Leveraging Local Patch Differences in Multi-Object Scenes for Generative Adversarial Attacks

Abhishek Aich, Shasha Li, Chengyu Song, M. Salman Asif,
Srikanth V. Krishnamurthy, Amit K. Roy-Chowdhury
University of California, Riverside, USA
{aaich001@, sli057@, csong@cs., sasif@ece., krish@cs., amitrc@ece.}ucr.edu

Abstract

*State-of-the-art generative model-based attacks against image classifiers overwhelmingly focus on single-object (*i.e.*, single dominant object) images. Different from such settings, we tackle a more practical problem of generating adversarial perturbations using multi-object (*i.e.*, multiple dominant objects) images as they are representative of most real-world scenes. Our goal is to design an attack strategy that can learn from such natural scenes by leveraging the local patch differences that occur inherently in such images (*e.g. *difference between the local patch on the object 'person' and the object 'bike' in a traffic scene). Our key idea is to misclassify an adversarial multi-object image by confusing the victim classifier for each local patch in the image. Based on this, we propose a novel generative attack (called Local Patch Difference or LPD-Attack) where a novel contrastive loss function uses the aforesaid local differences in feature space of multi-object scenes to optimize the perturbation generator. Through various experiments across diverse victim convolutional neural networks, we show that our approach outperforms baseline generative attacks with highly transferable perturbations when evaluated under different white-box and black-box settings.*

1. Introduction

Understanding and exposing security vulnerabilities of deep neural networks (DNNs) has been an important recent focus of the computer vision research community [1–4]. DNNs have been extremely effective in recognition and classification systems like pedestrian recognition [5–7] and health-care applications [8, 9]. Images of real-world scenes usually consist of multiple objects. Such scenes are often analyzed by classifiers which predict all the object labels present in such images for downstream tasks such as object annotation [10–15]. Since DNNs are known to be vulnerable to adversarial attacks, it is important to understand the vulnerabilities of such multi-object classifiers. For example, scenes monitored by drones can be attacked by adversaries where all object labels detected are changed for misinterpretation at the user end [16]. Investigating such scenarios where the multi-object classifiers fail is important in order to design robust and secure real-world systems.

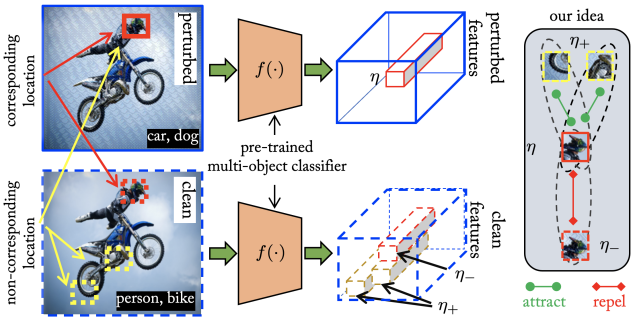


Figure 1: *Proposed attack **LPD-Attack**:* We aim to create perturbations using multi-object images. To do this, our proposed attack **LPD-Attack** leverages the rich *local* differences between the patches of features extracted from multi-object images. *e.g.*, the local feature patch of '*person's head*' will be different from local feature patch of '*bike's tire*' or '*bike's engine*'. **LPD-Attack** leverages these differences to misalign (repel) a query patch (η) from perturbed image feature with the corresponding patch (η_-) from clean image feature, while aligning (attract) with non-corresponding patches of different locations (η_+).

Adversarial attacks can be broadly classified as instance-driven approaches that are image (i.e. instance) specific [17–19]) and the distribution-driven or generative model-based approaches (*e.g.* GAP [20], CDA [21], and TDA [22]). Generative model attacks learn to craft perturbations by a generative model via training on a data distribution against a surrogate classifier. Victim classification attributes (*e.g.* kind of model architecture, data distribution, *etc.*) are generally unknown by attackers in practical cases. Hence, attackers aim towards creating strong transferable perturbations. Generative attacks provide this distinct advantage over instance-driven attacks for better transferability of perturbations for attacking unseen models [21] as well as better time complexity [20, 21, 23–25]. Our work focuses on generative attacks for learning to create perturbations using multi-object images and disrupt all labels predicted by victim classifiers. For example in Figure 1, we aim to change the labels associated with the image (*i.e.* '*person*' and

'*bike*') to labels whose objects do not exist in the input image with imperceptible perturbations (*e.g*. '*car*', '*dog*'). Existing generative model attacks (see Table 1) typically attempt to perturb images with a single dominant object in them which are analyzed by single-label classifiers. Using such single-object attacks on multi-object images would require independent object binary segmentation masks to focus on every single object in order to perturb them. This makes these attacks inefficient and impractical as an attacker cannot assume to have object binary masks for every possible distribution on the victim end.

The focus of this paper is to learn to create perturbations on multi-object images that can disrupt the output of various victim (multi-object or single-object) classifiers for *all* labels of the input image, *without* any need for independent attacks on individual objects. To this end, we propose a novel attack method that utilizes the local difference of patches in the multi-object image. As multi-object images generally contain multiple dominant objects, it is highly likely that the majority of the patches sampled are from different objects. Based on these "inherent local differences" in multi-object images, we propose a method that utilizes this property to train a perturbation generator.

Our core idea is: if the object is to be misclassified, a patch over the object should also be misclassified (in other words, make them ambiguous to the victim model). To create this misclassification, we exploit the rich local patch differences provided by multi-object images and train a perturbation generator using a novel contrastive learning loss. More specifically, given an image with multiple objects (*e.g*. '*bike*', and '*person*' in Figure 1), we aim to use the *local* difference of the feature patch on object '*bike's tire*' and the feature patch on object '*person's head*'. Assuming the size of clean and perturbed image are the same (*e.g*. 224×224), our proposed contrastive strategy **misaligns** a query patch from feature map of perturbed image (say patch from '*person's head*') with the patch from corresponding or the same location on feature map of a clean image, by simultaneously **aligning** it with patches from non-corresponding or different locations (say patch from '*bike's tire*' and '*bike's engine*') on feature map of clean image. Our **intuition** is that we want the feature patch on '*person's head*' in the perturbed image to change to some random features in order to create ambiguity and eventually confuse the victim classifier.

Unique to multi-object images, this location information is readily available in them due to the spatial arrangement of objects, *without* the need for any kind of labels or segmentation maps. Further, local patches (on average) differ from each other even if they belong to the same object, *e.g*. the shape of the engine of a bike will differ from the shape of the tyre.

