A Two-Stage Data-Free Adversarial Patch Generation Framework

Anonymous authors
Paper under double-blind review

Abstract

General adversarial patch generation (APG) methods rely on training datasets of target models and are not applicable to data-free scenarios. This article presents a two-stage APG framework that exploits a determined proxy dataset in place of an unknown training dataset. The proxy dataset selection stage calculates the proposed average patch saliency (APS) of each available dataset to select a high-APS proxy dataset that can guarantee patches' fooling abilities. Then, the patch generation stage applies the proposed data-free Expectation over Transformation (DF-EoT) as the APG method in case only low-APS datasets are available. Evaluation results show that the determined high-APS proxy datasets enable EoT (benchmark APG method) to generate patches of comparable fooling abilities to patches utilising training datasets, and DF-EoT can further improve the fooling abilities for both low-APS and high-APS proxy datasets. Specifically, DF-EoT enhances average targeted fooling rates (ATFR) of patches utilising a low-APS dataset from 42.71% of EoT to 78.34% on target model VGG-19 and increases ATFR from 62.57% to 84.33% with a high-APS dataset on Inception-v1.

1 Introduction

Convolutional Neural Networks (CNNs) have been widely used for various computer vision tasks, such as object recognition (Simonyan & Zisserman, 2014), object detection (Ren et al., 2017) and semantic segmentation (He et al., 2017). However, recent studies show that CNNs are vulnerable to elaborated adversarial examples (Szegedy et al., 2013; Goodfellow et al., 2014; Kurakin et al., 2016; Carlini & Wagner, 2017; Madry et al., 2017; Serban et al., 2020). Given an input image and an image classifier as a target model, adversarial attacks generate an imperceptible adversarial perturbation, leading to misclassification of the perturbed image called an adversarial example. Such perturbation is also called an image-specific perturbation since its effectiveness on other input images can not be guaranteed.

Image-agnostic perturbations target scenarios where input images are agnostic, and the perturbations' fooling abilities are measured by fooling rates. Image-agnostic attacks generate a single perturbation to fool multiple input images due to attackers' limited knowledge of a target model's processing image at a specific moment (Moosavi-Dezfooli et al., 2017). In practice, the input images and the training dataset of a target model are usually seen as samples from the same training distribution. Thus, image-agnostic perturbations' fooling abilities are often measured by targeted fooling rates (TFR) or untargeted fooling rates (UFR) on the training dataset samples. Specifically, When attacking classifiers, TFR is the ratio of the number of training samples fooled a the target class to the number of all samples, and UFR is the ratio of the number of misclassified samples to the total number of samples. Generally, for attackers, achieving targeted classification results are more challenging than untargeted classification results (Serban et al., 2020).

Image-agnostic attacks update perturbations' pixels values over training dataset samples to guarantee the perturbations' fooling abilities (Moosavi-Dezfooli et al., 2017; Brown et al., 2017; Sharif et al., 2016; Eykholt et al., 2018; Thys et al., 2019; Hoory et al., 2020; Kaziakhmedov et al., 2019; Salman et al., 2020; Duan et al., 2020). During an image-agnostic attack, a perturbation is first added to an image sampled from the target classifier's training dataset. Then, the perturbation's pixel value is updated according to a designed objective function to make the model misclassify the current perturbed image. Finally, the above two steps repeat for the same perturbation over multiple training dataset samples to make the perturbation fool most input images. However, the vital role of training datasets makes such data-dependent attacks unsuitable when training data is unknown, especially regarding concerns about proprietary data and privacy issues.

This work selects adversarial patch (Brown et al., 2017) as image-agnostic perturbations' form and refers to the previous research (Chaubey et al., 2020) to call situations where training data is unknown as *data-free scenarios*. To make existed data-dependent APG methods applicable to data-free scenarios, a natural practice is to replace the training dataset of the methods with a proxy dataset to generate patches (See Appendix A for more related works). However, such a practice still faces the following challenges. 1) *PDS Metric Determination*: APG methods' performance varies with proxy datasets. Thus a metric for PDS is needed. 2) *Fooling Ability Metric Establishment*: Gradient-based APG methods are sensitive to initial conditions, suggesting patches' fooling abilities vary with choices of patches' initial values (Madry et al., 2017). Nonetheless, when training data is accessible, attackers can select the patch with the largest fooling rate as the final output from patches with different initial values (i.e., restart selection). Thus, a fooling ability metric in data-free scenarios must be established to alleviate such sensitivity. 3) *Fooling Ability Improvement for Available Datasets*: Even if a PDS metric is provided, attackers can not guarantee that the available datasets at hand happen to contain a dataset that can be used to achieve satisfactory fooling ability of patches. Therefore, attackers need a new APG method to guarantee the patches' fooling abilities on a wide range of proxy datasets.

To address the associated challenges, the proposed two-stage framework first uses a PDS metric to determine a proxy dataset and then applies a APG method to generate patches. Specifically, the proxy dataset selection (PDS) stage calculates the proposed average patch saliency (APS) of each attackers' available dataset in order to selects a high-APS dataset as the proxy dataset. Then, the patch generation (PG) stage applies the designed data-free Expectation over Transformation (DF-EoT) to generate patches based on the determined dataset. Note that the proposed framework is tested in the digital setting in order to focus on the problem of obtaining patches' fooling abilities in the absence of training data and eliminate the influence of environmental factors.

Our main contribution is summarized as follows. 1)A two-stage data-free APG framework is proposed. 2) The proposed APS enables attackers to select a proxy dataset on which data-dependent EoT (benchmark APG method) can achieve considerable performance. 3) The proposed patch saliency (PS) can alleviate the sensitivity of patch generation to initial conditions. 4) DF-EoT enhance patches' fooling abilities than EoT for both high-APS and low-APS datasets. For readability, a list of abbreviations is provided in Appendix B.

2 Methodology

Section 2.1 first formalises the APG problem. Then, Section 2.2 introduces how EoT solves the APG problem and explains why EoT is taken as the benchmark in this work. Last, Section 2.3 presents the proposed two-stage data-free APG framework, where Section 2.3.1 introduces the APS and PS calculation process, and Section 2.3.2 presents the proposed APG method, DF-EoT.

2.1 APG Problem Formalisation

This work selects image classifier as the target model. Given an input image $\boldsymbol{X} \in \mathbb{R}^d$ [1] and its ground-truth class label $y \in \{1, ..., m\}$ sampled from a training dataset $\boldsymbol{\mathcal{D}}_{train}$, a trained m-class image classifier $\mathcal{F} : \mathbb{R}^d \to \mathbb{R}^m$ estimates the class of the input image by

$$\hat{y} = arg \max_{j \in \{1,2,..,m\}} [\mathcal{F}(\boldsymbol{X})]_j, \tag{1}$$

where $[\cdot]_j$ denotes the j-th component of an inner vector and the final layer of $\mathcal{F}$ is a softmax layer.

