BEYOND IMAGENET ATTACK: TOWARDS CRAFTING ADVERSARIAL EXAMPLES FOR BLACK-BOX DOMAIN

Anonymous authors
Paper under double-blind review

ABSTRACT

Adversarial examples have posed a severe threat to deep neural networks because they can even transfer without knowing the specific parameters of the target model. Currently, various works have paid great efforts to enhance the cross-model transferability, which assumes the substitute model is from the same domain as the target model, *e.g.*, all are ImageNet pre-trained models. However, in the real world, to ensure the security of the model deployment, the owner will never leak the relevant information to unauthorized users. Therefore, it is necessary to build a more practical black-box threat model. In this paper, with the only knowledge of the ImageNet domain, we propose a Beyond ImageNet Attack (BIA) to investigate the transferability towards the black-box domain (unknown classification task). Specifically, we leverage the generator model to learn the adversarial function with the aim of perturbing low-level features rather than optimizing domain-specific loss functions. Based on this framework, we further propose two variants to narrow the gap between the source and target domains from the data and model perspectives, respectively. Extensive experiments on coarse-grained and fine-grained domains demonstrate the effectiveness of our proposed methods. Notably, our methods can outperform state-of-the-art approaches by up to 7.71% (towards coarse-grained domain) and 25.91% (towards fine-grained domain) on average. The code is available in our supplementary materials.

1 INTRODUCTION

Deep neural networks (DNNs) have achieved remarkable success in image classification tasks in recent years. However, advances in the field of adversarial machine learning (Szegedy et al., 2014; Goodfellow et al., 2015) make DNNs no longer reliable. By adding well-designed perturbations on benign images (*a.k.a* adversarial attack), the resulting adversarial examples can easily fool state-of-the-art deep learning models, *i.e.*, leading to misclassification. To make the matter worse, this adversarial attack technique can even be applied in the physical world (Sharif et al., 2016; Kurakin et al., 2017a; Xu et al., 2020), which inevitably raises concerns about the stability of deployed models. Therefore, exposing as many "blind spots" of DNNs as possible to the spotlight is a top priority.

In general, the deployed models are mainly challenged with two threat models: white-box and black-box. For white-box threat model (Kurakin et al., 2017b; Moosavi-Dezfooli et al., 2016; Carlini & Wagner, 2017; Shi et al., 2019), the attacker can obtain complete knowledge of the target model, such as the whole training dataset and target model gradient for input. However, the deployed models are usually transparent to unauthorized users or have restricted access for security reasons. In this scenario, the black-box threat model (Goodfellow et al., 2015; Dong et al., 2018; 2019; Inkawhich et al., 2019; 2020) usually executes attacks by resorting to the cross-model transferability of adversarial examples. Specifically, perturbations are crafted via accessible white-box model (*a.k.a* substitute model), and the resulting adversarial examples sometimes can fool other black-box models as well. In this threat model, most of prior works (Yosinski et al., 2014; Poursaeed et al., 2018; Xie et al., 2019; Lin et al., 2020; Gao et al., 2020; 2021; Wang & He, 2021) assume that the source data for training the target model is available and mainly explore the transferability among the model that trained in the same data distribution. However, these works ignore a pivotal issue: *A model owner will never leak the relevant information of the deployed model, e.g., the training data distribution.*

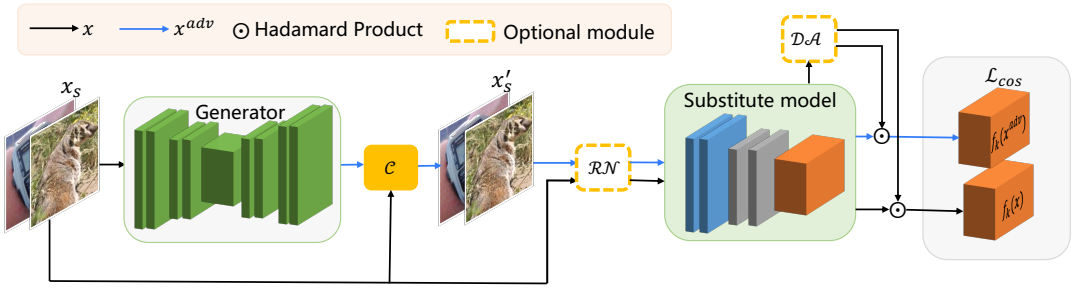

Figure 1: Our proposed generator framework with the aim of enlarging the cosine similarity of feature between benign image $\boldsymbol{x_s}$ and adversarial example $\boldsymbol{x'_s}$. Training data and substitute model are all from ImageNet domain. $\mathcal{RN}$ and $\mathcal{DA}$ are optional, which can further improve the transferability.

In such case, although the query-based black-box threat model (Papernot et al., 2016; Brendel et al., 2018; Chen et al., 2020; Li et al., 2021) can adjusting adversarial examples via the output of target model, the resource-intensive query budget is very expensive and easily alerts the model owner. Therefore, we need a more "practical" black-box threat model to tackle this concern, *i.e.*, even without any clue about the training data distribution as well as the pre-trained model based on this distribution. Intuitively, this threat model is more challenging to build for attackers but more threatening to model owners. To the best of our knowledge, a recent work called CDA (Naseer et al., 2019) is the first to attempt such an attack. Specifically, CDA seeks a transferable adversarial function via the generator network against the model trained in a different data distribution (ChestX-ray (Wang et al., 2017)). During inference, it aims to fool the black-box model trained in the ImageNet domain. Although CDA avoids expensive per-instance iterative optimization, its cross-domain transfer strength is still moderate. Besides, relying on small-scale datasets to train a generator may limit the generalizability of the threat model.

Considering that ImageNet is a large-scale dataset containing most of common categories in life and there are various off-the-shelf pre-trained models, one can easily dig out much useful information to build a strong but practical black-box threat model. Therefore, in this paper, we introduce a more effective **Beyond ImageNet Attack (BIA)** to enhance the cross-domain transferability of adversarial examples. To reflect the generalizability of our approach, we consider eight different image classification tasks (listed in Table 1). Figure 1 illustrates an overview of our method. In particular, we learn a flexible generator network $\mathcal{G}_\theta$ against ImageNet domain. Different from CDA, our method focuses on perturbing low-level features. Furthermore, we propose two variants based on BIA to narrow the gap between the source and target domains. Specifically, from the data perspective, we propose a random normalization ($\mathcal{RN}$) module to simulate different data distributions; from the model perspective, we propose a domain-agnostic attention ($\mathcal{DA}$) module to capture essential features for perturbing. In the inference phase, our $\mathcal{G}_\theta$ accepts images of any domain as input, and we only need one forward to craft powerful adversarial examples. Extensive experiments demonstrate the effectiveness of our proposed methods. Towards coarse-grained and fine-grained domain, we can outperform state-of-the-art approaches by up to 7.71% and 25.91% on average, respectively. Besides, in source domain, our methods can also enhancing the cross-model transferability.

Table 1: A comparison of datasets from different domains.

