theta$
```

IV. Experiments

A. Datasets and Performance Measure

We perform experiments on several public datasets, including Canadian Institute For Advanced Research-10 (CIFAR-10), CIFAR-100 [28], ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [29], and ImageNet-R [30].

1) CIFAR-10 & CIFAR-100: As labeled subsets of the 80 million tiny images dataset [28], the CIFAR-10 and CIFAR-100 are commonly-used datasets in the object recognition task. Each dataset provides a set of 60,000 32x32 colour images. In our experiments, we randomly select 50,000 and 10,000 images from each dataset for the training and validation stages, respectively.

2) ILSVRC: As a subset of ImageNet, the ILSVRC dataset [29] contains 1,281,167 training images, 50,000 validation images, and 100,000 test images. In our experiments, we randomly select 50,000 and 10,000 images from the ILSVRC dataset for the training and validation stages, respectively.

3) ImageNet-R: The ImageNet-R dataset [30] is a set of images labelled with ImageNet labels. In our experiments, we randomly select 10,000 images as the test data.

The adversarial samples from these datasets are constructed with FGSM, PGD, and SSAH attacks. We select these attacks because they are robust to novel adversarial attack detection and defense techniques [15], [21]. Each one is The error rate

TABLE I

ATTACK DETECTION RATIO WITH THE SAME ATTACK BETWEEN THE TRAINING AND TEST STAGES BUT UNSEEN **DATASET** (IMAGENET-R). EACH RESULT IS THE AVERAGE OF 10,000 EXPERIMENTS. **BOLD** INDICATES THE BEST RESULTS. *italic* SHOWS THE PROPOSED METHODS.

	Computation		Detection Ratio (%)		
Method	Para. (M)	Time (s)	FGSM	PGD	SSAH
MetaQDA [27]	37.9	495.9	55.1 ± 1.8	59.2 ± 2.3	48.8 ± 2.5
Epi-FCR [2]	62.7	728.4	57.3 ± 1.4	60.0 ± 1.6	49.4 ± 0.7
Adversarial Training [16]	8.2	102.1	57.8 ± 1.3	61.1 ± 1.1	49.5 ± 1.7
CCAT [31]	36.5	454.7	57.5 ± 1.2	61.4 ± 0.8	48.2 ± 0.9
L-RED [20]	78.5	811.3	59.9 ± 0.9	62.3 ± 1.2	53.1 ± 1.5
BulletTrain [32]	2.2	**58.3**	60.7 ± 1.9	64.6 ± 1.4	58.5 ± 1.3
Sim-DNN [21]	134.9	1291.6	64.5 ± 1.6	66.9 ± 1.9	59.2 ± 1.1
DGAD (3)	**2.1**	99.6	67.3 ± 0.9	69.4 ± 1.1	65.6 ± 0.8
DGAD (4)	4.8	141.7	69.5 ± 0.7	72.2 ± 1.0	69.9 ± 0.5
DGAD (5)	6.9	168.0	**75.0 ± 0.4**	**76.3 ± 0.5**	**72.5 ± 0.3**

is randomly set from 0.01 to 0.04 in both the training and test stages.

In the experiment, the detection rate (DR) [21] is used as the performance measure.

$$DR(\%) = \frac{TP + TN}{TP + TN + FP + FN} \times 100 \tag{6}$$

where TP and TN are true positive and true negative results, and FP and FN are false positive and false negative results.

B. Competitors and Model Configuration

We compare the proposed method with domain generalization methods and attack detection methods. The MetaQDA is a Bayesian meta-learning based domain generalization method with wide residual networks (WRN-28-10) [33]. As the configuration with the best performance in [2], the Epi-FCR combines the vanilla aggregation loss, domain-specific loss, episodic loss, and random classifier loss implemented on a Resnet18 [34]. For attack detection methods, Adver encourages a Resnet56 network to learn latent representations that better distinguish adversarial examples from normal ones [16]. L-RED presents a Lagrangian-based RED algorithm with an arbitrary number of source classes on a Resnet18-DNN network [20]. As an imperceptible adversarial attack detection method, [21] introduces a similarity layer, a conditional probability layer, and a prototype identification layer on a DNN with a VGG-16 [35] as the feature extractor. BulletTrain and confidence-calibrated adversarial training (CCAT) exploit WRN-40-2 and WRN-10-28 as backbones, respectively [31], [32].

In the proposed generalization attack detection (DGAD) model training, different network components help to improve domain generalization, but lose some performance due to the forgetting effect. Therefore, we combine the training features in the current domain, i.e., $\mathcal{D}_i$, with trained features from the previous domain, i.e., $\mathcal{D}_{i-1}$, by using concatenation and mean-pooling operations. The proposed model is trained by using the M-SGD optimizer with a learning rate set empirically to 0.0008. The batch size is set to 32. We train the networks for 100 epochs. All the experiments are run on the High End Computing (HEC) Cluster with Tesla V100 GPUs.