An adversarial patch attacks by replacing an input image's local pixels with itself. Specifically, given any image $\boldsymbol{X}$ sampled from $\boldsymbol{\mathcal{D}}_{train}$, a targeted adversarial patch $\boldsymbol{P}^* \in \mathbb{R}^n$ is expected to mislead the classification result of the perturbed image $\boldsymbol{X}_{\boldsymbol{P}^*}$ into a target class y'. The generation of $\boldsymbol{P}^*$ is

[1] The input image is defined $\boldsymbol{X} \in \mathbb{R}^d$ instead of $\boldsymbol{X} \in \mathbb{R}^{C \times W \times H}$ for the following mathematical expression simplicity.

formalised as solving the following optimisation problem in (Brown et al., 2017):

$$\boldsymbol{P}^* = arg\min_{\boldsymbol{P}} \mathbb{E}_{\boldsymbol{X}\sim\boldsymbol{\mathcal{D}}_{train}, L\sim\boldsymbol{\mathcal{L}}, T\sim\boldsymbol{\mathcal{T}}} \mathcal{O}(\mathcal{F}(\boldsymbol{X}_{\boldsymbol{P}}), y'), \tag{2}$$

where $\boldsymbol{\mathcal{L}}$ is a collection of potential locations of $\boldsymbol{P}^*$ in $\boldsymbol{X}$, $\boldsymbol{\mathcal{T}}$ denotes a collection of potentially existed natural transformations on $\boldsymbol{P}^*$. The objective function $\mathcal{O} : \mathbb{R}^m \times \{1, ..., m\} \rightarrow \mathbb{R}$ is

$$\mathcal{O}(\mathcal{F}(\boldsymbol{X}_{\boldsymbol{P}}), y') = -\log([\mathcal{F}(\boldsymbol{X}_{\boldsymbol{P}})]_{y'}) \tag{3}$$

which is the negative log of the probability that $\mathcal{F}$ predicts the input $\boldsymbol{X}_{\boldsymbol{P}}$ being the class y'. The perturbed image $\boldsymbol{X}_{\boldsymbol{P}}$ is defined as

$$\boldsymbol{X}_{\boldsymbol{P}} = \mathcal{A}(\boldsymbol{P}, T, L, \boldsymbol{X}), \tag{4}$$

where the function $\mathcal{A}$ denotes an *attachment operation* that first transforms the patch $\boldsymbol{P}$ with sampled transformation T and then attaches the transformed patch $\boldsymbol{P}'$ onto image $\boldsymbol{X}$ at location L (see Figure 1).

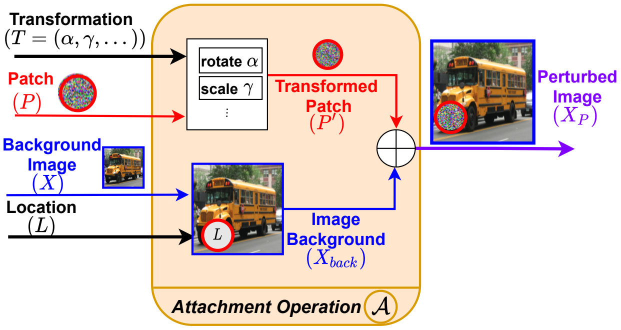

Figure 1: Attachment operation. The adopted ranges for sampled transformations in (Athalye et al., 2018) are $\alpha \in (-22.5°, 22.5°)$ and $\gamma \in (0.9, 1.4)$, and α, γ are sampled uniformly at random from their ranges. Note the difference between *background image* $\boldsymbol{X}$ and *image background* $\boldsymbol{X}_{back}$. This work refers $\boldsymbol{X}_{back}$ to the remaining pixels in $\boldsymbol{X}$ except for the pixels of patch $\boldsymbol{P}$.

2.2 The Benchmark APG Method: EoT

(Brown et al., 2017) apply the EoT (Athalye et al., 2018) to solve Equation (2). As shown in Figure 2, EoT iteratively updates the pixel values of $\boldsymbol{P}$ from randomly initialised values (with uniform random noise). Specifically, in each iteration, EoT first performs the attachment operation $\mathcal{A}$ with the patch $\boldsymbol{P}$, the sampled transformation T, location L and background image $\boldsymbol{X}$ to get a perturbed image $\boldsymbol{X}_{\boldsymbol{P}}$. After that, the patch's pixel values in $\boldsymbol{X}_{\boldsymbol{P}}$ take a one-time update with Projected Gradient Descent (PGD) (Madry et al., 2017) to minimise Equation (3). Finally, EoT stops after reaching certain iteration times.

EoT is taken as the benchmark APG method in this work. First, EoT is representative and reliable given the fact that it can even be used to synthesize 3-D adversarial objects. Second, EoT mainly differs from the other APG methods (Sharif et al., 2016; Kaziakhmedov et al., 2019; Hoory et al., 2020; Duan et al., 2020) in the practices coping with the natural transformations. Nonetheless, the approaches of iteratively updating the patch while placing it on training dataset samples to achieve fooling ability are the same for these APG methods. Therefore, this work disregards natural transformations of EoT to eliminate the gaps between different methods and focuses on achieving patches' fooling abilities in data-free scenarios.

2.3 Two-Stage Data-Free APG Framework

Since EoT requires access to training data, a data-free APG method is desired in scenarios where data privacy matters. Specifically, given attackers' available datasets $\{\boldsymbol{\mathcal{D}}_{\mathbf{1}}, \boldsymbol{\mathcal{D}}_{\mathbf{2}}, ...\}$, the proposed two-stage data-free APG frameworks formalises the patch generation problem as the following equation:

$$\boldsymbol{P}^* = arg\min_{\boldsymbol{P}} \mathbb{E}_{\boldsymbol{X}\sim\boldsymbol{\mathcal{D}}_{proxy}, L\sim\boldsymbol{\mathcal{L}}, T\sim\boldsymbol{\mathcal{T}}} \mathcal{O}(\mathcal{F}(\boldsymbol{X}_{\boldsymbol{P}}), y'), \tag{5}$$