Our approach is fundamentally different from prior single-label image based generative attack approaches [20, 21, 26] which do not use any kind of aforesaid *local* differences in feature maps of clean and perturbed images. Specifically, we use the approach of contrastive learning where the perturbation generator learns to disassociate corresponding signals of clean and perturbed image features, in contrast to other non-corresponding signals. In our case, these corresponding signals are patches at the same spatial location in clean and perturbed image features, while non-corresponding signals are patches at different spatial locations in the clean image features. The contrastive learning approach has been extensively used in unsupervised learning [27–30] for various image downstream tasks. We demonstrate its benefits in optimizing perturbation generating models for highly potent adversarial attacks. We refer to our attack approach as **L**ocal **P**atch **D**ifference attack or **LPD-Attack** (see Figure 2). **LPD-Attack** uses our novel local-patch contrasting approach and learns to create strong imperceptible perturbations on multi-object images.

To validate our approach, we evaluate **LPD-Attack**'s generated perturbations in different challenging scenarios. For example, if a perturbation generator is trained on Pascal-VOC [31] dataset with a Res152 [32] Pascal-VOC pre-trained multi-object classifier as a surrogate, then from the attacker's perspective, we show that **LPD-Attack** crafts highly transferable perturbations under following settings (in order of least realistic to most)

• *Setting* 1. *white-box*: victim classifier is *seen*, victim data dataset is *seen*, victim task is *seen* (*e.g*. Res152 multi-object classifier, Pascal-VOC dataset, multi-object classification task)

• *Setting* 2. *black-box*: victim classifier is *unseen*, victim data dataset is *seen*, victim task is *seen* (*e.g*. VGG19 multi-object classifier, Pascal-VOC dataset, multi-object classification task)

• *Setting* 3. *strict black-box*: victim classifier is *unseen*, victim data dataset is *unseen*, victim task is *seen* (*e.g*. VGG19 multi-object classifier, MS-COCO [33] dataset, multi-object classification task)

• *Setting* 4. *extreme black-box*: victim classifier is *unseen*, victim dataset is *unseen*, victim task is *unseen* (*e.g*. VGG16 single-label classifier, ImageNet [34] dataset, single-label classification task)

'*Setting* 4' is especially useful to test the strength of crafted perturbations by different attacks because it presents real-world use case for attackers where all victim attributes like classifier architecture, data distribution and task is unseen. To summarize, we make the following *contributions*.

1. **New practical problem.** We tackle a new problem of learning to craft perturbations for multi-object data distributions, the situation in most real-life scenes, using generative model-based attacks to disrupt decisions. To the best of our knowledge, this is the first work that considers to create **generative attacks** using multi-object images.

2. **Novel attack framework.** To this end, we propose a novel generative model-based attack approach namely **LPD-Attack**, where the perturbation generator is trained using a contrastive loss that uses rich local patch differences of multi-object image features.

3. **Extensive experiments.** Through extensive experiments on two multi-object benchmarks, we show that **LPD-Attack** has

overall better attack transferability and outperforms its baselines under aforementioned settings (see Table 2 and Table 3).

2. Related Work

Adversarial attacks on image classifiers. Most existing state-of-the-art adversarial attack works [17, 18, 20, 21, 23, 24, 26, 35–41] have been designed to attack single-object classifiers. Among these attacks, instance (or image)-driven perturbations [17, 35–37, 42] have been extensively explored, both to showcase the various shortcomings of single-object classifiers [43]. Instance-driven attacks are characterized by their method of computation of perturbations only on corresponding clean images. This results in perturbations being computed for each image individually, without using knowledge from other images [21]. The current literature on instance-driven approaches broadly consists of methods that use gradient ascent on the images [17, 19, 42, 44] or the those that generate adversarial examples using optimization-based methods [18, 40] for attacking single-object classifiers. Attacks on multi-object classifiers using instance-driven approaches have been proposed in [45–47]. [46] proposed a method to create multi-object adversarial examples by optimizing for a linear programming problem. [45] proposed a method to exploit label-ranking relationships based framework to attack multi-object ranking algorithms. More recently, [47] presented a method to disrupt the top-k labels of multi-object classifiers. Although effective for perturbing single images, instance-driven approaches are inefficient when it comes to attacking a large dataset of images, as the perturbations will have to be generated by iterating over these images individually multiple times [21, 24]. Different from [45–47], **LPD-Attack** falls under the category of *generative model-based* adversarial attacks (which we discuss next) that are distribution-driven approaches. Such approaches train a generative network over a large number of images to create perturbations. Once the model is trained, it can be used to perturb multiple images simultaneously.

Generative model-based adversarial attacks. To address the shortcomings of instance-driven approaches, generative model-based or distribution-driven attack approaches [20–25, 48] have been explored recently for learning perturbations on single-object images. For example, GAP [20] presents a distribution-driven attack that trains a generative model for creating adversarial examples by utilizing the cross-entropy loss. Recently, CDA [21] proposed a generative network that is trained using a relativistic cross-entropy loss function. Both GAP [20] and CDA [21] rely on the final classification layer of the surrogate model to train the perturbation generator which has been shown to have inferior transferability of perturbations to unknown models. Different from these, [22] presented an attack methodology to enhance the transferability of perturbations using feature separation loss functions (*e.g.* mean square error loss). However, their attack requires a manual selection of a specific mid-layer for every model against which the generator is to be trained. In contrast to these aforementioned works, **LPD-Attack** is designed to learn to craft imperceptible adversarial perturbations using *multi-object* images. Rather than focusing on the feature map globally, we take a more fine-grained approach of (feature map) patch contrasting via a novel contrastive loss. More specifically, **LPD-Attack** uses the local feature differences at multiple mid-level layers and uses an InfoNCE loss [49] based framework to create highly effectual perturbations. We summarize the differences of **LPD-Attack** with the aforementioned generative attack methods in Table 1.

Table 1: *Characteristic comparison.* Better than prior generative attacks [20–22], **LPD-Attack** is a generative attack method designed for "multi-object" images. Here, CE($\cdot$): Cross-Entropy loss, MSE($\cdot$): Mean-Square Error loss, $\boldsymbol{f}$: surrogate classifier used for training perturbation generator $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$ (weights $\boldsymbol{\theta}$). $\boldsymbol{x}$: clean image, $\boldsymbol{x}_{\boldsymbol{\delta}}$: perturbed image, and $\boldsymbol{\delta}$: perturbation. ℓ: output from specific pre-defined layer. t: misclassification label depending on type of attack (targeted or untargeted). Proposed loss ($\mathcal{L}_{\text{G}}+\mathcal{L}_{\text{LPCL}}$) is detailed in Section 3.