C. Results

1) Domains with Seen Attacks: In the first experiment, we use the same attack but different datasets in domains to evaluate and compare the detection ratio of comparison and proposed methods in Table I. The number of training domains is set to 3 corresponding to training datasets, i.e., CIFAR-10, CIFAR-100, and ILSVRC. To show the efficiency of the proposed method, the parameters and training time of the models are compared in Table I. Moreover, the number of layers is presented in brackets.

From Table I, it can be observed that: (1) In all the evaluated models, the proposed DGAD methods with different numbers of layers offer the best effectiveness. For example, when evaluating the model with the PGD attack, the DGAD model with five layers achieves 25.9% better accuracy than Sim-DNN. The reason is that the proposed FFN fuses feature information from different resolutions and error rates, which recovers the lost information than conventional methods. Moreover, the proposed domain generalization method train parameters ϕ and θ robust to unseen domains. (2) The proposed DGAD models offer the best trade-off between performance and model size. These results suggest that the proposed DGAD method is quite promising for imperceptible adversarial attack detection.

2) Domains with Unseen Attacks: In the second experiment, to evaluate and compare the detection performance in a more challenging case, we use both different attacks and datasets in domains between the training and test stages in Table II. Different from Section IV. C. 1), the number of training domains is set to 2. Each domain contains one dataset (CIFAR-10, CIFAR-100) and one attack type (FGSM, PGD, SSAH). For example, we train models with two domains containing PGD and SSAH, respectively, while test models with the FGSM attack.

We can observe from Table II that the proposed DGAD models achieve better performance than previous models. Moreover, compared to Table II, it is possible to note that the proposed DGAD models performance tends to fall less than the previous models when less training data applied and unseen attack type evaluated. For example, the proposed model with 5 layers drops from 75.0% to 73.8%, while the MetaQDA

TABLE II
ATTACK DETECTION RATIO WITH UNSEEN **ATTACK AND DATASET** IN THE TEST STAGE. EACH RESULT IS THE AVERAGE OF 10,000 EXPERIMENTS. **BOLD** INDICATES THE BEST RESULTS. *italic* SHOWS THE PROPOSED METHODS.

	Detection Ratio (%)		
Method	FGSM	PGD	SSAH
MetaQDA [27]	50.4 ± 2.0	55.5 ± 2.1	43.7 ± 2.9
Epi-FCR [2]	56.9 ± 1.6	59.4 ± 1.6	49.1 ± 0.8
Adversarial Training [16]	53.2 ± 1.9	57.6 ± 1.4	45.5 ± 1.8
CCAT [31]	54.7 ± 1.5	59.1 ± 1.2	48.0 ± 1.5
L-RED [20]	56.1 ± 1.4	58.8 ± 1.5	48.2 ± 2.1
BulletTrain [32]	58.2 ± 1.9	58.9 ± 1.8	48.9 ± 1.6
Sim-DNN [21]	60.8 ± 1.7	63.3 ± 2.4	55.2 ± 1.5
DGAD (3)	65.8 ± 0.8	68.1 ± 1.3	64.0 ± 1.0
DGAD (4)	68.9 ± 0.7	71.0 ± 1.3	69.1 ± 0.7
DGAD (5)	**73.8** ± **0.6**	**73.2** ± **0.9**	**69.5** ± **0.7**

model drops from 55.1% to 50.4%.

3) Ablation Study: In this section, we investigate the effectiveness of each contribution based on the ImageNet-R dataset. The cross mark 7 for $\mathcal{L}_{\text{ds}}^{\phi}+\mathcal{L}_{\text{ds}}^{\theta}$ means we only use one paired component to train the feature extractor or detector, respectively. The ablation study is presented in Table III and the experimental setting is the same as Table I.

TABLE III
ABLATION STUDY OF THE THREE CONTRIBUTIONS IN THE PROPOSED METHOD. EACH RESULT IS THE AVERAGE OF 30,000 EXPERIMENTS (10,000 IMAGES×3 ATTACKS).

Ablation Settings			DR (%)
$\mathcal{L}_{\text{ds}}^{\phi}+\mathcal{L}_{\text{ds}}^{\theta}$	$\mathcal{L}_{\text{at}}^{\phi}+\mathcal{L}_{\text{at}}^{\theta}$	FFN	
✗	✗	✗	33.5
✓	✗	✗	53.7
✗	✓	✗	59.6
✗	✗	✓	57.0
✗	✓	✓	61.8
✓	✗	✓	70.9
✓	✓	✗	63.5
✓	✓	✓	**74.6**

Initially, the effectiveness of the domain-specific losses is studied. Compared to the baseline, the detection performance is significantly improved by adding $\mathcal{L}_{\text{ds}}^{\phi}$ and $\mathcal{L}_{\text{ds}}^{\theta}$. The reason is the performance degradation due to the domain transfer is reduced by the proposed domain generalization method. The parameters ϕ and θ are trained robust by using different extracted features from ϕ_i and predictions from θ_i.