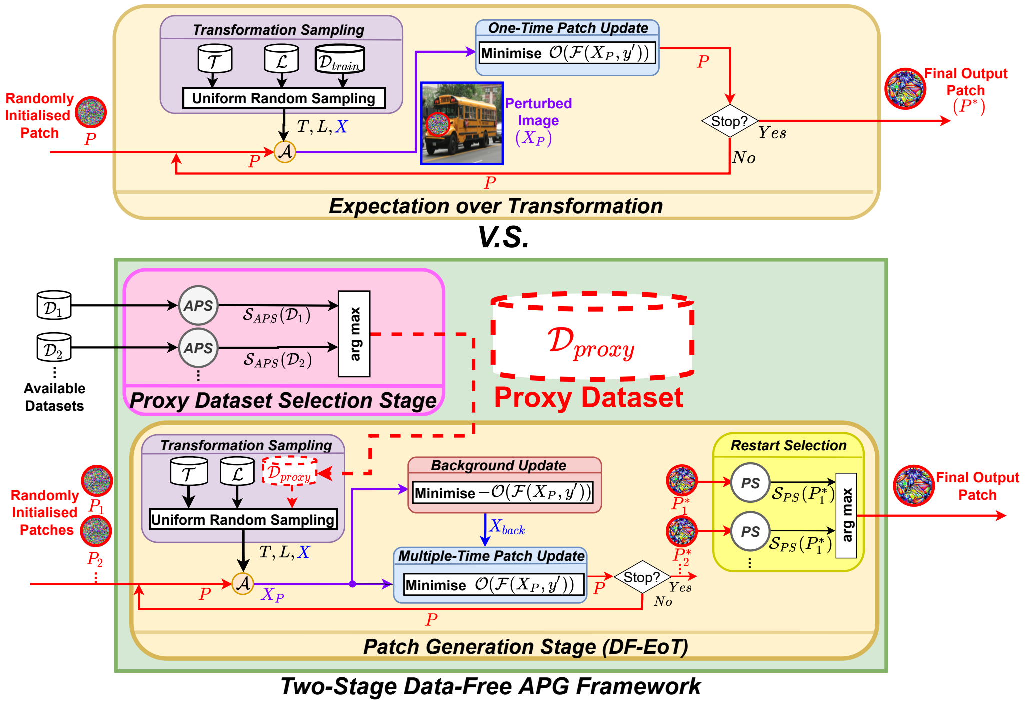
Transformation Sampling
$\mathcal{T}$
$\mathcal{L}$
$\mathcal{D}_{train}$
Uniform Random Sampling
T, L, X
One-Time Patch Update
Minimise $\mathcal{O}(\mathcal{F}(X_P, y'))$
Randomly Initialised Patch
P
$\mathcal{A}$
Perturbed Image (X_P)
Stop?
Yes
No
Final Output Patch (P^*)
Expectation over Transformation
V.S.
$\mathcal{D}_1$
$\mathcal{D}_2$
Available Datasets
APS
$\mathcal{S}_{APS}(\mathcal{D}_1)$
$\mathcal{S}_{APS}(\mathcal{D}_2)$
arg max
Proxy Dataset Selection Stage
$\mathcal{D}_{proxy}$
Proxy Dataset
Transformation Sampling
$\mathcal{D}_{proxy}$
Uniform Random Sampling
Randomly Initialised Patches
P_1
P_2
T, L, X
X_P
Background Update
Minimise $-\mathcal{O}(\mathcal{F}(X_P, y'))$
X_{back}
Multiple-Time Patch Update
Minimise $\mathcal{O}(\mathcal{F}(X_P, y'))$
P_1^*
P_2^*
Restart Selection
PS
$\mathcal{S}_{PS}(P_1^*)$
arg max
Final Output Patch
Patch Generation Stage (DF-EoT)
Two-Stage Data-Free APG Framework

Figure 2: EoT v.s. the two-stage data-free APG framework. All the patches are initialised with uniform random noise and are of the same target class y'. DF-EoT differs from EoT in the following step: i) background update, ii) multiple-time patch update and iii) restart selection. It first takes a set of initialised patches $\{\boldsymbol{P}_1, \boldsymbol{P}_2, ...\}$ as input one by one, then generates adversarial patches $\{\boldsymbol{P}_1^*, \boldsymbol{P}_2^*, ...\}$ of the same target class, and last selects the final output patch through the restart selection step. Note that, EoT can be used as the APG method in the PG stage, while this work applies the proposed DF-EoT.

where

$$\boldsymbol{\mathcal{D}}_{proxy} = arg \max_{\boldsymbol{\mathcal{D}} \in \{\boldsymbol{\mathcal{D}}_1, \boldsymbol{\mathcal{D}}_2, ...\}} \mathcal{S}_{PDS}(\boldsymbol{\mathcal{D}}, \mathcal{F}) \tag{6}$$

and $\mathcal{S}_{PDS}$ is a PDS metric[2] reflecting APG methods' performance on $\boldsymbol{\mathcal{D}}$. As shown in Figure 2, the two-stage framework first selects $\boldsymbol{\mathcal{D}}_{proxy}$ using the proposed APS to solve Equation (6), then generates adversarial patches with the designed DF-EoT to solve Equation (5).

2.3.1 Stage I: Proxy Dataset Selection

APS is the proposed PDS metric and its calculation process is shown in Figure 3. Specifically, given an available dataset $\boldsymbol{\mathcal{D}}$, APS first randomly selects k different target classes $\{y_1', y_2', ..., y_k'\}$ and initialises k patches $\{\boldsymbol{P}_1, \boldsymbol{P}_2, ..., \boldsymbol{P}_k\}$ with uniform random noise. Second, APS takes $\boldsymbol{\mathcal{D}}$ as the proxy dataset and uses EoT to generate patches of the selected target classes, i.e., $\{\boldsymbol{P}_1^*, \boldsymbol{P}_2^*, ..., \boldsymbol{P}_k^*\}$. Finally, the APS of the dataset $\boldsymbol{\mathcal{D}}$, $\mathcal{S}_{APS}(\boldsymbol{\mathcal{D}}) \in \mathbb{R}$, is computed as the following equation:

$$\mathcal{S}_{APS}(\boldsymbol{\mathcal{D}}) = \sum_{c=1}^{k} \frac{1}{k} \mathcal{S}_{PS}(\boldsymbol{P}_c^*), \tag{7}$$

where $\mathcal{S}_{PS}$ is the proposed patch fooling ability metric in data-free scenarios, patch saliency (PS). APS measures the overall fooling abilities of a group of patches generated based on the dataset $\boldsymbol{\mathcal{D}}$. By making the target classes $\{y_1', y_2', ..., y_k'\}$ in Equation (7) cover all learned classes of $\mathcal{F}$ (i.e., $k = m$), APS can thus serve as a PDS metric.

[2] The symbol $\mathcal{F}$ is omitted in the following part for simplicity.

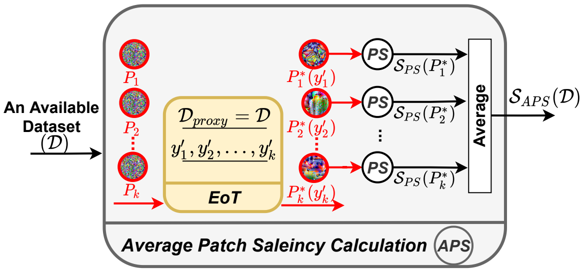
An Available
Dataset
$(\mathcal{D})$
P_1
P_2
P_k
$\mathcal{D}_{proxy} = \mathcal{D}$
$y'_1, y'_2, \ldots, y'_k$
EoT
$P^*_1(y'_1)$
$P^*_2(y'_2)$
$P^*_k(y'_k)$
PS
PS
PS
$\mathcal{S}_{PS}(P^*_1)$
$\mathcal{S}_{PS}(P^*_2)$
$\mathcal{S}_{PS}(P^*_k)$
Average
$\mathcal{S}_{APS}(\mathcal{D})$
Average Patch Saleincy Calculation
APS

Figure 3: APS calculation process. The EoT takes initialised patches $\{\boldsymbol{P}_1, \boldsymbol{P}_2, ..., \boldsymbol{P}_k\}$ as input one by one, and generates adversarial patches $\{\boldsymbol{P}^*_1, \boldsymbol{P}^*_2, ..., \boldsymbol{P}^*_k\}$ of k different target classes. The notation $\boldsymbol{P}^*_1(y'_1)$ denotes the patch $\boldsymbol{P}^*_1$ of target class y'_1.