Dataset	Resolution	Type	Test size	Classes
ImageNet (Russakovsky et al., 2015)	224×224	-	50,000	1,000
CIFAR-10 (Krizhevsky, 2009)	32×32	coarse-grained	10,000	10
CIFAR-100 (Krizhevsky, 2009)	32×32	coarse-grained	10,000	100
STL-10 (Coates et al., 2011)	96×96	coarse-grained	8,000	10
SVHN (Netzer et al., 2011)	32×32	coarse-grained	26,032	10
CUB-200-2011 (Wah et al., 2011)	448×448	fine-grained	5,740	200
Stanford Cars (Krause et al., 2013)	448×448	fine-grained	8,041	196
FGVC Aircraft (Maji et al., 2013)	448×448	fine-grained	3,333	100

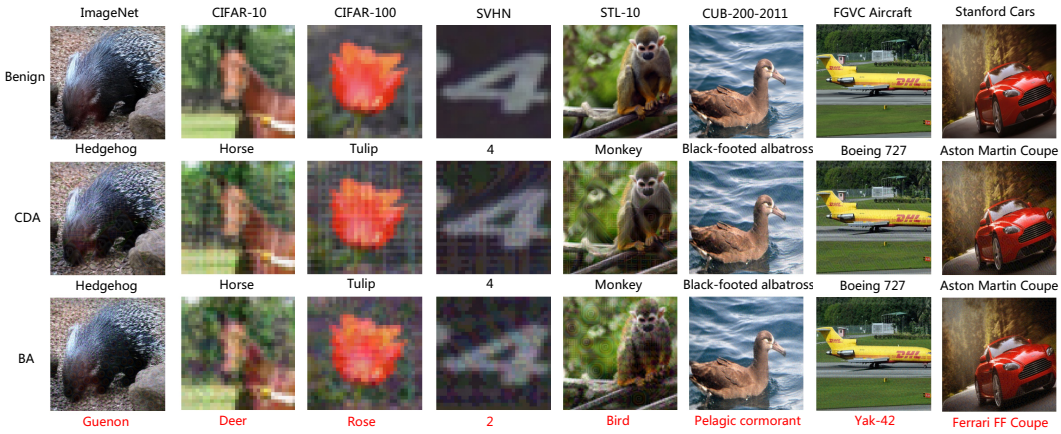

Figure 2: Adversarial examples crafted by CDA and our baseline BIA ($\epsilon = 10$). Both generator networks are trained against ImageNet pre-trained VGG-16 (Simonyan & Zisserman, 2015). Red highlighted labels represent misclassification.

2 Related Works

Iterative Optimization Approaches. Under the black-box threat model, iterative attack methods are a popular branch, which usually adopt task-specific loss or intermediate feature loss to craft adversarial examples. For the former, (Madry et al., 2018) extend (Goodfellow et al., 2015) to perform projected gradient descent from randomly chosen starting points inside ϵ-ball. (Dong et al., 2018) introduce momentum term to stable the update direction. Xie et al. (2019) apply random transformations of the input at each iteration, thus mitigating overfitting. (Gao et al., 2020) propose patch-wise perturbation to better cover the discriminative region. (Wu et al., 2020) explore the security weakness of skip connections (He et al., 2016b; Huang et al., 2017) to boost adversarial attacks.

Different from the methods mentioned above, intermediate feature-based methods focus on perturbing low-level features. For example, (Inkawhich et al., 2019) make the source image close to the target image in feature space. (Gao et al., 2021) attempt to measure similarities between feature maps by high-order statistics. (Lu et al., 2020) propose a dispersion reduction attack to make the low-level features featureless. (Naseer et al., 2020) design a self-supervised perturbation mechanism for enabling a transferable defense approach.

Generator-oriented Approaches. Compared with iterative optimization approaches, generator-oriented attacks are more efficient (*i.e.*, only need one inference) to generate adversarial examples. In this branch, (Baluja & Fischer, 2017) propose an adversarial transformation network to modify the output of the classifier given the original input. (Poursaeed et al., 2018) present trainable deep neural networks for producing both image-agnostic and image-dependent perturbations. (Naseer et al., 2019) leverage datasets from other domain instead of ImageNet to train generator networks against pre-trained ImageNet models, and inference is performed on ImageNet domain with the aim of fooling black-box ImageNet model. They also attempt a practical black-box threat model (from ChestX to ImageNet), and the attack success rate can outperform the result of Gaussian noise.

3 Transferable Adversarial Examples beyond ImageNet

In our paper, we assume ImageNet domain (including pre-trained model $f_s(\cdot)$ and dataset $\chi_s \in \mathbb{R}^{N\times H_s\times W_s}$) is the only available knowledge. Given any deep learning classifier $f_t(\cdot)$ trained in a specific dataset $\chi_t \in \mathbb{R}^{N\times H_t\times W_t}$, our goal is to craft adversarial perturbation $\boldsymbol{\delta_t} \in \mathbb{R}^{N\times H_t\times W_t}$ for benign image $\boldsymbol{x_t} \sim \chi_t$ so that can fool $f_t(\cdot)$ successfully, *i.e.*, $f_t(\boldsymbol{x_t}+\boldsymbol{\delta_t}) \neq f_t(\boldsymbol{x_t})$. Here we denote $\boldsymbol{x'_t} = \boldsymbol{x_t} + \boldsymbol{\delta_t}$ as the resulting adversarial example. Intuitively, building such a practice black-box threat model is very challenging. As shown in Table 1 and Figure 2, images from different domain vary greatly, such as resolution and classes.

Figure 3: **Left:** We calculate the data distribution for datasets from different domains. The result is the average over the three channels. **Right:** We show two intermediate feature maps (*Maxpool.3*) of VGG-16 (Simonyan & Zisserman, 2015) (trained in ImageNet domain) for the input image from CUB-200-2011 (Wah et al., 2011).

3.1 Generative Model

Prior works (Baluja & Fischer, 2017; Poursaeed et al., 2018; Naseer et al., 2019) have shown the powerful attack capability of the generative model. Therefore, our proposed method is also based on a generator $\mathcal{G}_\theta$ parameterized by θ to craft adversarial examples. The framework is illustrated in Figure 1. In the training phase, output adversarial examples $\boldsymbol{x'_s}$ are constrained in the ℓ_∞-ball of the input image $\boldsymbol{x_s}$ by clip module ($\mathcal{C}$), *i.e.*,

$$||\boldsymbol{x'_s} - \boldsymbol{x_s}||_\infty \leq \epsilon, \tag{1}$$

where ϵ is the maximum perturbation with the aim of ensuring $\boldsymbol{x_s}$ are minimally changed.