Moreover, the experiment is performed by adding the proposed attack losses. As the most influential contribution, the $\mathcal{L}_{\text{at}}^{\phi}$ and $\mathcal{L}_{\text{at}}^{\theta}$ play the most important role in the attack detection. Different from the conventional attack detection approaches, the proposed method attacks the training sample with different error rates. The loss between the attacked and original images is calculated to train ϕ and θ to learn rich features and detect attacks from the fused feature, respectively. Therefore, the attack detection performance of the model with $\mathcal{L}_{\text{at}}^{\phi}$ and $\mathcal{L}_{\text{at}}^{\theta}$ is further boosted than the baseline.

The final experiment in the ablation study is performed by adding the FFN that obtains the feature maps from different layers of the feature extractor. Different from conventional BiFPN [11], the proposed FFN fuses features with different error rates in each image to boost the attack detection performance of the model. Therefore, the attack detection performance of the model with the FFN is further improved as shown in Table III.

Furthermore, the visualizations are given in Fig. 3, which are related to the reconstructions after detecting attacks of three randomly selected images from the ImageNet-R dataset. After comparing the reconstructed images with the original and attacked images, it can be observed that the reconstructions obtained via the proposed method, i.e., Fig. 3 (d)&(e), are closer to original images, which again confirms the efficacy of the proposed method.

V. Conclusion

In this paper, we proposed a domain generalization method to solve two challenges in cross-domain imperceptible adversarial attack detection. To accomplish this, we first trained the feature extractor and detector with a partner who was well-tuned for different domains, i.e., datasets and attacks. The feature extractor and detector were trained with the domain-specific loss so that the trained model's robustness was improved to unseen domains. Moreover, we proposed a feature fusion network to feature maps from images that were attacked with different error rates. Our experiments demonstrate the efficacy of our training approach in unseen domains, and the computational complexity is reduced.

Acknowledgment

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

References

[1] 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.

[2] D. Li, J. Zhang, Y. Yang, C. Liu, Y.-Z. Song, and T. M. Hospedales, "Episodic training for domain generalization," *International Conference on Computer Vision (ICCV)*, 2019.

[3] H. Zhang, Y.-F. Zhang, W. Liu, A. Weller, B. Scholkopf, and E. P. Xing, "Towards principled disentanglement for domain generalization," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[4] Y. Li, Y. Sun, K. Horoshenkov, and S. M. Naqvi, "Domain adaptation and autoencoder based unsupervised speech snhancement," *Submitted to IEEE Transactions on Artificial Intelligence*, 2021.

[5] J. Dong, Y. Cong, G. Sun, B. Zhong, and X. Xu, "What can be transferred: unsupervised domain adaptation for endoscopic lesions segmentation," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[6] J. Dong, Y. Cong, G. Sun, Z. Fang, and Z. Ding, "Where and how to transfer: knowledge aggregation-induced transferability perception for unsupervised domain adaptation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14, no. 8, pp. 1 – 17, 2021.

[7] Y. Zhang, M. Li, R. Li, K. Jia, and L. Zhang, "Exact feature distribution matching for arbitrary style transfer and domain generalization," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[8] B. Li, Y. Shen, Y. Wang, W. Zhu, C. Reed, K. Keutzer, D. Li, and H. Zhao, "Invariant information bottleneck for domain generalization," *Association for the Advancement of Artificial Intelligence (AAAI)*, 2021.