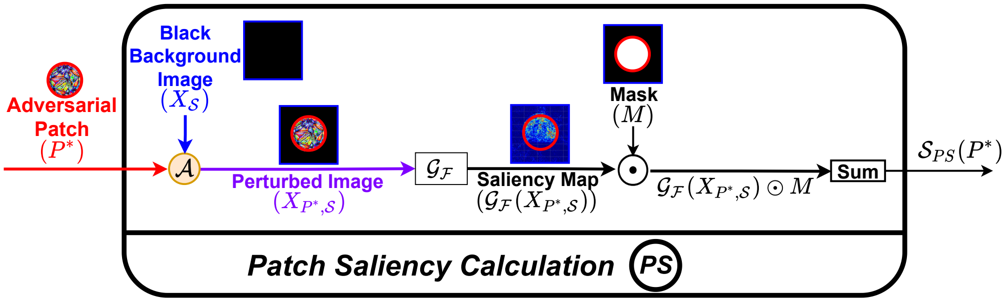

Figure 4: PS calculation process. Omitting T and L in the attachment operation $\mathcal{A}$ refers to that the patch $\boldsymbol{P}^*$ is put on the centre of the background image, and none of transformations is applied on $\boldsymbol{P}^*$.

PS is the proposed fooling ability metric in data-free scenarios and its calculation process is shown in Figure 4. Specifically, given an adversarial patch $\boldsymbol{P}^*$ of target class y', PS first attaches $\boldsymbol{P}^*$ onto the centre of a background image $\boldsymbol{X}_\mathcal{S} \in \mathbb{R}^d$ to get a perturbed image $\boldsymbol{X}_{\boldsymbol{P}^*,\mathcal{S}}$. Then, the Integrated Gradient function $\mathcal{G}_\mathcal{F} : \mathbb{R}^d \to \mathbb{R}^d$ (see Appendix C) takes $\boldsymbol{X}_{\boldsymbol{P}^*,\mathcal{S}}$ as input and output the saliency map $\mathcal{G}_\mathcal{F}(\boldsymbol{X}_{\boldsymbol{P}^*,\mathcal{S}})$, where $\mathcal{G}_\mathcal{F}(\boldsymbol{X}_{\boldsymbol{P}^*,\mathcal{S}})$ measures the importance of each component in the input $\boldsymbol{X}_{\boldsymbol{P}^*,\mathcal{S}}$ for a predicted class label y' of the target model $\mathcal{F}$. The PS of the patch $\boldsymbol{P}^*$, $\mathcal{S}_{PS}(\boldsymbol{P}^*) \in \mathbb{R}$, is defined as:

$$\mathcal{S}_{PS}(\boldsymbol{P}^*) = \sum_{b=1}^{d} [\boldsymbol{S}_{\boldsymbol{P}^*,\boldsymbol{X}_s}]_b, \tag{8}$$

where

$$\boldsymbol{S}_{\boldsymbol{P}^*,\boldsymbol{X}_s} = \boldsymbol{M} \odot \mathcal{G}_\mathcal{F}(\mathcal{A}(\boldsymbol{P^*}, \boldsymbol{X}_\mathcal{S})), \tag{9}$$

$[\cdot]_b$ denotes the b-th component of an inner vector, $\boldsymbol{M} \in \mathbb{R}^d$ is the mask of the $\boldsymbol{P}^*$ on $\boldsymbol{X}_\mathcal{S}$ and $\odot$ refers to element-wise product. PS actually measures the fooling ability of $\boldsymbol{P}^*$ when the patch is placed on the background image $\boldsymbol{X}_\mathcal{S}$. By placing each same-class patches on the same background image $\boldsymbol{X}_\mathcal{S}$, PS can measure the fooling abilities of patches generated with different proxy datasets on the same contextual condition. This work selects a *black image* as the background image through an empirical study presented in Appendix D.

2.3.2 Stage II: Patch Generation

Even a selected high-APS proxy dataset enables EoT to achieve considerable performance, in practice, attackers can not guarantee the available datasets at hand happen to include a high-APS dataset due to their limited data resource. The proposed DF-EoT pursues the performance improvement of EoT on a wide range of proxy datasets in data-free scenarios. As shown in Figure 2, compared with EoT, DF-EoT considers the update of image backgrounds and patch initial value sensitivity. Specifically, DF-EoT differs from EoT mainly in the following three steps.

i) *Background Update*. One of the most significant differences between DF-EOT and EoT is introducing the background update step. Specifically, after the *background image* $\boldsymbol{X}$ sampled from $\boldsymbol{\mathcal{D}}_{proxy}$, the *image background* $\boldsymbol{X}_{back}$ is first updated with PGD to solve the following equation:

$$\boldsymbol{X}_{back} = arg \min_{\boldsymbol{X}_{back}} -\mathcal{O}(\mathcal{F}(\boldsymbol{X}_{\boldsymbol{P}}), y'), \tag{10}$$

where $\mathcal{O}$ is defined in Equation (3), the number of iteration of PGD is β_{back} and the update step size is η_{back}. Equation (10) actually indicates updating the pixel values of $\boldsymbol{X}_{back}$ to suppress $\boldsymbol{X}_{\boldsymbol{P}}$ from being classified as the target class y'.

ii) *Multiple-Time Patch Update*. After the background update step, the multiple-time patch update step updates the patch $\boldsymbol{P}$ with PGD to minimise the same objective function of Equation (3) as EoT. DF-EoT updates the patch β_{patch} times for each sampled image $\boldsymbol{X}$ with the step size η_{patch}, while EoT takes one-time update (i.e., $\beta_{patch} = 1$).

iii) *Restart Selection*. DF-EoT iterates the above two steps β_{total} times to get an adversarial patch $\boldsymbol{P}^*$. Since gradient-based attacks are sensitive to the initial values of the patch $\boldsymbol{P}$ (Madry et al., 2017), the restarts selection step is further introduced to alleviate the PGD's sensitivity to initial conditions. Specifically, DF-EoT first generates adversarial patches $\{\boldsymbol{P}_1^*, \boldsymbol{P}_2^*, ...\}$ of the same target class y' from different randomly initialised patches $\{\boldsymbol{P}_1, \boldsymbol{P}_2, ...\}$. Then the PS of each patch is computed. Finally, the patch with the greatest PS is taken as the final output patch.

3 Experimentation and Discussion

This section evaluates the proposed APG framework. Section 3.1 presents the experimental setting. Section 3.2 introduces average targeted fooling rate (ATFR) and targeted fooling rate (TFR) as the evaluation metrics. Section 3.3 evaluates APS and PS by building their consistency with ATFR and TFR. Section 3.4 assesses the proposed DF-EoT by using ATFR to compare the generated patches' fooling abilities of DF-EoT and EoT.