Considering that we want to transfer adversarial examples to other unseen domain, relying on the last layer with task-specific loss functions (*e.g.*, cross-entropy loss adopted in (Naseer et al., 2019)) might lead to overfitting in source domain. Instead, the intermediate layer of the DNN presumably extract general features which can share across different models (Yosinski et al., 2014). Therefore, our baseline BIA turns to **destroy the low-level features** of the substitute model at layer L as much as possible. Specifically, we decrease the cosine similarity between $\boldsymbol{x'_s}$ and $\boldsymbol{x_s}$ to make the feature featureless:

$$\theta^* = \arg\min_\theta \mathcal{L}_{cos}(f_s^L(\boldsymbol{x'_s}), f_s^L(\boldsymbol{x_s})). \tag{2}$$

In the inference phase, our generator $\mathcal{G}_{\theta^*}$ can directly craft adversarial examples $\boldsymbol{x'_t}$ for input images $\boldsymbol{x_t}$ from any other target domain:

$$\boldsymbol{x'_t} = clip(\min(\boldsymbol{x_t} + \boldsymbol{\epsilon}, \max(\mathcal{G}_{\theta^*}(\boldsymbol{x_t}), \boldsymbol{x_t} - \boldsymbol{\epsilon}). \tag{3}$$

The resulting adversarial examples are depicted in Figure 2. Compared with CDA, our BIA is more effective in both source (white-box) and target (black-box) domains. Yet, as shown in Figure 3, BIA still has some challenges for crafting more transferable adversarial examples:

- **Data Perspective:** The distribution (*i.e.*, mean and standard deviation) of source domain is largely different from any other target domain. For example, the standard deviation of ImageNet is about twice that of SVHN.
- **Model Perspective:** Although some feature map of $f_s^L(\cdot)$ can capture the object ($\in \chi_t$) for feature representation (*e.g.*, the first feature map in Figure 3), there are also some feature maps that are significantly noisy (*e.g.*, the second feature map in Figure 3).

To alleviate the concern of generating poor transferable adversarial examples that may arise from the above limitations, we propose two variants, equipped with random normalization ($\mathcal{RN}$) or domain-agnostic attention ($\mathcal{DA}$) modules, to deal with them, respectively.

3.2 Random Normalization Module

Generally, DNNs (Simonyan & Zisserman, 2015; He et al., 2016b; Huang et al., 2017) are usually equipped with normalization for input images[1] so that they can subject to the standard normal dis-

[1]For Pytorch pre-trained ImageNet model, input images should be normalized using mean = [0.485, 0.456, 0.406] and standard deviation = [0.229, 0.224, 0.225]

tribution. However, as illustrated in Figure 3 (left), the distribution of dataset from different domain can vary dramatically. Therefore, a target model owner may also adopt a different normalization module for training, which further enlarging the gap between source domain and target domain.

To address this limitation, commonly used strategy of label-preserving data augmentation (Krizhevsky et al., 2012; Simonyan & Zisserman, 2015; He et al., 2016a) is less effective because it has little effect on changing the distribution of the inputs (the experiment result is shown in Appendix A.1). Instead, fusing knowledge from different domains might be helpful. As shown in Table 3 of (Naseer et al., 2019), using training data from other domains against the pre-trained model in the source domain can enhance the transferability of adversarial examples towards the target domain. However, this is impractical under our threat model because we cannot get access to other domains. Therefore, in this section, we propose a random normalization ($\mathcal{RN}$) module to simulate different data distribution in the training phase:

$$\mathcal{RN}(\boldsymbol{x_s}) = \boldsymbol{\sigma} \cdot \frac{\boldsymbol{x_s} - \mu'}{\sigma'} + \boldsymbol{\mu}, \tag{4}$$

where $\boldsymbol{\sigma}$ and $\boldsymbol{\mu}$ are default standard deviation and mean vectors for ImageNet, and $\mu' \sim \mathcal{N}(\mu'_{mean}, \mu'_{std})$ and $\sigma' \sim \mathcal{N}(\sigma'_{mean}, \sigma'_{std})$ are two random scales sampled from Gaussian distribution[2]. Combined with $\mathcal{RN}$, and the update rule can be expressed as:

$$\theta^* = \arg\min_{\theta} \mathcal{L}_{cos}(f_s^L(\mathcal{RN}(\boldsymbol{x'_s})), f_s^L(\mathcal{RN}(\boldsymbol{x_s}))). \tag{5}$$

3.3 Domain-agnostic Attention Module

Prior works (Hansen & Salamon, 1990; Caruana et al., 2004; Dong et al., 2018) have demonstrated that ensemble methods can avoid getting trapped in the local optimum and improve performance. Since there are many feature maps at layer L and each of them can model the input, *i.e.,* extracts the features, we can also integrate them to produce a more robust feature representation $\mathcal{A}$, thus mitigating the impact of several noisy feature maps, *e.g.*, the second feature map of Figure 3. Specifically, we apply a channel-wise average pooling to the feature maps at layer L:

$$\mathcal{A} = \frac{|\sum_{i=0}^{C} f_s^L(\boldsymbol{x_s})|}{C}, \tag{6}$$

where C is the channel number of $f_s^L(\boldsymbol{x_s})$.

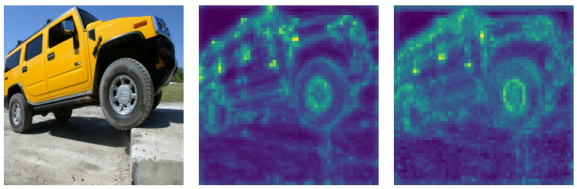

Figure 4: **Left:** An image from Stanford Cars (Krause et al., 2013). **Middle & Right:** We apply the channel-wise average pooling to the intermediate feature maps ($Maxpool.3$) of VGG-16 and ($Conv3_8$) of target model DCL (Chen et al., 2019) (backbone is Res-50), respectively.

As depicted in Figure 4, even if our source model is not trained in the target domain, the robust feature representation is still able to capture the essential feature of object very well and even similar to the one that derived from a completely different target model. Therefore, this robust feature representation can serve as a domain-agnostic attention ($\mathcal{DA}$) to enhance the cross-domain transferability of adversarial examples. Specifically, in the training phase, we leverage $\mathcal{A}$ to assign weight for each pixel of feature maps so that generator $\mathcal{G}_\theta$ can focus on perturbing the crucial region of the input. The resulting object function can be rewritten as:

$$\theta^* = \arg\min_{\theta} \mathcal{L}_{cos}(\mathcal{A} \odot f_s^L(\boldsymbol{x'_s}), \mathcal{A} \odot f_s^L(\boldsymbol{x_s})), \tag{7}$$

where $\odot$ is Hadamard product.

4 Experiments

Generator Architecture. Our generator $\mathcal{G}_\theta$ adopts the same architecture as (Naseer et al., 2019), which is a composite of downsampling, residual (He et al., 2016b) and upsampling blocks (more

[2] μ'_{mean}, μ'_{std}, σ'_{mean} and σ'_{std} are all hyper-parameters.

details can be found in Appendix A.3). The output size of $\mathcal{G}_\theta$ is equal to the input size. Thus, our $\mathcal{G}_\theta$ can accept any input image[3] with a resolution higher than 4×4.