[9] G. Ghiasi, T.-Y. Lin, R. Pang, and Q. V. Le, "Nas-fpn: learning scalable feature pyramid architecture for object detection," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

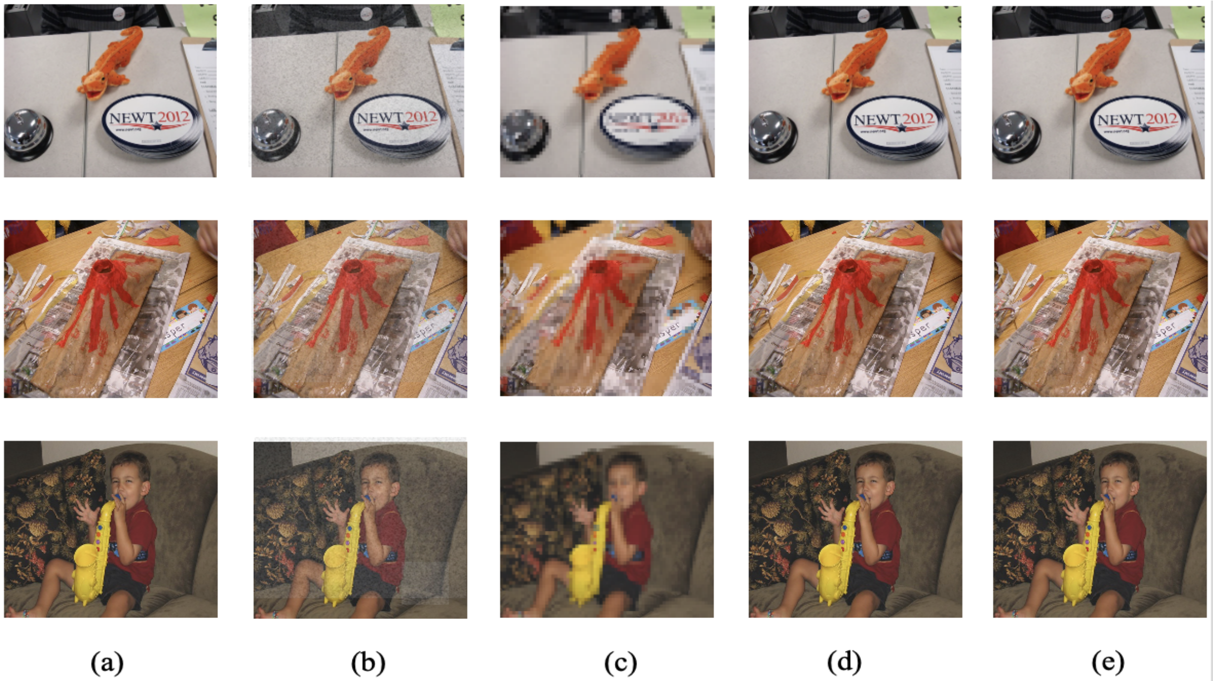

Fig. 3. Attack detection results: (a) original images; (b) attacked by PGD; (c) attacked by SSAH; (d) reconstruction from PGD attack detection; (e) reconstruction from SSAH attack detection.

[10] E. Real, A. Aggarwal, Y. Huang, and Q. V. Le, "Regularized evolution for image classifier architecture search," *Association for the Advancement of Artificial Intelligence (AAAI)*, 2019.

[11] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: scalable and efficient object detection," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[12] 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.

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

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

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

[16] T. Pang, C. Du, Y. Dong, and J. Zhu, "Towards Robust Detection of Adversarial Examples," *Conference on Neural Information Processing Systems (NeurIPS)*, 2018.

[17] 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.

[18] A. Nguyen and A. Tran, "WaNet - imperceptible warping-based backdoor attack," *International Conference on Learning Representations (ICLR)*, 2021.

[19] K. D. Doan and Y. Lao, "Backdoor attack with imperceptible input and latent modification," *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[20] Z. Xiang, D. J. Miller, and G. Kesidis, "L-Red: efficient post-training detection of imperceptible backdoor attacks without access to the training set," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

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

[22] J. Rony, L. G. Hafemann, L. S. Oliveira, I. B. Ayed, R. Sabourin, and E. Granger, "Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[23] Q. Lang, L. Zhang, W. Shi, W. Chen, and S. Pu, "Exploring implicit domain-invariant features for domain adaptive object detection," *IEEE Transactions on Circuits and Systems for Video Technology*, 2022.

[24] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Conference on Neural Information Processing Systems (NeurIPS)*, 2014.

[25] H. S. Le, R. Akmeliawati, and G. Carneiro, "Domain generalisation with domain augmented supervised contrastive learning," *Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI)*, 2020.

[26] Z. Wang and Z. Wang, "A domain transfer based data augmentation method for automated respiratory classification," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.

[27] X. Zhang, D. Meng, H. Gouk, and T. Hospedales, "Shallow bayesian meta learning for real-world few-shot recognition," *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

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

[29] A. Howard, E. Park, and W. Kan, "Imagenet object localization challenge," *International Journal of Computer Vision (IJCV)*, 2015.

[30] 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," *IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[31] D. Stutz, M. Hein, and B. Schiele, "Confidence-calibrated adversarial training: generalizing to unseen attacks," *International Conference on Machine Learning (ICML)*, 2020.

[32] W. Hua, Y. Zhang, C. Guo, Z. Zhang, and G. E. Suh, "BulletTrain: accelerating robust neural network training via boundary example mining," *Conference on Neural Information Processing Systems (NeurIPS)*, 2021.

[33] S. Zagoruyko and N. Komodakis, "Wide residual networks," *The British Machine Vision Conference (BMVC)*, 2016.

[34] K. M. He, X. Y. Zhang, S. Q. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for

large-scale image recognition," *International Conference on Learning Representations (ICLR)*, 2015.