3.1 Experimental Setting

For the sake of generality, the proposed framework is evaluated on different model architectures widely deployed in industry, including VGG-16, VGG-19 (Simonyan & Zisserman, 2014), ResNet-18, ResNet-34 (He et al., 2016), and Inception-v1 (Szegedy et al., 2015) trained on ImageNet training dataset (Krizhevsky et al., 2017). Note that this work randomly selects 100 (out of 1000) learned classes as all the learned classes of the target classifiers in order to reduce computation cost (i.e., m=100).

To evaluate APS and PS, this work takes *five* datasets as attackers' available datasets. The three large-scale datasets are 1) IMAGENET: 40k images of ImageNet's validation set, 2) MSCOCO: training set of MSCOCO dataset (Lin et al., 2014), 3) KITTI: training set of KITTI dataset (Geiger et al., 2013). Considering some attackers' limited data resources, two datasets that can be manually synthesised are also involved, i.e., 4) UNIFORM: 3000 uniform-random-noise images and 5) WHITE: 3000 white images. Taking the five datasets as the proxy datasets of EoT finally results in different degrees of patches' fooling abilities, through which this paper hopes to simulate a wide range of attackers' possibly available datasets. To evaluate the determined black background image $\boldsymbol{X}_{\mathcal{S}}$ in Equation (8), *two* datasets mainly composed of black images are synthesised, where the two datasets are 1) BLACK: 3000 black images and 2) ENSEMBLE: 3000 images that evenly comes from UNIFORM, WHITE and BLACK datasets (see Appendix D). In order to evaluate DF-EoT on a wide range of proxy datasets, all the mentioned *seven* datasets are used as the proxy dataset of DF-EoT and EoT, respectively, to compare the fooling abilities of the generated patches.

The patches generated in this work are circles with a radius of 25, which accounts for 3.91% of the input image pixel space of shape 224 × 224. For the patch generation process, only the location sampling and background images sampling are considered in EoT and DF-EoT. For the patch test process, 10k images of ImageNet's validation set are used to compute TFR and ATFR. For EoT, the total iteration times are 2000 and the patch update step size is $1/255$. For DF-EoT, we select

$\beta_{total} = 400$, $\beta_{back} = 2$, $\eta_{back} = 1/255$, $\beta_{patch} = 10$, $\eta_{patch} = 1/255$ and the number of randomly initialised patches to be 10. Note that the average time of generating a patch using DF-EoT on GPU RTX 2080 is approximately 90 seconds. In the same condition, EoT takes about 38 seconds. Nonetheless, the computational cost is acceptable since adversarial patches are image-agnostic.

3.2 Evaluation Metrics: ATFR and TFR

TFR is a patch fooling ability metric when training data is available. Given a patch $\boldsymbol{P}^*$ of target class y' and images $\{\boldsymbol{X}_h\}_{h=1}^e$ of classes $\{y_h\}_{h=1}^e$ sampled from $\boldsymbol{\mathcal{D}}_{train}$, TFR measures the fooling ability of $\boldsymbol{P}^*$, and is defined as the ratio of samples fooled into target class y' to all samples, i.e.,

$$\mathcal{S}_{TFR}(\boldsymbol{P}^*) = \frac{1}{e}\sum_{h=1}^{e} 1_{\hat{y}_{h,\boldsymbol{P}^*}=y'}, \tag{11}$$

where $\hat{y}_{h,\boldsymbol{P}^*}$ denotes the target model's estimated class of the perturbed image $\boldsymbol{X}_{\boldsymbol{P}^*,h}$ ($\boldsymbol{X}_h$ attached with $\boldsymbol{P}^*$) and $\{\boldsymbol{X}_h\}_{h=1}^e$, $\{y_h\}_{h=1}^e$ and y' are omitted in $\mathcal{S}_{TFR}(\boldsymbol{P}^*)$ for simplicity. Similarly, untargeted fooling rate (UFR) is defined as the ratio of samples fooled into the class different from the class y_h to all samples.

ATFR can be seen as PDS metric when training data can be accessed. Given adversarial patches $\{\boldsymbol{P}_c^*\}_{c=1}^k$ of different target classes $\{y_c'\}_{c=1}^k$ and the corresponding proxy dataset $\boldsymbol{\mathcal{D}}$, the ATFR is the average TFR of $\{\boldsymbol{P}_c^*\}_{c=1}^k$ and is defined as:

$$\mathcal{S}_{ATFR}(\boldsymbol{\mathcal{D}}) = \frac{1}{k}\sum_{c=1}^{k} \mathcal{S}_{TFR}(\boldsymbol{P}_c^*). \tag{12}$$

By making $\{y_c'\}_{c=1}^k$ covers all the learned classes of the target model, ATFR measures the overall fooling abilities of patches generated with benchmark EoT and based on $\boldsymbol{\mathcal{D}}$, thus can be treated as the PDS metric.

3.3 Evaluation of the Proxy Dataset Selection Stage

3.3.1 Consistency between APS and ATFR

Model	**Metric**	**Available Dataset**				
		IMAGENET	MSCOCO	KITTI	UNIFORM	WHITE
VGG16	ATFR	95.65%	95.26%	93.24%	57.42%	79.59%
	APS	**27.91**	26.51	26.07	12.15	19.66
VGG19	ATFR	91.26%	93.55%	92.58%	42.71%	74.90%
	APS	**28.12**	26.89	26.87	9.21	18.04
ResNet18	ATFR	77.30%	76.33%	73.76%	27.67%	44.17%
	APS	**15.21**	14.44	14.97	6.83	11.03
ResNet34	ATFR	71.51%	78.20%	73.31%	20.67%	32.15%
	APS	**15.60**	15.19	15.87	5.06	8.72
Inception-v1	ATFR	60.71%	62.57%	60.39%	9.43%	24.35%
	APS	10.06	9.68	**10.32**	2.25	5.25

Table 1: APSs and ATFRs of available datasets.

Since ATFR is a PDS metric computed using training data, building the consistency between ATFR and APS is desired, i.e., the dataset of greater ATFR is expected to have greater APS. As shown in Table 1, the high-APS datasets such as IMAGENET, MSCOCO and KITTI have much greater ATFRs than the low-APS datasets such as UNIFORM and WHITE. Especially, APS assigns the training dataset IMAGENET with the greatest value for all the considered target models except for Inception-v1. Nonetheless, the selected proxy dataset for Inception-v1 achieves comparable ATFR with IMAGENET. Overall, Table 1 enlightens us to apply APS as a PDS metric. Besides that, the achieved ATFR on high-APS proxy datasets suggests EoT can be directly applied to data-free scenarios.

Actually, APS guarantees the generated patches' fooling abilities by access to the target model in its calculation process. In other words, APS ranks the available datasets by comparing the saliency of the model's extracted information. On the one hand, since the target model is trained on the training dataset, it can memorise the features of its seen training data. The adversarial patch generation process can be seen as extracting such memorised information and mimicking these features, which explains why EoT using uniform random noise can achieve a certain degree of patch fooling ability in Table 1. On the other hand, the saliency of the extracted features varies with the selected proxy dataset. The closer the features in proxy datasets are to the memorized features, the greater the PS of extracted features, which explains why the original training dataset is assigned the highest APS value and uniform random noise is assigned the lowest value in Table 1. Therefore, by comparing the average PS of generated patches on each dataset, APS can serve as a proxy dataset selection metric.