DD Attacks	Venue	Classifier Attack Strategy	
		image type?	$\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$ **loss**
GAP [20]	CVPR2018	single-object	CE($\boldsymbol{f}(\boldsymbol{x}_{\boldsymbol{\delta}})$, t)
CDA [21]	NeurIPS2019	single-object	CE($\boldsymbol{f}(\boldsymbol{x}_{\boldsymbol{\delta}})$ - $\boldsymbol{f}(\boldsymbol{x})$, t)
TDA [22]	NeurIPS2021	single-object	MSE($\boldsymbol{f}_{\ell}(\boldsymbol{x}_{\boldsymbol{\delta}})$, $\boldsymbol{f}_{\ell}(\boldsymbol{x})$)
LPD-Attack	Ours	multi-object	$\mathcal{L}_{\text{G}}+\mathcal{L}_{\text{LPCL}}$

3. Proposed Attack Methodology

Here, we explain our proposed generative adversarial attack **LPD-Attack** that learns from multi-object images. It includes training the perturbation generator with a novel local patch contrasting learning loss that uses local regions of features extracted from clean and perturbed images. We start with the notations and defining the problem statement.

3.1. Problem Formulation

Notations. Let C be the total number of classes and N be the number of training samples in a dataset $\mathcal{T}$. We define $\mathcal{T} = \{(\boldsymbol{x}^{(1)},\boldsymbol{y}^{(1)}),\cdots,(\boldsymbol{x}^{(N)},\boldsymbol{y}^{(N)})\}$ where $\boldsymbol{x}^{(i)} \in \mathbb{R}^{H\times W\times C}$ and $\boldsymbol{y}^{(i)} = [y_1^{(i)},\cdots,y_C^{(i)}] \in \mathcal{Y} \subseteq \{0,1\}^C$ are the ith image (with height H, width W, and channels Z) and ground-truth label vector, respectively. For an example data point $\boldsymbol{x}^{(i)}$ and class c, $y_c^{(i)} = 1$ (or $=0$) indicates the presence (or absence) of an object from class c in $\boldsymbol{x}^{(i)}$. We define a surrogate multi-object classifier trained on $\mathcal{T}$ as $\boldsymbol{f}(\cdot)$, which is utilized to train a perturbation generator $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$ (parameterized by weight $\boldsymbol{\theta}$). In further discussions and Figure 2, we drop the superscript i for ease of exposition.

Problem Statement. Given a clean multi-object image $\boldsymbol{x}$ from data-distribution $\mathcal{T}$ containing multiple dominant objects and the victim classifier $\boldsymbol{g}(\cdot)$, we aim to flip *all* labels of $\boldsymbol{x}$ with an allowable perturbation budget ϵ defined by an ℓ_∞ norm. Specifically, the problem objective is to craft a perturbation $\boldsymbol{\delta}$ such that the prediction of $\boldsymbol{g}(\cdot)$ for all labels $\boldsymbol{y}$ associated with $\boldsymbol{x}$ is changed. Mathematically, this can be represented as $\boldsymbol{y} \neq \widehat{\boldsymbol{y}}$ where, $\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x})$ and $\widehat{\boldsymbol{y}} = \boldsymbol{g}(\boldsymbol{x}+\boldsymbol{\delta})$ with $\|\boldsymbol{\delta}\|_\infty \leq \epsilon$.

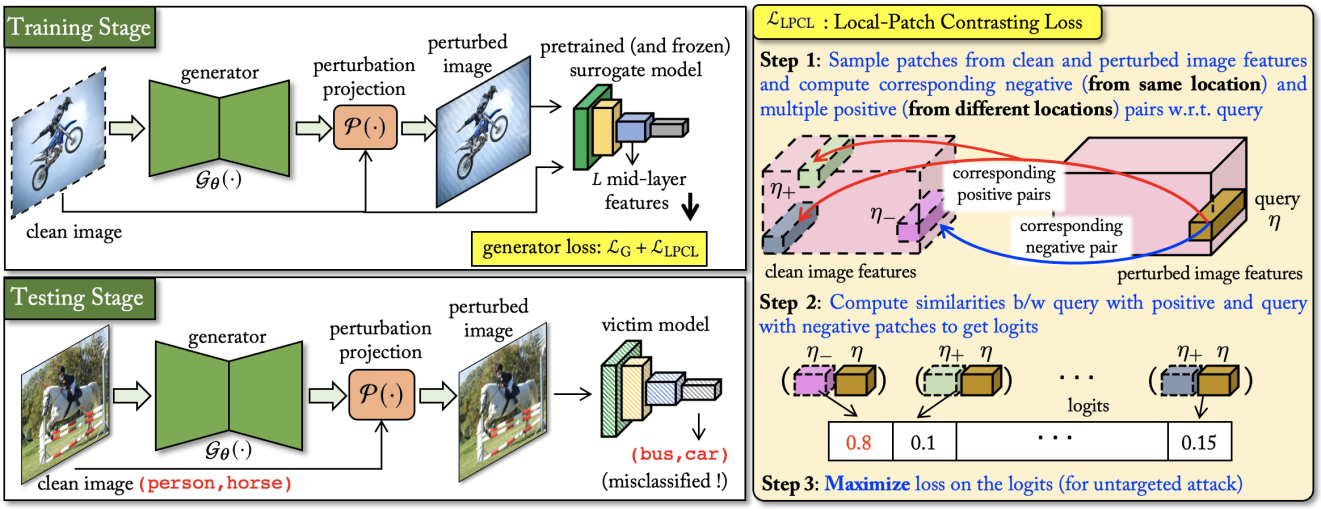


Figure 2: *Framework overview.* Our proposed **LPD-Attack** framework (*top*) aims to learn from multi-object images using a contrastive learning mechanism ($\mathcal{L}_{\text{LPCL}}$) to maximize the difference of corresponding patches of same locations while minimizing the difference between non-corresponding patches of distinct locations, from features extracted from clean and perturbed images. This results in highly effective and transferable perturbations for input clean images during inference (*bottom-left*).