Source (white-box) Domain. Our training data is the large-scale ImageNet (Russakovsky et al., 2015) training set which including about 1.2 million images. The generator is trained against four ImageNet pre-trained models. Specifically, we attack the output of $Maxpool.3$ for VGG-16 and VGG-19 (Simonyan & Zisserman, 2015), the output of $Conv3_8$ for ResNet152 (Res-152) (He et al., 2016b) and the output of $DenseBlock.2$ for DenseNet169 (Dense-169) (Huang et al., 2017). In this domain, we also consider another three pre-trained models, *i.e.*, DenseNet121 (Dense-121) (Huang et al., 2017), ResNet50 (Res-50) (He et al., 2016b) and Inception-v3 (Inc-v3) (Szegedy et al., 2016), to analyze the cross-model transferability. All models are available in the Torchvision[4].

Target (black-box) Domain. In addition to ImageNet (white-box domain), we also consider another seven black-box domains (shown in Table 1) which including four coarse-grained (CIFAR-10, CIFAR-100 (Krizhevsky, 2009), STL-10 (Coates et al., 2011) and SVHN (Netzer et al., 2011)) and three fine-grained (CUB-200-2011 (Wah et al., 2011), Standford Cars (Krause et al., 2013) and FGVC Aircraft (Maji et al., 2013)) classification tasks. For the fine-grained classification, we use DCL framework (Chen et al., 2019) with three different backbones: ResNet50 (Res-50) (He et al., 2016b), SENet154 and SE-ResNet101 (SE-Res101) (Hu et al., 2018). The pre-trained models for the coarse-grained classification are from Github[5], and each model is relatively shallow and simple.

Implementation Details. We compare our proposed methods with projected gradient descent (PGD) (Madry et al., 2018), diverse inputs method (DIM) (Dong et al., 2018; Xie et al., 2019), dispersion reduction (DR) (Lu et al., 2020), self-supervised perturbation (SSP) (Naseer et al., 2020) and cross-domain attack (CDA) (Naseer et al., 2019). The maximum perturbation ϵ is set to 10 for all experiments. Follow (Lu et al., 2020), we set the step size $\alpha = 4$ and the number of iterations $T = 100$ for all iterative methods. For DIM, we set the default decay factor $\mu = 1.0$ and the transformation probability $p = 0.7$. For CDA and our methods, we use Adam optimizer (Kingma & Ba, 2015) with a learning rate of 2e-4 and the exponential decay rate for first and second moments is set to 0.5 and 0.999, respectively. All generators in our experiments are trained for one epoch with the batch size 16. The Gaussian smoothing (gs) for CDA is applied by 3×3 Gaussian kernel. For our $\mathcal{RN}$ module, we set $\mu' \sim \mathcal{N}(0.50, 0.08)$ and $\sigma' \sim \mathcal{N}(0.75, 0.08)$.

Evaluation Metrics. To avoid cherry-picking, we use the top-1 accuracy on the whole test set (test size is shown in Table 1) after attacking to evaluate the performance of different methods.

4.1 The Effect of Selected Gaussian Distribution for Random Normalization

In this section, we conduct the experiment to select an optimal distribution for the μ' and σ' in Equation 4. Here we tune the mean of μ' and σ' from 0.25 to 0.75 with a granularity of 0.25. For the standard deviation of them, we fix it to 0.08 so that the sampled μ' and σ' can basically take values between 0 and 1 (according to the three-sigma rule).

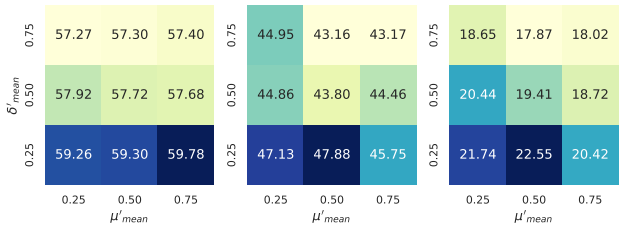

Figure 5: The average top-1 accuracy after attacking on coarse-grained (left), fine-grained (middle) and source (right) domains with different μ'_{mean} and σ'_{mean} for $\mathcal{RN}$.

We craft adversarial examples via VGG-16 and report the average top-1 accuracy in both source and black-box domains. From the result of Figure 5, we observe that a bigger σ'_{mean} can effectively improve the transferability. For example, if we increase σ'_{mean} from 0.25 to 0.75, the top-1 accuracy on fine-grained models can be further decreased by 3.16% on average. Although the influence of μ'_{mean} is relatively moderate when $\sigma'_{mean} = 0.75$, setting μ'_{mean} to 0.5 is usually better. Therefore, we set $\mu' \sim \mathcal{N}(0.50, 0.08)$ and $\sigma' \sim \mathcal{N}(0.75, 0.08)$ in our paper.

[3]Note that Inception-v3 (Szegedy et al., 2016) cannot inference low-resolution inputs, e.g., CIFAR-10.
[4]https://pytorch.org/vision/stable/models.html
[5]https://github.com/aaron-xichen/pytorch-playground

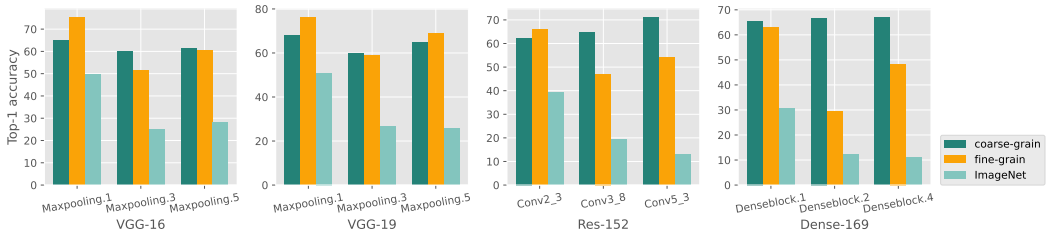

Figure 6: The average top-1 accuracy after attacking (the lower, the better) on coarse-grained, fine-grained and source domains. Our generator $\mathcal{G}_\theta$ is trained against different layers of ImageNet pre-trained VGG-16, VGG-19, Res-152 and Dense-169, respectively.

4.2 The Effect of Selected Layer for Attacking

In this section, we analyze the impact of different intermediate layers of the substitute model on the transferability of resulting adversarial examples. The results are illustrated in Figrue. 6.

In general, training against the shallow and middle layers yields more cross-domain transferable but less cross-model transferable adversarial examples than against the deep layer. For example, if the substitute model is Res-152, perturbing shallow layer $Conv2_3$ is more effective for transferring towards coarse-grained models and middle layer $Conv3_8$ is more effective for transferring towards fine-grained models. In contrast, perturbing deep layers like $Conv5_3$ can better reduce the average top-1 accuracy in the source domain. This demonstrates that low-level features are more similar across domains and high-level features are more domain-specific.

Table 2: Transferability comparisons on four coarse-grained classification tasks. Here we report the top-1 accuracy rate after attacking (the lower, the better). The generator $\mathcal{G}_\theta$ is trained in the ImageNet domain, and adversarial examples are within the perturbation budget of $\ell_\infty \leq 10$.