3.3.2 Consistency between PS and TFR

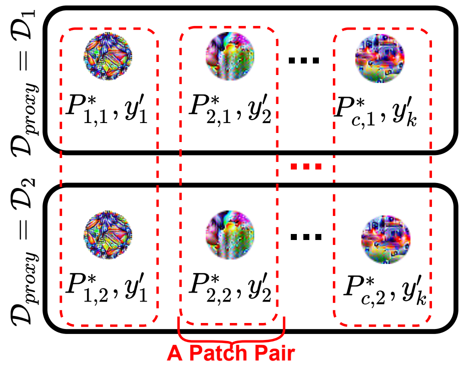

Figure 5: Patch pairs for consistency rate test. The patches are generated with EoT.

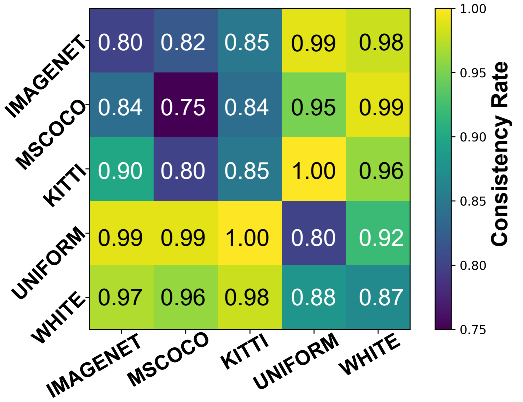

Figure 6: Consistency rates for any two proxy datasets.

This work further proposes *consistency rate test* to build the consistency between PS and TFR, which can be seen as establishing a case-level consistency between APS and ATFR as Equation (7) and Equation (12) suggest. The experimental steps of the consistency rate test are as follows. As shown in Figure 5, EoT first generates two sets of patches based on proxy datasets $\mathcal{D}_1$ and $\mathcal{D}_2$ respectively. Then, two patches of the same target class are formed a *patch pair*, where the patch pair is *consistent* if the patch with a larger TFR has a greater PS than the other patch. Finally, the consistency rate is computed as the ratio of the number of consistent pairs to all the formed pairs. Ideally, the consistency rate for any two datasets is expected to be 1, which indicates PS can totally replace TFR to serve as a general fooling ability metric.

Figure 6 presents the consistency rates for any two available datasets, where the mean consistency rate reaches 0.91. Specifically, the average consistency rate is 0.82 for two high-APS datasets and 0.87 for two low-APS datasets, while 0.98 for a high-APS and a low-APS dataset. The mean of diagonal values of Figure 6 reaches 0.80, where a diagonal values is the consistency rate of two sets of patches generated based on the same proxy dataset.

The high consistency rates further prove that APS can serve as a reliable PDS metric for data-free scenarios, especially for datasets with vastly different ATFRs. The diagonal values suggest PS can be a metric for comparing patches' fooling abilities generated based on the same proxy dataset, which enlightens us to introduce PS as a restart selection metric of DF-EoT.

Model	Metric	Method	Proxy Dataset						
			IMAGENET	MSCOCO	KITTI	UNIFORM	WHITE	BLACK	ENSEMBLE
VGG-16	ATFR	EoT	95.65%	95.26%	93.24%	57.42%	79.59%	78.74%	88.97%
		DF-EoT	96.47%	96.72%	97.08%	83.99%	86.99%	87.74%	95.35%
	AUFR	EoT	99.23%	99.10%	98.73%	93.73%	95.58%	95.81%	97.46%
		DF-EoT	99.68%	99.68%	99.61%	97.55%	97.42%	97.42%	98.93%
VGG-19	ATFR	EoT	91.26%	93.55%	92.58%	42.71%	74.90%	72.21%	86.44%
		DF-EoT	92.26%	93.30%	95.99%	78.34%	85.00%	83.45%	94.09%
	AUFR	EoT	98.74%	98.97%	98.83%	91.73%	95.30%	94.56%	96.75%
		DF-EoT	99.52%	99.53%	99.50%	97.28%	97.12%	97.05%	98.66%
ResNet-18	ATFR	EoT	77.30%	78.20%	73.76%	27.67%	44.17%	46.93%	58.15%
		DF-EoT	76.87%	79.09%	79.56%	49.06%	52.21%	53.15%	69.90%
	AUFR	EoT	94.41%	93.13%	92.83%	82.41%	85.12%	85.56%	88.08%
		DF-EoT	96.20%	96.30%	95.25%	88.02%	87.08%	87.15%	91.18%
ResNet-34	ATFR	EoT	71.51%	78.20%	73.31%	20.67%	32.15%	38.67%	52.01%
		DF-EoT	72.64%	74.33%	78.02%	40.90%	39.61%	45.58%	67.06%
	AUFR	EoT	92.17%	93.13%	90.59%	76.35%	76.89%	78.73%	82.54%
		DF-EoT	95.37%	95.34%	94.05%	82.14%	79.59%	81.17%	87.17%
Inception-v1	ATFR	EoT	60.71%	62.57%	60.39%	9.43%	24.35%	24.90%	38.40%
		DF-EoT	84.83%	84.33%	80.71%	26.28%	32.44%	34.61%	58.00%
	AUFR	EoT	84.83%	84.33%	80.71%	59.54%	64.49%	64.95%	69.70%
		DF-EoT	89.92%	89.16%	86.41%	68.02%	68.62%	69.62%	78.02%

Table 2: ATFRs and AUFRs of EoT and DF-EoT.

3.4 Evaluation of Patch Generation Stage

3.4.1 Improved Patch Fooling Ability via DF-EoT

This work evaluates DF-EoT and EoT with ATFR and AUFR metrics. Note that the metrics reflect the APG methods' performance when the proxy dataset is controlled. As shown in Table 2, DF-EoT increases patches' fooling abilities on a wide range of proxy datasets than EoT. For instance, DF-EoT improves ATFR to 78.34% by 35.63% for a low-APS dataset UNIFORM on the target model VGG-19 and increases ATFR to 84.33% by 21.76% for a high-APS dataset MSCOCO on the target model Inception-v1. Evaluation results of the generated patches' transferability are present in Appendix E, and ablation studies of DF-EoT's three main steps are provided in Appendix F. A performance comparison between EoT using the training dataset and the two-stage data-free APG framework is provided in Appendix G.

4 Conclusion

This work proposes a two-stage data-free APG framework. The PDS stage applies the proposed APS metric to select high-APS datasets from available datasets. Then the PG stage employs the proposed APG methods, DF-EoT, to generate patches on the determined proxy dataset. The key findings via comprehensive experiments on widely selected datasets are as follows. 1) APS is consistent with the ATFR to a certain extent and thus can serve as a PDS metric in a data-free scenario. 2) PS is consistent with the TFR to a certain extent and thus can serve as a restart selection metric to alleviate APG methods' sensitivity to initial conditions in data-free scenarios. 3) DF-EoT increases the ATFR and AUFR of the generated patches compared to EoT on a wide range of proxy datasets and thus can be adopted as an APG method in data-free scenarios.