3.2. Proposed Approach: LPD-Attack

Our proposed framework is presented in Figure 2. It contains a perturbation generator $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$ that is trained to craft imperceptible perturbations $\boldsymbol{\delta}$ on $\boldsymbol{x}$. $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$ is trained against a surrogate pre-trained multi-object classifier $\boldsymbol{f}(\cdot)$. More precisely, $\boldsymbol{f}(\cdot)$ acts as a discriminator against which generator $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$ is trained ($\boldsymbol{f}(\cdot)$ remains fixed or frozen). During training, $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$ takes $\boldsymbol{x}$ as input and generates an unbounded perturbed image $\mathcal{G}(\boldsymbol{x}) = \widehat{\boldsymbol{x}_{\boldsymbol{\delta}}}$. This unbounded perturbed image $\widehat{\boldsymbol{x}_{\boldsymbol{\delta}}}$ is clipped to be within an pre-defined perturbation budget ϵ on $\boldsymbol{x}$ under the ℓ_∞ norm using the projection operator $\mathcal{P}(\cdot)$. The perturbed image is then estimated as $\boldsymbol{x}_{\boldsymbol{\delta}} = \mathcal{P}(\widehat{\boldsymbol{x}_{\boldsymbol{\delta}}})$. To compute the generator loss, $\boldsymbol{x}_{\boldsymbol{\delta}}$ is sent to the discriminator, $\boldsymbol{f}(\cdot)$, to be misclassified. At multiple L mid-layers from $\boldsymbol{f}(\cdot)$, we compute the features of clean image $\big[\boldsymbol{f}_k(\boldsymbol{x})\big]_{k=1}^{L}$ and features of perturbed image $\big[\boldsymbol{f}_k(\boldsymbol{x}_{\boldsymbol{\delta}})\big]_{k=1}^{L}$, where $\boldsymbol{f}_k(\boldsymbol{x}), \boldsymbol{f}_k(\boldsymbol{x}_{\boldsymbol{\delta}}) \in \mathbb{R}^{h_k \times w_k \times c_k}$. Here, $h_k \times w_k$ denote the spatial size of ith layer feature map with c_k channels. The effectiveness of using mid-level features to craft powerful perturbations have been extensively studied in [24, 26, 50–53]. Therefore, we leverage these mid-level features of $\boldsymbol{f}(\cdot)$ and define our generative model loss via two functions. The *first* loss function is a global loss $\mathcal{L}_{\text{G}}$ that compares extracted features directly as follows:

$$\mathcal{L}_{\text{G}} = \frac{1}{L}\sum_{k=1}^{L} \text{dist}\big(\boldsymbol{f}_k(\boldsymbol{x}), \boldsymbol{f}_k(\boldsymbol{x}_{\boldsymbol{\delta}})\big) \tag{1}$$

Here, dist$(\cdot)$ can be any distance measuring function, *e.g.* mean square error function, etc. The *second* loss function is a novel objective, namely, Local-Patch Contrasting Loss (LPCL) which compares the extracted features $\big[\boldsymbol{f}_k(\boldsymbol{x})\big]_{k=1}^{L}$ and $\big[\boldsymbol{f}_k(\boldsymbol{x}_{\boldsymbol{\delta}})\big]_{k=1}^{L}$ at a local or patch level. Better than prior generative attacks which only compare perturbed and clean images globally, our proposed LPCL loss leverages the local difference of patches from multiple objects in the input image to disrupt the victim classifier's decisions. We expand on the details of LPCL next.

Algorithm 1: LPD-Attack Training Algorithm

Input : clean images $\boldsymbol{x}$ from distribution $\mathcal{T}$, perturbation ℓ_∞ bound ϵ, surrogate classifier $\boldsymbol{f}(\cdot)$
Input : learning rate α
Output : perturbation generator $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$'s weights $\boldsymbol{\theta}$

/* Large-Scale training of $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$ */
Randomly initialize $\boldsymbol{\theta}$
Load and freeze multi-object classifier $\boldsymbol{f}(\cdot)$ trained on $\mathcal{T}$
while *not done* **do**
  /* Obtain clean image features */
  Input $\boldsymbol{x}$ to $\boldsymbol{f}(\cdot)$ and get L mid-layer features $\big[\boldsymbol{f}_k(\boldsymbol{x})\big]_{k=1}^{L}$
  /* Obtain perturbed image features */
  Create unbounded perturbed image $\mathcal{G}(\boldsymbol{x})$
  Project it within bound ϵ using $\mathcal{P}(\cdot)$ to obtain $\boldsymbol{x}_{\boldsymbol{\delta}}$
  Input $\boldsymbol{x}_{\boldsymbol{\delta}}$ to $\boldsymbol{f}(\cdot)$, get L mid-layer features $\big[\boldsymbol{f}_k(\boldsymbol{x}_{\boldsymbol{\delta}})\big]_{k=1}^{L}$
  /* Compute loss */
  Compute $\mathcal{L} = \mathcal{L}_{\text{G}} + \mathcal{L}_{\text{LPCL}}$
  /* Update $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$'s weights */
  Update $\boldsymbol{\theta}$ with respect to $\mathcal{L}$ using Adam
  $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} \mathcal{L}(\boldsymbol{\theta})$

3.3. Contrasting Patches of Multi-Object Images

Motivation. We make the observation that due to existence of multiple objects in a multi-object image $\boldsymbol{x}$, we can utilize the local feature patches from $\boldsymbol{f}_k(\boldsymbol{x})$ (and $\boldsymbol{f}_k(\boldsymbol{x}_{\boldsymbol{\delta}})$). The local patches of input clean image belong to individual dominant objects and thus, prompt the multi-object classifier to output their respective

associated labels. Therefore, for each object in a perturbed image to be misclassified, each patch within its feature map should look different to the classifier than the same location corresponding patch in the feature map of a clean image. To create this difference, we use the feature maps from different location non-corresponding patches to create ambiguity for the victim classifier to prompt incorrect decisions on the overall perturbed image. This patch location-wise contrasting of the clean and perturbed image features at the local level allows for stronger supervision for training the perturbation generator $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$.

Proposed contrasting loss ($\mathcal{L}_{\text{LPCL}}$). To misclassify the perturbed image $\boldsymbol{x}_{\delta}$, we need to maximize the difference between its features and that of the clean image $\boldsymbol{x}$. We propose to achieve this by misaligning corresponding clean-perturbed image feature patches at a specific location to maximize the difference at a local level. This misalignment is enabled by utilizing the other patches from the clean image features at non-corresponding locations. We start with computing the features of clean and perturbed image from surrogate model $\boldsymbol{f}(\cdot)$ as $\big[\boldsymbol{f}_k(\boldsymbol{x}) \in \mathbb{R}^{h_k \times w_k \times c_k}\big]_{k=1}^{L}$ and $\big[\boldsymbol{f}_k(\boldsymbol{x}_{\delta}) \in \mathbb{R}^{h_k \times w_k \times c_k}\big]_{k=1}^{L}$, respectively. We convert these feature maps to tensors $\boldsymbol{D}_k$ and $\widehat{\boldsymbol{D}}_k$, respectively, of size $v_k \times c_k$ (where, $v_k = h_k w_k$). Next, we chose a query vector $\boldsymbol{\eta}_k^q \in \mathbb{R}^{c_k}$ from a qth spatial location of $\widehat{\boldsymbol{D}}_k$ and choose the corresponding spatial location vector $\boldsymbol{\eta}_k^-$ from $\boldsymbol{D}_k$, which we call $\widehat{\boldsymbol{\eta}}_k$'s negative. Then, from R other (or different) locations of $\boldsymbol{D}_k$, we choose a collection of positives denoted by $\boldsymbol{\eta}_k^+ \in \mathbb{R}^{R \times c_k}$. The $\mathcal{L}_{\text{LPCL}}$ loss is now defined as an $(R+1)$-way classification objective, with logits representing the similarity between query $\boldsymbol{\eta}_k^q$ and set $[\boldsymbol{\eta}_k^-, \boldsymbol{\eta}_{k1}^+, \boldsymbol{\eta}_{k2}^+, \cdots, \boldsymbol{\eta}_{kR}^+]$, as follows.