Model	Attacks	CIFAR-10	CIFAR-100	STL10	SVHN	Model	Attacks	CIFAR-10	CIFAR-100	STL10	SVHN
	Clean	93.78	74.27	77.59	96.03		Clean	93.78	74.27	77.59	96.03
VGG-16	PGD	79.63	48.02	74.32	94.66	Res-152	PGD	86.17	56.38	74.51	93.94
	DIM	77.16	44.75	72.74	91.53		DIM	80.50	48.03	71.2	90.87
	DR	72.49	39.04	72.56	93.27		DR	78.86	48.62	71.66	93.26
	SSP	68.54	33.63	72.77	93.98		SSP	75.54	42.38	72.66	92.63
	CDA	66.41	32.37	72.91	92.17		CDA	66.47	39.30	69.81	88.09
	CDA+gs	86.70	59.43	73.38	91.61		CD+gA	85.61	57.27	73.06	90.34
	BIA (Ours)	57.38	22.47	69.45	90.44		BIA (Ours)	65.49	33.48	69.91	89.46
	BIA+$\mathcal{DA}$ (Ours)	55.16	21.71	70.00	91.76		BIA+$\mathcal{DA}$ (Ours)	65.34	**32.68**	69.65	91.38
	BIA+$\mathcal{RN}$ (Ours)	**52.81**	**20.82**	**67.55**	**88.03**		BIA+$\mathcal{RN}$ (Ours)	**61.23**	32.84	**68.04**	**85.79**
VGG-19	PGD	79.15	47.73	74.71	94.86	Dense-169	PGD	84.55	54.29	74.55	93.83
	DIM	77.54	43.81	72.96	91.68		DIM	80.89	49.06	72.64	89.47
	DR	70.72	37.59	71.98	93.73		DR	78.24	48.67	70.75	93.2
	SSP	70.46	35.28	73.21	93.67		SSP	77.13	42.18	72.53	91.64
	CDA	81.60	51.53	71.43	92.64		CDA	67.75	35.03	69.00	88.76
	CDA+gs	88.55	61.90	73.64	92.18		CDA+gs	85.01	54.71	72.61	88.69
	BIA (Ours)	57.88	23.12	69.84	88.89		BIA (Ours)	72.02	38.99	69.80	86.12
	BIA+$\mathcal{DA}$ (Ours)	57.26	23.04	70.16	90.08		BIA+$\mathcal{DA}$ (Ours)	71.69	38.95	70.60	88.02
	BIA+$\mathcal{RN}$ (Ours)	**54.47**	**22.61**	**68.23**	**88.08**		BIA+$\mathcal{RN}$ (Ours)	**66.67**	**34.41**	**68.79**	**81.54**

4.3 Transferability Comparsions on Black-box Domain

In this section, we first conduct the experiment for coarse-grained classification in Section 4.3.1 and then report the results for fine-grained classification in Section 4.3.2. For CDA and our proposed methods, the generator $\mathcal{G}_\theta$ is trained in the ImageNet domain, and inference is directly performed on the target domain. For the results of $\mathcal{RN}$ + $\mathcal{DA}$ variant, we leave it in Appendix A.2.

4.3.1 Results on Coarse-grained Classification

In this section, we craft transferable adversarial examples towards coarse-grained classification tasks. The results are shown in Table 2, where we leverage four ImageNet pre-trained models, including VGG-16, VGG-19, Res-152 and Dense-169, to train generators, respectively.

From Table 2, a first glance shows that our proposed methods consistently outperform state-of-the-art approaches by a large margin. For example, if the substitute model is VGG-16, the most effective method CDA still obtain a top-1 accuracy of 66.41% for CIFAR-10 after attacking, while our baseline BIA and $\mathcal{DA}$ variant can effectively decrease it to **57.38%** and **55.16%**, respectively. Among all tasks, the SVHN domain is the most difficult to attack. In this domain, $\mathcal{DA}$ is less effective, but with the help of our proposed $\mathcal{RN}$ variants, we can significantly enhancing the transfeability. Notably, if the substitute model is Dense-169, $\mathcal{RN}$ variant can further decrease the top-1 accuracy from 86.12% to **81.54%**.On average, our $\mathcal{RN}$ variant can significantly outperforms state-of-the-art CDA by **7.71%**. This demonstrates that our proposed $\mathcal{RN}$ module is effective in improving the generalization of the generator to cope with different distributions of inputs.

Table 3: Transferability comparisons on three fine-grained classification tasks. Here we report the top-1 accuracy rate after attacking (the lower, the better). The generator $\mathcal{G}_\theta$ is trained in ImageNet domain and adversarial examples are within the perturbation budget of $\ell_\infty \leq 10$.

Model	Attacks	CUB-200-2011			Standford Cars			FGVC Aircraft		
		Res-50	SENet154	SE-Res101	Res-50	SENet154	SE-Res101	Res-50	SENet154	SE-Res101
	Clean	87.35	86.81	86.56	94.35	93.36	92.97	92.23	92.08	91.9
VGG-16	PGD	80.65	79.58	80.69	87.45	89.04	90.30	84.88	83.92	82.15
	DIM	70.02	62.86	70.57	74.72	78.10	84.33	73.54	66.88	62.38
	DR	81.08	82.05	82.52	90.82	90.59	91.12	84.97	87.55	85.54
	SSP	62.27	60.44	71.52	58.02	75.71	83.02	54.91	68.74	63.79
	CDA	69.69	62.51	71.00	75.94	72.45	84.64	71.53	58.33	63.39
	CDA+gs	70.19	63.19	68.92	85.03	79.52	83.52	78.55	65.62	68.38
	BIA (Ours)	32.74	52.99	58.04	39.61	69.90	70.17	28.92	60.31	51.43
	BIA+$\mathcal{DA}$ (Ours)	**25.00**	**40.27**	**53.24**	22.24	**59.48**	**61.52**	**15.36**	**47.91**	**40.83**
	BIA+$\mathcal{RN}$ (Ours)	26.13	46.15	55.07	**20.61**	62.64	63.38	16.50	52.54	45.48
VGG-19	PGD	80.98	79.00	80.60	87.54	88.87	90.56	84.70	84.01	83.35
	DIM	69.93	61.60	70.90	75.02	78.55	84.63	74.59	67.69	65.26
	DR	80.83	81.57	81.95	91.00	90.23	91.15	84.43	85.96	84.57
	SSP	62.94	58.34	70.45	61.90	76.27	83.77	58.78	69.52	66.88
	CDA	59.48	61.08	68.50	58.53	70.70	80.70	59.26	52.24	62.26
	CDA+gs	67.88	59.42	67.57	82.83	78.61	82.64	79.36	65.89	68.35
	BIA (Ours)	48.90	52.33	56.47	66.34	72.45	75.08	50.95	54.04	51.79
	BIA+$\mathcal{DA}$ (Ours)	**27.46**	**37.61**	**50.14**	**35.27**	**61.40**	**64.41**	**17.97**	**45.81**	**44.01**
	BIA+$\mathcal{RN}$ (Ours)	31.77	43.41	51.09	42.81	68.90	66.27	27.75	52.48	45.57
Res-152	PGD	73.21	75.89	76.11	83.99	86.89	88.24	79.00	79.30	75.64
	DIM	56.30	59.35	63.36	67.88	76.37	79.77	64.21	62.95	54.49
	DR	77.58	82.07	80.60	87.96	90.11	90.67	77.89	82.27	80.08
	SSP	47.64	66.17	67.90	53.29	79.09	83.45	58.35	73.36	69.34
	CDA	45.15	53.69	52.86	57.72	63.09	73.05	64.87	46.74	59.11
	CDA+gs	59.32	64.24	60.08	81.15	82.91	82.56	75.52	73.72	66.58
	BIA (Ours)	43.55	49.50	55.54	31.65	54.12	67.21	38.49	32.46	51.19
	BIA+$\mathcal{DA}$ (Ours)	25.80	**39.71**	52.83	33.43	52.41	68.03	**27.51**	**27.42**	47.28
	BIA+$\mathcal{RN}$ (Ours)	**23.54**	40.13	**51.36**	**12.39**	**47.92**	**60.59**	32.34	32.85	**46.89**
Dense-169	PGD	79.29	81.14	79.74	87.66	90.13	89.80	83.23	84.31	81.24
	DIM	63.17	62.01	65.96	72.88	78.29	81.25	68.89	65.26	54.79
	DR	74.84	78.89	77.93	86.54	88.82	89.52	77.80	78.73	74.47
	SSP	41.80	49.95	59.72	26.65	68.71	74.36	16.80	55.78	44.07
	CDA	52.92	60.96	57.04	53.64	73.66	75.51	62.23	61.42	59.83
	CDA+gs	60.86	61.34	60.10	74.95	76.35	78.86	72.94	68.68	64.66
	BIA (Ours)	21.79	**29.29**	39.13	9.58	44.46	49.06	8.04	27.84	33.87
	BIA+$\mathcal{DA}$ (Ours)	12.36	29.31	**35.78**	**4.56**	**40.85**	**31.82**	3.90	**13.59**	**14.55**
	BIA+$\mathcal{RN}$ (Ours)	**10.67**	32.50	46.62	7.76	44.14	42.76	**3.81**	34.20	27.00