This work mainly focuses on fooling ability achievement in data-free scenarios and eliminates environmental factors' influence. Regarding the future work, intensive study of environmental factors' influence on the patches' fooling abilities in data-free scenarios will be further considered.

5 Acknowlegement

This work was supported in part by the National Nature Science Foundation of China under Grant 61903152, in part by the Jilin Provincial Science Foundation of China under Grant 20200201162JC, Grant JJKH20200986KJ, and in part by the Exploration Foundation of State Key Laboratory of Automotive Simulation and Control, and in part by the Interdisciplinary Integration and Innovation Project of JLU under Grant JLUXKJC2020202.

REFERENCES

Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning (ICML)*, 2018.

Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *European Symposium on Security and Privacy (EuroS&P)*, 2017.

Ashutosh Chaubey, Nikhil Agrawal, Kavya Barnwal, Keerat K Guliani, and Pramod Mehta. Universal adversarial perturbations: A survey. *arXiv preprint arXiv:2005.08087*, 2020.

Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A Kai Qin, and Yun Yang. Adversarial camouflage: Hiding physical-world attacks with natural styles. In *Computer Vision and Pattern Recognition (CVPR)*, 2020.

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 1625–1634, 2018.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32:1231–1237, 2013.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *International Conference on Computer Vision (ICCV)*, 2017.

Shahar Hoory, Tzvika Shapira, Asaf Shabtai, and Yuval Elovici. Dynamic adversarial patch for evading object detection models. *arXiv preprint arXiv:2010.13070*, 2020.

Edgar Kaziakhmedov, Klim Kireev, Grigorii Melnikov, Mikhail Pautov, and Aleksandr Petiushko. Real-world attack on mtcnn face detection system. In *International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, pp. 0422–0427, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60:84–90, 2017.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision (ECCV)*, pp. 740–755, 2014.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 1765–1773, 2017.

Konda Reddy Mopuri, Utsav Garg, and R Venkatesh Babu. Fast feature fool: A data independent approach to universal adversarial perturbations. *arXiv preprint arXiv:1707.05572*, 2017.

Konda Reddy Mopuri, Phani Krishna Uppala, and R Venkatesh Babu. Ask, acquire, and attack: Data-free uap generation using class impressions. In *European Conference on Computer Vision (ECCV)*, pp. 19–34, 2018.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2017.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Sai Vemprala, Aleksander Madry, and Ashish Kapoor. Unadversarial examples: Designing objects for robust vision. *arXiv preprint arXiv:2012.12235*, 2020.

Alex Serban, Erik Poll, and Joost Visser. Adversarial examples on object recognition: A comprehensive survey. *ACM Comput. Surv.*, 2020.

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 *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1528–1540, 2016.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *International Conference on Machine Learning (ICML)*, pp. 3319–3328, 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.

Chaoning Zhang, Philipp Benz, Tooba Imtiaz, and In So Kweon. Understanding adversarial examples from the mutual influence of images and perturbations. In *Computer Vision and Pattern Recognition (CVPR)*, pp. 14521–14530, 2020.

A Additional Related Work

A.1 Image-Specific Attacks

Most adversarial attacks focus on image-specific attacks which generate perturbation specific to a given input image. (Szegedy et al., 2013) formalise adversarial perturbation generation of a specific input image as solving a box-constraint optimization problem with Limited-memory Broyden–Fletcher–Goldfarb–Shanno algorithm (L-BFGS). (Goodfellow et al., 2014) propose fast gradient sign method (FGSM) which first calculates the gradients of an input image with respect to the training loss function and then adds the gradients to the input image to perform an attack. (Kurakin et al., 2016) further propose basic iterative method (BIM), which is an iterative version of FGSM. BIM iteratively generates adversarial perturbation of an input image and thus achieves better attack performance than FGSM. (Madry et al., 2017) introduce the Projected Gradient Descent to solve perturbation's objective function, enabling attackers to conveniently deal with norm constraints of perturbations through projection operation. More recent image-specific attacks can be found in (Serban et al., 2020).

A.2 Image-Agnostic Attacks

A.2.1 Data-Dependent Attacks

Data-dependent attacks need access to the training data to guarantee perturbations' fooling abilities in image-agnostic scenarios. (Moosavi-Dezfooli et al., 2017) generate a universal perturbation by updating the perturbation on target model's training data and show that the fooling ability depends on the number of available data samples. (Sharif et al., 2016) attack a face recognition system to evade recognition or impersonate another individual by restricting the perturbation into an eyeglass frame. They optimize the pixel values of the eyeglass frame over a set of collected images of human faces, which can be seen as a subset of samples from training distribution. (Brown et al., 2017) apply EoT to generate adversarial patches where EoT traverses the training dataset to make the patches fool most training data. More recent data-dependent attacks can be found in (Eykholt et al., 2018; Thys et al., 2019; Hoory et al., 2020; Kaziakhmedov et al., 2019; Salman et al., 2020; Duan et al., 2020), where all the attacks need access to the training data or the picture of the object to be attacked.

A.2.2 Data-Free Attacks

Data-free attacks correspond to a threat model that attackers have no knowledge about the target models' training data (Chaubey et al., 2020). (Mopuri et al., 2017) first propose to maximise the mean activation values at multiple convolutional layers to generate untargeted universal perturbations in the absence of training data. (Mopuri et al., 2018) exploit class impressions extracted from the target classifier to learn a generative model for generating untargeted universal perturbations. (Zhang et al., 2020) generate targeted universal perturbations using proxy datasets which are selected for no reason.

This work mainly differs from the above attacks in the following aspects: 1) This work focuses on targeted APG problems in data-free scenarios and 2) introduces APS to select a proxy dataset. To the best of over knowledge, this is the first work focusing on targeted APG problems in strict data-free scenarios. Note that, we do not claim a state-of-art performance performance on the general adversarial patch generation problem.

B Abbreviation

Complete Spelling (Abbreviation)	**Meaning**
Adversarial Patch Generation (APG)	Generating a universal patch that attacks by replacing local pixels of input images with itself.
Proxy Dataset Selection (PDS)	Selecting a proxy dataset for patch generation from attackers' available dataset.
Expectation over Transformation (EoT)	An APG method.
Data-Free Expectation over Transformation (DF-EoT)	A variant of EoT designed for data-free scenarios.
Targeted Fooling Rate (TFR)	The ratio of training data samples fooled into the patch's target classes to all samples, a metric of patch fooling ability.
Untargeted Fooling Rate (UFR)	The ratio of misclassified training data samples fooled by the patch to all samples, a weaker metric of patch fooling ability than TFR.
Patch Saliency (PS)	A substitute patch fooling ability metric for TFR in data free s cenarios, restart selection metric in DF-EoT.
Average Targeted Fooling Rate (ATFR)	Average TFR of a group of patches generated based on the same proxy dataset, a PDS metric when training data is available.
Average Untargeted Fooling Rate (UTFR)	Average UFR of a group of patches generated based on the same proxy dataset.
Average Patch Saliency (APS)	Average PS of a group of patches generated based on the same proxy dataset, a PDS metric in data free scenarios.