$$\mathcal{L}_{\text{LPCL}} = -\frac{1}{L}\sum_{k=1}^{L} \log\left(\frac{\exp\big(sim(\boldsymbol{\eta}_k^q, \boldsymbol{\eta}_k^-)\big)}{\exp\big(sim(\boldsymbol{\eta}_k^q, \boldsymbol{\eta}_k^-)\big) + \sum_{r=1}^{R} \exp\big(sim(\boldsymbol{\eta}_k^q, \boldsymbol{\eta}_{kr}^+)\big)}\right) \tag{2}$$

where $sim(\boldsymbol{\eta}_a, \boldsymbol{\eta}_b) = \boldsymbol{\eta}_a^{\top} \boldsymbol{\eta}_b / \tau$ returns the similarity between two vectors, $\top$ represents the transpose operation, and τ is a scaling parameter. We set $\tau = 0.07$ following [27]. This loss envisions our idea that, if a feature patch on the perturbed image is to be disrupted, it should obtain a low similarity score with the corresponding (same location) "negative" feature patch of the clean image, and high similarity score with "positive" patches from non-corresponding locations. Note that "patch" does not correspond to "object" and it is possible that (*1*) group of patches can belong to one object and (*2*) one patch can contain parts of multiple objects. The only requirement for the R positive patches used in $\mathcal{L}_{\text{LPCL}}$ to operate properly is: these R positive patches should contain feature values that are different from the values in query feature patch $\boldsymbol{\eta}_k^q$. This requirement is easily fulfilled when we sample them from non-overlapping w.r.t. to each other and from different locations w.r.t. to $\boldsymbol{\eta}_k^q$.

3.4. Final Objective

Our final learning objective includes a loss function to train the generator over $\boldsymbol{x}_{\delta}$ both *globally* with a $\mathcal{L}_{\text{G}}$ objective and *locally* using our proposed contrasting loss $\mathcal{L}_{\text{LPCL}}$. This loss is computed over multiple L mid-level layers of $\boldsymbol{f}(\cdot)$ as $\mathcal{L} = \mathcal{L}_{\text{G}} + \mathcal{L}_{\text{LPCL}}$. Note that we maximize $\mathcal{L}$ for an untargeted attack with $\mathcal{L}_{\text{G}}$ set as the mean square error loss. For targeted attack, we minimize $\mathcal{L}$ with $\mathcal{L}_{\text{G}}$ set as binary cross-entropy loss to classify the perturbed image to the target label. The whole training procedure is summarized in Algorithm 1. During testing, we simply input the test image to the trained generator to create a perturbed image with the aim to fool the victim classifier for all the associated labels.

4. Experiments and Results

Here, we discuss the strength of **LPD-Attack** under diverse attack *Settings 1-4* (as described in Section 1) presented in Table 2 and Table 3. Furthermore, we analyze the strength of **LPD-Attack** on most realistic attack setting in Table 4 and Table 5, as well as other easier variations in Table 6. We also perform an ablation analysis of **LPD-Attack** in Figure 3 and show some examples of perturbed images and attention shift in Figure 4 to validate our method. Unless otherwise stated, perturbation budget is set to $\ell_{\infty} \leq 10$. We provide details of implementation, baselines (GAP [20], CDA [21], TDA [22]), and additional experiments in the Supplementary Material.

Training Datasets. We employ widely-used and publicly available PASCAL-VOC [31] and MS-COCO [33] datasets. For Pascal-VOC, we use *trainval* from 'VOC2007' and 'VOC2012' as our training dataset and the evaluations are carried out on 'VOC2007_test' dataset. For MS-COCO, we use *train2017* as our training dataset and *val2017* for evaluations.

Inference Metrics. We evaluate the attacks on multi-object classifiers using accuracy on test set defined for multi-object classification in [54, 55]. For attacks on single-object classifier in Table 4 and Table 5, we use top-1 accuracy on test set. For all untargeted attacks, a lower score indicates better attack. In case of targeted attack, a higher score indicates better attack result. Best results are in red, second best are in blue. Accuracy on clean images are provided in gray for reference.

Victim Models and Attack Settings. To attack the victim models, we first train all the perturbation model $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$ for baselines and **LPD-Attack** on Pascal-VOC and MS-COCO on their respective train set against surrogate multi-classifier model $\boldsymbol{f}(\cdot)$. We chose $\boldsymbol{f}(\cdot)$ to be (Pascal-VOC or MS-COCO) pre-trained multi-object classifiers Res152 [32], Dense169 [56], and VGG19 [57]. As discussed in Section 1, we then evaluate the trained $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$ under four following settings. Firstly for *Setting* 1 (white-box), we attack the surrogate multi-classifier model $\boldsymbol{f}(\cdot)$ on test set of *same* multi-object distribution used during training. Secondly for *Setting* 2 (black-box), we attack *other* multi-object classifiers different from the surrogate model also on test set of *same* multi-object distribution used during training. Thirdly for *Setting* 3 (strict black-box), we attack multi-object classifiers on test set of *different* multi-object distribution *other than* used during training. Finally, following [24] for *Setting* 4

Table 2: *Average Results when $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$ trained with Pascal-VOC*. We summarize the attack capability of prior generative attack works under various victim scenarios with training data as Pascal-VOC. Results are averaged over three surrogate classifiers for all methods.

Attack	Victim Details	Method	Mean result
Setting 1 (easy)	Pascal-VOC (victim model = surrogate model)	GAP [20]	55.22
		CDA [21]	54.79
		TDA [22]	53.73
		LPD-Attack	52.69
Setting 2	Pascal-VOC (victim model $\neq$ surrogate model)	GAP [20]	56.24
		CDA [21]	55.86
		TDA [22]	55.32
		LPD-Attack	54.37
Setting 3	MS-COCO	GAP [20]	40.86
		CDA [21]	40.51
		TDA [22]	39.79
		LPD-Attack	38.69
Setting 4 (difficult)	CIFAR(10,100), STL-10, SVHN (Coarse-Grained tasks)	GAP [20]	83.96
		CDA [21]	82.72
		TDA [22]	83.13
		LPD-Attack	70.72
	CUB-200, Stanford Cars, FGVC Aircraft (Fine-Grained tasks)	GAP [20]	90.50
		CDA [21]	90.21
		TDA [22]	88.61
		LPD-Attack	73.72
	ImageNet	GAP [20]	73.05
		CDA [21]	72.33
		TDA [22]	69.91
		LPD-Attack	45.12

Table 3: *Average Results when $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$ trained with MS-COCO* We summarize the attack capability of prior generative attack works under various victim scenarios with training data as MS-COCO. Results are averaged over three surrogate classifiers for all methods.