4.3.2 Results on Fine-grained Classification

We also analyze the transferability of adversarial examples towards fine-grained classification tasks. For each task, three black-box models with different backbones trained via the DCL framework are the target. The results are summarized in Table 3, where the leftmost column is the substitute model and the top row shows the target model.

In this scenario, the performance gap between the existing state-of-the-art algorithms and our proposed methods is further enlarged. On average, even the vanilla BIA can notably decrease the average top-1 accuracy to **46.65%**, while CDA, SSP, DR, DIM and PGD are still with the high average top-1 accuracy of 63.21%, 62.33%, 84.02%, 69.13% and 83.14% after attacking, respectively. Furthermore, by adding our proposed $\mathcal{DA}$ and $\mathcal{RN}$ modules in the training phase, the resulting generator $\mathcal{G}_{\theta^*}$ can craft more transferable adversarial examples. On average, our proposed $\mathcal{DA}$ and

$\mathcal{RN}$ variant can further reduces the top-1 accuracy from 46.65% to **36.42%** and **39.33%**, which remarkably outperform SSP by up to **25.91%** and **22.99%**.

4.4 Transferability Comparisons on Source Domain

Although our proposed methods are mainly designed for improving the threat of adversarial examples towards black-box domain, they are also effective for improving cross-model black-box transferability in the white-box domain. For example, by training against Dense-169, DIM still obtains a top-1 accuracy of 21.44% on Inc-v3, while our $\mathcal{RN}$ variant can achieve a relatively low top-1 accuracy of **14.24%** on it. Compared with vanilla BIA, $\mathcal{RN}$ module further decreases the top-1 accuracy on Dense-169 (white-box) and Inc-v3 (black-box) by 2.4% and 24.34%, respectively. This result demonstrates that our $\mathcal{RN}$ module can avoid getting stuck in the local optimum of a specific model when training on a large-scale dataset.

For $\mathcal{DA}$ variant, since our target domain and training domain are identical, $\mathcal{DA}$ variant is naturally able to improve the transferability as well. As shown in Table 4, compared with BIA, it can further degrade the top-1 accuracy from 18.45% to **13.47%** on average.

Table 4: Transferability comparisons on ImageNet (source domain). Here we report the top-1 accuracy rate after attacking (the lower, the better). The generator $\mathcal{G}_\theta$ is trained in ImageNet domain ("*" denotes white-box attack) and adversarial examples are within the perturbation budget of $\ell_\infty \leq 10$.

	Attack	VGG-16	Dense-169	VGG-19	Res-50	Res-152	Dense-121	Inc-v3
	Clean	70.14	75.75	70.95	74.61	77.34	74.22	76.19
VGG-16	PGD	2.49*	53.22	4.27	45.52	58.69	48.28	61.08
	DIM	3.32*	**22.72**	3.39	19.13	33.03	18.50	30.05
	DR	20.95*	67.91	43.59	64.90	70.00	64.73	69.22
	SSP	0.95*	39.56	3.42	26.22	41.68	34.45	47.46
	CDA	**0.40***	42.67	**0.77**	36.27	51.05	38.89	54.02
	CDA+gs	12.10*	57.09	20.48	51.87	60.83	52.21	55.12
	BIA (Ours)	1.55*	32.35	3.61	25.36	42.98	26.97	41.20
	BIA+$\mathcal{DA}$ (Ours)	1.04*	24.52	2.07	18.63	36.43	19.97	34.54
	BIA+$\mathcal{RN}$ (Ours)	1.44*	25.96	2.58	**16.52**	**31.80**	**18.25**	**28.54**
Dense-169	PGD	38.51	5.03*	40.53	33.91	44.97	21.18	58.30
	DIM	12.31	5.25*	13.20	8.98	12.93	5.91	21.44
	DR	38.45	23.99*	41.59	50.19	58.70	49.95	63.70
	SSP	11.53	1.32*	12.54	12.97	25.66	9.74	25.58
	CDA	7.26	**0.63***	7.91	6.46	15.56	5.13	43.78
	CDA+gs	27.98	22.95*	28.56	31.52	43.67	31.66	49.18
	BIA (Ours)	4.76	6.45*	7.15	6.97	13.83	6.60	38.58
	BIA+$\mathcal{DA}$ (Ours)	**3.17**	3.32*	**4.09**	**4.44**	**5.85**	**3.98**	26.51
	BIA+$\mathcal{RN}$ (Ours)	3.66	4.05*	5.23	6.91	13.25	4.21	**14.24**

5 Conclusion

In this paper, we present a more practice black-box threat model for the cross-domain attack. Specifically, we train a generator network in the large-scale ImageNet domain to perturb low-level features better, thus generating transferable adversarial examples for the black-box domain. Based on this framework, we further propose two variants to narrow the gap between the source and target domains from the data and model perspectives, respectively. Extensive experiments demonstrate the effectiveness of our proposed methods. We also remind the model owner that "your deployed model is not safe even you do not leak any information to the public". We hope our proposed approaches will serve as a benchmark for evaluating the stability of various deployed models.

References

Shumeet Baluja and Ian Fischer. Adversarial transformation networks: Learning to generate adversarial examples. *CoRR*, abs/1703.09387, 2017.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *ICLR*, 2018.