C Integrated Gradient

Integrated Gradient (Sundararajan et al., 2017) is a useful technique for visualising pixel importance (i.e., saliency) in the field of explainable machine learning. Given an image-label pair $(\boldsymbol{X}, y)$ and a classier $\mathcal{F}$, the integrated gradient function $\mathcal{G}_{\mathcal{F}}$ output a saliency map by the following equation:

$$\mathcal{G}_{\mathcal{F}}(\boldsymbol{X}, y) = \int_{l_{\boldsymbol{X}_{\mathcal{G}}\boldsymbol{X}}} \frac{\partial[\mathcal{F}(\boldsymbol{X})]_y}{\partial \boldsymbol{X}} ds = (\boldsymbol{X} - \boldsymbol{X}_{\mathcal{G}}) \times \int_{\alpha=0}^{1} \frac{\partial[\mathcal{F}(\boldsymbol{X}_{\mathcal{G}} + \alpha(\boldsymbol{X} - \boldsymbol{X}_{\mathcal{G}}))]_y}{\partial \alpha} d\alpha, \quad (13)$$

where $\boldsymbol{X}_{\mathcal{G}}$ is a black image. Equation (13) is actually a path integral along the line segment between points $\boldsymbol{X}_{\mathcal{G}}$ and $\boldsymbol{X}$, and the saliency map is the importance of each component in $\boldsymbol{X}$ to the class y with respect to $\mathcal{F}$. More detailed information about the integrated gradient function can be found in (Sundararajan et al., 2017).

D Black Background Image for APS Calculation

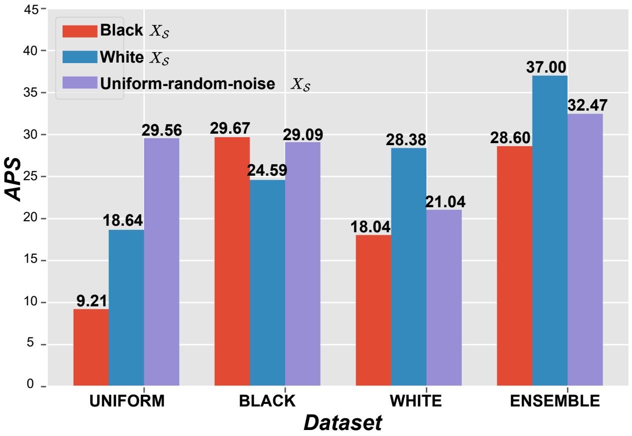

Figure 7: APSs for different background image selection. The target model is VGG-19, the considered background images for APS calculation includes a black image, white image and uniform-random-noise image.

As mentioned in Section 2.3.1, APS (PS) selects the same black background image $\boldsymbol{X}_{\mathcal{S}}$ in Equation (8) in order to measure the fooling abilities of patches (generated on different proxy datasets) on the same contextual condition. To prove the necessity of such practice, we further craft two datasets (i.e., BLACK and ENSEMBLE) mainly composed of the black images and calculates the APS of BLACK, WHITE, UNIFORM and ENSEMBLE with different background image selections such as the black, white and uniform-random-noise background images.

Figure 7 presents the calculated APSs on the target model VGG-19, where APS of each dataset varies with different choices of background images. For UNIFORM, BLACK and WHITE datasets, the APSs of each dataset reaches its highest value when the selected background image $\boldsymbol{X}_{\mathcal{S}}$ is the same as a sample from the dataset. For example, the APS of UNIFORM increases from 9.21 to 29.56 after the uniform-random-noise image replaces the black background image. Such discrepancies of APSs for the same dataset suggest that attackers should calculate the APS of different datasets on the same background image $\boldsymbol{X}_{\mathcal{S}}$. We further speculate that PS will overestimate those patches that have been iteratively updated on the background images.

Additionally, except for BLACK, the APSs of all the datasets reach their lowest values when selecting a black background. It is worth noting that even though ENSEMBLE is uniformly composed of the considered background images, the black background image still results in the lowest APS of ENSEMBLE. The above observations show that, compared with other considered background images, black background as a contextual condition has the least impact on the fooling ability measurement, which enlightens us to take a black image as the background $\boldsymbol{X}_{\mathcal{S}}$.

E Transferability Improvement of DF-EoT Compared with EoT

Table 3 presents the ATFR of patches generated on VGG-19 (proxy model) and tested on VGG-16, ResNet-34 and Inception-v1 (target models), where the ATFR actually measures the patches'

Target model	Method	Proxy Dataset	
		IMAGENET	ENSEMBLE
VGG-16	EoT	23.48%	15.82%
	DF-EoT	**26.24**%	**18.66**%
ResNet-18	EoT	0.81%	0.33%
	DF-EoT	**0.86**%	**0.55**%
Inception-v1	EoT	0.40%	0.20%
	DF-EoT	0.32%	**0.38**%

Table 3: ATFRs of patches generated on the proxy model VGG-19 and tested on the target models.

Method	**Proxy Dataset**	
	IMAGENET	ENSEMBLE
EoT	91.26%	86.44%
+ multiple-time	95.27%	90.27%
+ restart	97.26%	93.40%
+ background	78.64%	92.30%
DF-EoT	92.26%	94.09%

Table 4: Ablation study on DF-EoT with VGG-19 as the target model.

cross-model attacking generalisation ability (Transferability). As we can see, the transferability of the generated patches are rather limited for both EoT and DF-EoT. Nonetheless, DF-EoT can improve the patches' transferability to a certain extent in most cases except for the IMAGENET on Inception-v1.

F Ablation Studies on DF-EoT

Table 4 selects VGG-19 as the target model and presents ablation studies on DF-EoT. Note that EoT traverses fewer samples after adding multiple-time patch update step (400 samples v.s. 2000 samples of EoT). As we can see, multiple-time patch update and restart selection are more general techniques to improve fooling ability for both high-APS and low-APS datasets. Applying both of them enables the ATFR increase from 91.26% to 97.26% for IMAGENET, and from 86.44% to 92.03% for ENSEMBLE. Meanwhile, after adding the background update step to the multiple-step patch step, the ATFR drops from 95.27% to 78.64% for the IMAGENET, while increases from 90.27% to 92.30% for the ENSEMBLE, which suggests background update step is a useful technique for patch fooling ability improvement on low-APS datasets.

G Comparison between EoT and the Data-Free APG Framework

Method	Model				
	VGG-16	VGG-19	ResNet-18	ResNet-34	Inception-V1
EoT	95.65%	91.26%	77.30%	71.51%	60.71%
TS-DF Framework	96.47%	92.26%	76.87%	72.64%	80.71%

Table 5: ATFRs of EoT and the proposed two-stage data-free APG framework (TS-DF Framework).

Table 5 compares the final performance between EoT and the proposed data-free APG framework, where EoT uses the original training dataset while DF-EoT uses the proxy dataset selected with APS.