Attack	Victim Details	Method	Mean result
Setting 1 (easy)	MS-COCO (victim model = surrogate model)	GAP [20]	41.09
		CDA [21]	39.96
		TDA [22]	34.31
		LPD-Attack	34.91
Setting 2	MS-COCO (victim model $\neq$ surrogate model)	GAP [20]	41.05
		CDA [21]	41.17
		TDA [22]	37.08
		LPD-Attack	36.97
Setting 3	Pascal-VOC	GAP [20]	56.03
		CDA [21]	55.63
		TDA [22]	51.84
		LPD-Attack	52.06
Setting 4 (difficult)	CIFAR(10,100), STL-10, SVHN (Coarse-Grained tasks)	GAP [20]	84.07
		CDA [21]	81.52
		TDA [22]	70.40
		LPD-Attack	65.53
	CUB-200, Stanford Cars, FGVC Aircraft (Fine-Grained tasks)	GAP [20]	90.64
		CDA [21]	89.98
		TDA [22]	74.88
		LPD-Attack	63.39
	ImageNet	GAP [20]	73.25
		CDA [21]	71.94
		TDA [22]	42.37
		LPD-Attack	27.51

(extreme black-box), we attack various single-object classifiers for CIFAR10 [58], CIFAR100 [58], STL-10 [59], and SVHN [60] (coarse-grained tasks), CUB-200-2011 [61], Stanford Cars [62], and FGVC Aircrafts [63] (fine-grained tasks), and ImageNet [64] models on their respective test sets. The pre-trained victim models of coarse-grained tasks are available in [65], for fine-grained tasks (Res50 [32] and SENet154 [66]) in [67] and ImageNet task in [68]. Briefly, the coarse-grained single-object classification task is to distinguish labels like 'cats *vs* dogs', whereas the fine-grained single-object classification task is to distinguish difficult labels like species of cats (*e.g.* 'tiger *vs* panther'). Analyzing attacks on such diverse tasks after learning perturbations from multi-object images will show the transferability of perturbations which is important for real-world attacks.

4.1. Quantitative Results

We evaluated **LPD-Attack** against baselines under four different attack scenarios. We summarize them in Table 2 and Table 3, and discuss them below.

Observation 1. *The proposed method **LPD-Attack** has the overall best performance.* We outperform the prior best SOTA method TDA [22] in 10 out of 12 cases, demonstrating the efficacy of our proposed method. For example in Pascal-VOC, we outperform TDA by a margin of 10% (ours: 55.88%, TDA: 65.08%), and in MS-COCO, by a margin of 3.5% (ours: 46.73%, TDA: 51.00%). Furthermore, TDA carries an expensive computational overhead (discussed by the authors themselves in Section 4.6 under "Limitations"): the attacker needs to incur high time complexity (by training the generator separately for each possible mid-layer) to search for the most effective mid-layer of the surrogate model in order to optimize the generator. Through our results, especially on the ImageNet dataset, we show that TDA's manually selected specific layer is highly sensitive to the training data distribution as the results on ImageNet degrade drastically if the generator is trained on datasets different from ImageNet (in this case, Pascal-VOC, MS-COCO). In contrast, since we select a group of layers, we do not need this laborious time and resource-consuming analysis.

Observation 2. *SOTA tends to comparatively overfit more to the attacker's training data distribution than the proposed method.* The aforementioned four attack scenarios (after the generator is trained on Pascal-VOC and MS-COCO) show that: as the victim data distribution starts varying (*e.g.* ImageNet, STL-10, FGVC Aircraft classification), there is a huge performance drop in the prior attacks due to weaker transferability of perturbations. For example, TDA shows a comparable performance when the victim distribution is similar to the attacker's training distribution (see Table 6) but shows surprisingly low attack results (20% difference) when the victim distribution changes to single-object classifications tasks like ImageNet, STL-10, FGVC Aircraft (see Table 4). This clearly demonstrates that prior works tend to overfit to the attacker's training distribution and perform poorly when there is no overlap in the victim's data distribution and type of classification task. On the other hand, our proposed method **LPD-Attack** alleviates this issue and shows better transferability of perturbations. We attribute the better performance of our method, in better alleviating the

Table 4: *Setting 4 attack comparison when $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$ is trained with Pascal-VOC:* Perturbations created on test set of each task. $\boldsymbol{f}(\cdot)$: Res152.

(a) Coarse-Grained task

Method	CIFAR10	CIFAR100	STL-10	SVHN
	All Victim Models from [65]			
	93.79%	74.28%	77.60%	96.03%
GAP [20]	92.94%	72.56%	74.33%	96.01%
CDA [21]	91.97%	72.18%	70.99%	95.74%
TDA [22]	92.49%	70.80%	73.31%	95.93%
LPD-Attack	76.61%	47.51%	70.49%	88.27%

(b) Fine-Grained tasks

Method	CUB-200-2011		Stanford Cars		FGVC Aircraft	
	Res50	SENet154	Res50	SENet154	Res50	SENet154
	87.35%	86.81%	94.35%	93.36%	92.23%	92.05%
GAP [20]	86.24%	86.40%	93.79%	93.09%	91.69%	91.78%
CDA [21]	85.90%	86.11%	93.28%	92.69%	91.36%	91.90%
TDA [22]	83.93%	82.33%	92.92%	91.79%	90.04%	90.64%
LPD-Attack	59.34%	76.58%	77.35%	81.98%	73.78%	73.27%

(c) ImageNet task (on ImageNet validation set (50k samples))

Method	ImageNet Trained Victim Classifiers					
	VGG16	VGG19	Res50	Res152	Dense121	Dense169
	70.15%	70.95%	74.60%	77.34%	74.21%	75.74%
GAP [20]	69.19%	70.23%	73.71%	76.62%	73.36%	75.21%
CDA [21]	68.20%	69.41%	72.67%	75.95%	72.93%	74.79%
TDA [22]	65.60%	66.28%	70.47%	74.35%	70.11%	72.62%
LPD-Attack	32.24%	35.05%	48.53%	50.54%	49.99%	54.37%

Table 5: *Setting 4 attack comparison when $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$ is trained with MS-COCO:* Perturbations created on test set of each task. $\boldsymbol{f}(\cdot)$: Dense169.