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *Symposium on Security and Privacy*, 2017.

Rich Caruana, Alexandru Niculescu-Mizil, Geoff Crew, and Alex Ksikes. Ensemble selection from libraries of models. In *ICML*, 2004.

Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *SP*, 2020.

Yue Chen, Yalong Bai, Wei Zhang, and Tao Mei. Destruction and construction learning for fine-grained image recognition. In *CVPR*, 2019.

Adam Coates, Andrew Y. Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In Geoffrey J. Gordon, David B. Dunson, and Miroslav Dudík (eds.), *AISTATS*, 2011.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, 2018.

Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *CVPR*, 2019.

Lianli Gao, Qilong Zhang, Jingkuan Song, Xianglong Liu, and Hengtao Shen. Patch-wise attack for fooling deep neural network. In *ECCV*, 2020.

Lianli Gao, Yaya Cheng, Qilong Zhang, Xing Xu, and Jingkuan Song. Feature space targeted attacks by statistic alignment. In *IJCAI*, 2021.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun (eds.), *ICLR*, 2015.

Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.*, 1990.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *ECCV*, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016b.

Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

Nathan Inkawhich, Wei Wen, Hai (Helen) Li, and Yiran Chen. Feature space perturbations yield more transferable adversarial examples. In *CVPR*, 2019.

Nathan Inkawhich, Kevin J. Liang, Binghui Wang, Matthew Inkawhich, Lawrence Carin, and Yiran Chen. Perturbing across the feature hierarchy to improve standard and strict blackbox attack transferability. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *NeurIPS*, 2020.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Jonathan Krause, Jia Deng, Michael Stark, and Li Fei-Fei. Collecting a large-scale dataset of fine-grained cars. 2013.

Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeurPIS*, 2012.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *ICLR*, 2017a.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *ICLR*, 2017b.

Xiaodan Li, Jinfeng Li, Yuefeng Chen, Shaokai Ye, Yuan He, Shuhui Wang, Hang Su, and Hui Xue. QAIR: practical query-efficient black-box attacks for image retrieval. In *CVPR*, 2021.

Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *ICLR*, 2020.

Yantao Lu, Yunhan Jia, Jianyu Wang, Bai Li, Weiheng Chai, Lawrence Carin, and Senem Velipasalar. Enhancing cross-task black-box transferability of adversarial examples with dispersion reduction. In *CVPR*, 2020.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. volume abs/1306.5151, 2013.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *CVPR*, 2016.

Muzammal Naseer, Salman H. Khan, Muhammad Haris Khan, Fahad Shahbaz Khan, and Fatih Porikli. Cross-domain transferability of adversarial perturbations. In *NeurPIS*, 2019.

Muzammal Naseer, Salman H. Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *CVPR*, 2020.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. 2011.

Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1605.07277, 2016.

Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge J. Belongie. Generative adversarial perturbations. In *CVPR*, 2018.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *IJCV*, 2015.

Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *SIGSAC*, 2016.

Yucheng Shi, Siyu Wang, and Yahong Han. Curls & whey: Boosting black-box adversarial attacks. In *CVPR*, 2019.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun (eds.), *ICLR*, 2015.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun (eds.), *ICLR*, 2014.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, California Institute of Technology, 2011.

Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *CVPR*, 2021.

Xiaosong Wang, Yifan Peng, Le Lu, Zhiyong Lu, Mohammadhadi Bagheri, and Ronald M. Summers. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *CVPR*, 2017.

Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. In *ICLR*, 2020.

Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L. Yuille. Improving transferability of adversarial examples with input diversity. In *CVPR*, 2019.

Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm (eds.), *ECCV*, 2020.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? In Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (eds.), *NeurIPS*, 2014.

A APPENDIX

A.1 DATA AUGMENTATION VS. RANDOM NORMALIZATION

Data augmentation (Krizhevsky et al., 2012; Simonyan & Zisserman, 2015; He et al., 2016a) is a widely used strategy for improving the generalization of the model. However, these label-preserving transformations are less effective for training a generator to craft more transferable adversarial examples.

As demonstrated in Figure 7, where we report the results for our baseline BIA, BIA with data augmentation (BIA+AUG)[6] and BIA with $\mathcal{RN}$ (BIA+$\mathcal{RN}$), BIA+AUG is less effective than our proposed BIA+$\mathcal{RN}$ and might even degrade the performance of our vanilla BIA. For example, when training against VGG-19, BIA gets an average top-1 accuracy of 58.71% on the fine-grained domain, yet BIA+AUG degrades it to 62.29%. In contrast, our proposed BIA+$\mathcal{RN}$ can significantly decrease the result to **47.78%**. This is mainly because that the common data augmentation cannot effectively change the distribution of the training dataset, thus decreasing the generalization towards the black-box domain.

A.2 THE EFFECT OF COMBINING DOMAIN-AGNOSTIC ATTENTION AND RANDOM NORMALIZATION

In this section, we report the results for BIA equipped with both $\mathcal{RN}$ and $\mathcal{DA}$. As illustrated in Figure 8, $\mathcal{RN}$ module and $\mathcal{DA}$ module are not always mutually reinforcing. For example, when training against Res-152 and transferring adversarial examples to fine-grained or source domains, combining $\mathcal{RN}$ module and $\mathcal{DA}$ module can further decrease the average top-1 accuracy to 35.75% and 13.11%, respectively. However, if the black-box domain is coarse-grain, only applying $\mathcal{RN}$ module is better than applying both $\mathcal{RN}$ and $\mathcal{DA}$ modules. We speculate that it may be because the $\mathcal{RN}$ module affects the low-level features extract by the substitute model, and thus be incompatible with the $\mathcal{DA}$ module.

A.3 INSIGHT INTO THE GENERATOR

Although many works (Poursaeed et al., 2018; Naseer et al., 2019) have leveraged the generator to craft adversarial examples, they hardly analyze the role of each block of the generator. This section will give some insight into the generator and understand how it processes an input.

In Figure 9, we feed an input image into our generator $\mathcal{G}_{\theta^*}$ and visualize the outputs of each block in Figure 10, *i.e.*, downsampling block, residual block and upsampling block. Here we also apply the channel-wise average pooling for visualization. As we can observe, these blocks play different roles in generating adversarial examples:

- Downsampling block: it mainly extracts the features of the input image. As the size of the down-sampled images gets smaller, there is no significant noise overall.
- Residual block: Different from downsampling block, it is responsible for adding noise and its behavior looks very similar to the iterative algorithm such as I-FGSM (Kurakin et al., 2017b).
- Upsampling block: This block gradually reconstructs the adversarial example from abstract features.

[6]Specifically, we test three common data augmentation techniques including "RandomResizedCrop", "RandomHorizontalFlip" and "ColorJitter". All implementations are from TorchVision.