(a) Coarse-Grained task

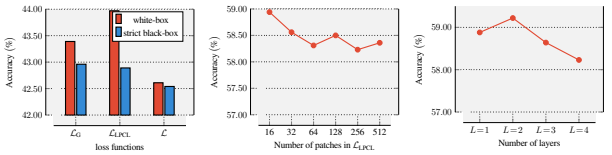
Method	CIFAR10	CIFAR100	STL-10	SVHN
	All Victim Models from [65]			
	93.79%	74.28%	77.60%	96.03%
GAP [20]	93.12%	72.72%	74.78%	95.65%
CDA [21]	90.77%	69.20%	70.31%	95.79%
TDA [22]	76.37%	40.35%	72.19%	92.67%
LPD-Attack	66.16%	35.12%	70.28%	90.56%

(b) Fine-Grained tasks

Method	CUB-200-2011		Stanford Cars		FGVC Aircraft	
	Res50	SENet154	Res50	SENet154	Res50	SENet154
	87.35%	86.81%	94.35%	93.36%	92.23%	92.05%
GAP [20]	86.69%	86.33%	94.12%	93.10%	91.84%	91.78%
CDA [21]	85.57%	86.04%	93.10%	92.71%	91.15%	91.30%
TDA [22]	60.30%	70.04%	76.21%	80.48%	81.07%	81.19%
LPD-Attack	22.25%	74.77%	64.98%	81.31%	60.37%	76.66%

(c) ImageNet task (on ImageNet validation set (50k samples))

Method	ImageNet Trained Victim Classifiers					
	VGG16	VGG19	Res50	Res152	Dense121	Dense169
	70.15%	70.95%	74.60%	77.34%	74.21%	75.74%
GAP [20]	69.32%	70.39%	73.89%	76.75%	73.75%	75.38%
CDA [21]	67.24%	68.45%	72.17%	75.69%	73.12%	74.96%
TDA [22]	31.59%	33.11%	45.74%	58.15%	46.11%	39.49%
LPD-Attack	20.60%	23.60%	30.42%	37.07%	29.50%	23.88%



(a) loss analysis (b) Impact of R (c) Impact of L

Figure 3: *Ablation analysis of **LPD-Attack**:* Figure 3(a): $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$ trained on Pascal-VOC against Res152, strict black-box attacks on MS-COCO; Figure 3(b), Figure 3(c): $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$ trained on Pascal-VOC against Dense169 for all cases; perturbation bound was set $\ell_\infty \leq 10$.

overfitting issue than SOTA, to the unique strategy of comparing local feature patches rather than just global differences.

Observation 3. *As attack scenarios become more difficult and realistic, the proposed method's performance is much better than the SOTA baselines*. White-box attacks (*Setting* 1) are easy and least realistic attacks, whereas extreme black-box attacks (*Setting* 4) are the most difficult but most realistic (the attacker has no knowledge of the victim model or task) attack settings. We observe that as the difficulty level of attack increases, the performance of TDA crafted perturbations show increasingly poor performance than the proposed method **LPD-Attack**. For example, though **LPD-Attack** and TDA show comparable performance in the white-box attacks, it outperforms TDA by a huge margin of 18% in extreme black-box attacks (see Table 5 and Table 4). This implies existing attacks perform poorly in real-world use cases, whereas **LPD-Attack** poses a greater threat to the victim model than prior SOTA attacks.

Targeted attacks. We performed a white-box targeted attack on Dense169 with the target label set to 'person' (*i.e.* all perturbed images should output the label 'person'). We observed that GAP [20] and CDA [21] result in an accuracy of 34.58% and 34.86% whereas **LPD-Attack** resulted in 35.00% attack performance (perturbation bound $\ell_\infty \leq 16$).

4.2. Ablation Study

We perform an ablation analysis of **LPD-Attack** with respect to loss objectives in Figure 3(a), impact of number of patches R in Figure 3(b), and impact of number of layers L in Figure 3(c) utilized from the surrogate model $\boldsymbol{f}(\cdot)$ to train $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$. From Figure 3(a), we observe the impact of components of our loss objective when $\mathcal{G}_{\boldsymbol{\theta}}(\cdot)$ was trained against Res152 on Pascal-VOC both for white-box (test against Pascal-VOC) and strict black-box (test against MS-COCO). It can be observed that the perturbations are most effective when both the global loss $\mathcal{L}_{\text{G}}$ and local loss $\mathcal{L}_{\text{LPCL}}$ are utilized. Next from Figure 3(b), we observe that the best performance is observed with $R = 256$ patches (note that we use $R = 128$ for a slightly better training time-accuracy trade-off). Finally, we analyze the impact of using multiple mid-level features from $\boldsymbol{f}(\cdot)$ and observe that $L = 4$ results in best attacks as it allows the use of diverse features to learn the perturbations. This also shows that we do not need to manually choose a specific layer for better attacks as in the case of TDA [22], and an average choice of a group of layers creates effective attacks.

4.3. Qualitative Results

We visualize some examples of perturbed images and shift in attention (using CAM [69]) for misclassified images from clean

Table 6: *Generative Attack Comparison when $\mathcal{G}_{\theta}(\cdot)$ is trained with Pascal-VOC:* Gray colored cells represent the *Setting* 1 attacks. $f(\cdot)$ in both Table 6(a) and Table 6(b) are pre-trained on Pascal-VOC.

(a) *Setting* 1 and *Setting* 2 attacks

$f(\cdot)$	Method	Pascal-VOC Trained Victim Models		
		Res152	**VGG19**	**Dense169**
		83.12%	83.18%	83.73%
Res152	GAP [20]	58.78%	48.52%	61.31%
	CDA [21]	58.62%	48.69%	60.93%
	TDA [22]	58.45%	48.19%	61.16%
	LPD-Attack	57.22%	46.07%	59.63%
VGG19	GAP [20]	58.88%	45.60%	61.32%
	CDA [21]	58.18%	45.26%	60.73%
	TDA [22]	57.47%	42.61%	59.39%
	LPD-Attack	57.84%	42.62%	59.66%
Dense169	GAP [20]	58.83%	48.58%	61.29%
	CDA [21]	58.39%	48.25%	60.48%
	TDA [22]	58.04%	47.66%	60.12%
	LPD-Attack	57.21%	45.82%	58.23%

(b) *Setting* 3 attacks

$f(\cdot)$	Method	MS-COCO Trained Victim Models		
		Res152	**VGG19**	**Dense169**
		67.95%	66.49%	67.60%
Res152	GAP [20]	44.91%	34.70%	44.15%
	CDA [21]	44.99%	34.89%	44.35%
	TDA [22]	44.45%	34.46%	43.88%
	LPD-Attack	42.36%	32.16%	42.37%
VGG19	GAP [20]	45.02%	31.10%	44.14%
	CDA [21]	43.41%	30.94%	43.31%
	TDA [22]	43.22%	27.74%	42.23%
	LPD-Attack	43.12%	28.31%	42.54%
Dense169	GAP [20]	44.88%	34.72%	44.12%
	CDA [21]	44.50%	34.42%	43.82%
	TDA [22]	44.21%	34.30%	43.58%
	LPD-Attack	43.09%	32.40%	41.86%