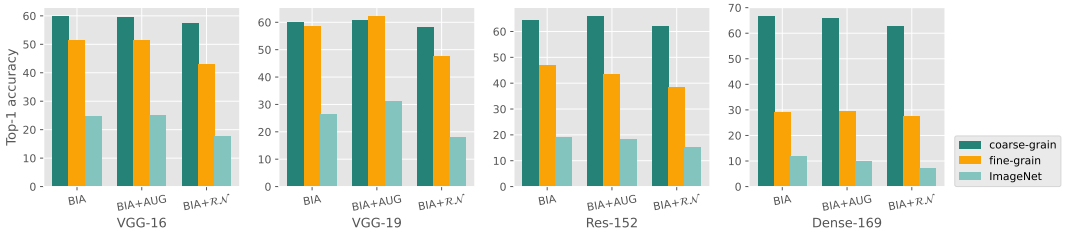

Figure 7: The average top-1 accuracy after attacking (the lower, the better) on coarse-grained, fine-grained and source domains. Here we compare the results of vanilla BIA, BIA with data augmentation (AUG) and BIA with $\mathcal{RN}$. Our generator $\mathcal{G}_\theta$ is trained against ImageNet pre-trained VGG-16, VGG-19, Res-152 and Dense-169, respectively.

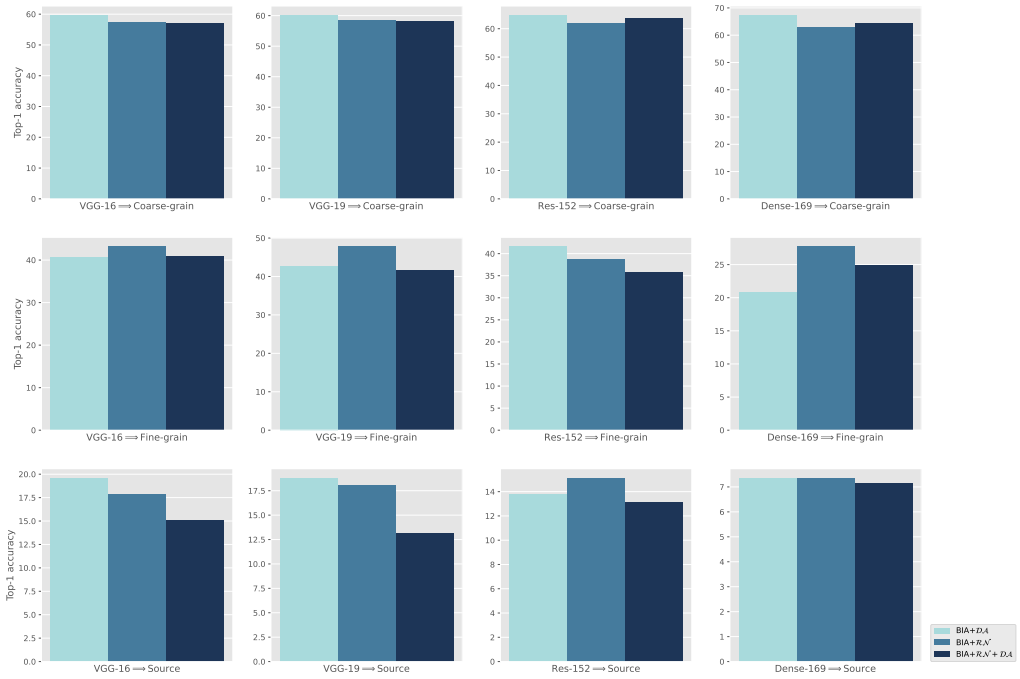

Figure 8: The average top-1 accuracy after attacking (the lower, the better) on coarse-grained, fine-grained and source domains. Here we compare the results of BIA with $\mathcal{DA}$, BIA with $\mathcal{RN}$ and BIA with both $\mathcal{DA}$ and $\mathcal{RN}$. Our generator $\mathcal{G}_\theta$ is trained against ImageNet pre-trained VGG-16, VGG-19, Res-152 and Dense-169, respectively.

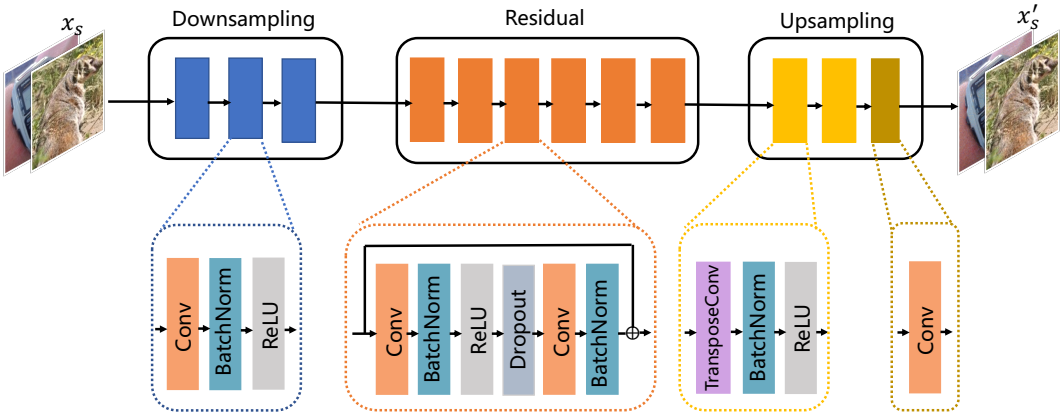

Figure 9: The structure of the generator.

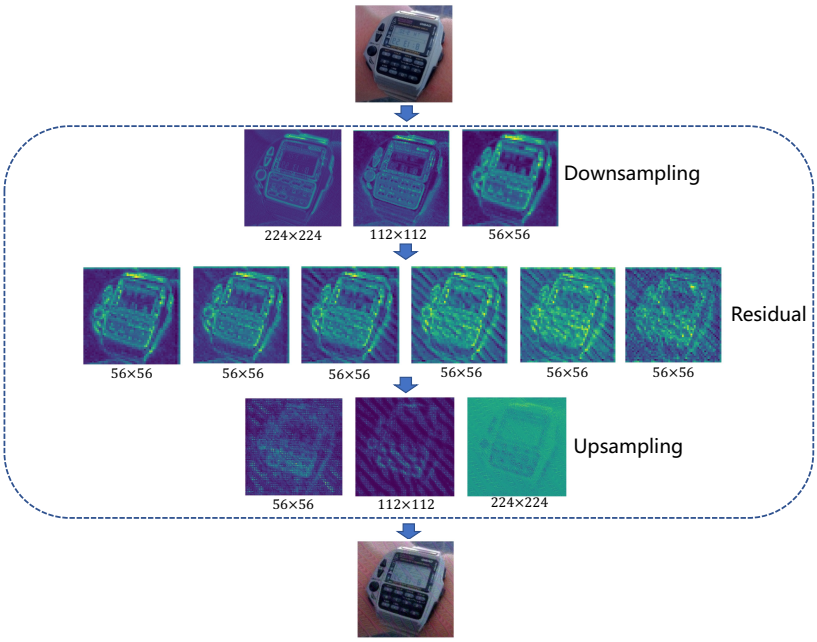

Figure 10: We visualize the outputs of each block of our generator $\mathcal{G}_{\theta^*}$. For each block, we arrange the output from left to right, the closer to the left the closer to the shallow side of the block.