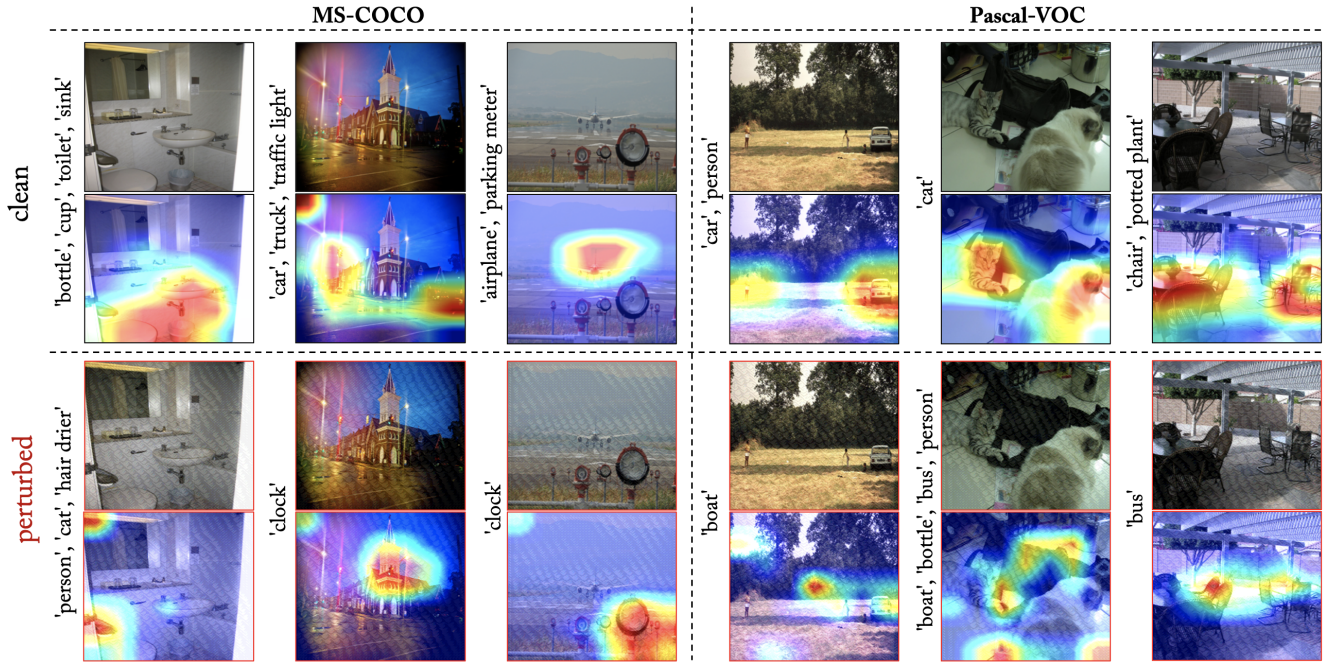


Figure 4: *Illustration of perturbed images and attention shift:* Row 1: clean images, Row 2: CAM [69] attention map on clean images, Row 3: perturbed images ($\ell_\infty \leq 10$), Row 4: CAM [69] attention map on perturbed images. $\mathcal{G}_{\theta}(\cdot)$ was trained against Res152 for both datasets, examples are visualized on test sets with attention maps extracted from Res152.

images in Pascal-VOC and MS-COCO in Figure 4 for Res152 multi-object classifier. It can be observed that **LPD-Attack** changes the focus of the victim classifier to irrelevant regions leading to highly successful attacks.

5. Conclusion

In this paper, we tackle a novel problem of altering the decisions of victim classifiers by learning to create perturbations on multi-object images. To this end, we proposed a novel generative adversarial attack (**LPD-Attack**) framework that trains the perturbation generators by exploiting the local differences in multi-object image features. **LPD-Attack** achieves high attack rates both in white-box and different practical black-box settings. For example, when we learn to craft perturbations on Pascal-VOC and create black-box attack on ImageNet, **LPD-Attack** outperforms existing attacks by ∼25% points. In our future work, we will explore the case of black-box multi-object targeted attacks for multi-object images, as well as video generative models [70, 71] for adversarial attacks on video classifiers.

Acknowledgement. This material is based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00112090096.

References

[1] Zhaoyin et.al. Jia. Object detection neural networks, June 11 2019. US Patent 10,318,827.

[2] Jiajun et.al. Zhu. Method and system for hierarchical human/crowd behavior detection, June 25 2020. US Patent 10,572,717.

[3] Georgescu et.al. Bogdan. Method and system for anatomical object detection using marginal space deep neural networks, June 15 2017. US Patent 9,730,643.

[4] Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A Kai Qin, and Yun Yang. Adversarial camouflage: Hiding physical-world attacks with natural styles. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1000–1008. IEEE, 2020.

[5] Mate Szarvas, Akira Yoshizawa, Munetaka Yamamoto, and Jun Ogata. Pedestrian detection with convolutional neural networks. In *IEEE Proceedings. Intelligent Vehicles Symposium, 2005.*, pages 224–229. IEEE, 2005.

[6] Jonathan Lwowski, Prasanna Kolar, Patrick Benavidez, Paul Rad, John J Prevost, and Mo Jamshidi. Pedestrian detection system for smart communities using deep convolutional neural networks. In *2017 12th System of Systems Engineering Conference (SoSE)*, pages 1–6. IEEE, 2017.

[7] Abhishek Aich, Meng Zheng, Srikrishna Karanam, Terrence Chen, Amit K. Roy-Chowdhury, and Ziyan Wu. Spatio-temporal representation factorization for video-based person re-identification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 152–162, October 2021.

[8] Eugene Shkolyar, Xiao Jia, Timothy C Chang, Dharati Trivedi, Kathleen E Mach, Max Q-H Meng, Lei Xing, and Joseph C Liao. Augmented bladder tumor detection using deep learning. *European urology*, 76(6):714–718, 2019.

[9] Angel Cruz-Roa, Hannah Gilmore, Ajay Basavanhally, Michael Feldman, Shridar Ganesan, Natalie NC Shih, John Tomaszewski, Fabio A González, and Anant Madabhushi. Accurate and reproducible invasive breast cancer detection in whole-slide images: A deep learning approach for quantifying tumor extent. *Scientific reports*, 7(1):1–14, 2017.

[10] Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.

[11] Min-Ling Zhang and Zhi-Hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.

[12] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine learning*, page 333, 2011.

[13] Shang-Fu Chen, Yi-Chen Chen, Chih-Kuan Yeh, and Yu-Chiang Wang. Order-free rnn with visual attention for multi-label classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, 2018.

[14] Erin L Allwein, Robert E Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of machine learning research*, 1(Dec):113–141, 2000.

[15] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-Label Image Recognition with Graph Convolutional Networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, 2019.

[16] Ashley Varghese, Jayavardhana Gubbi, Akshaya Ramaswamy, and P Balamuralidhar. Changenet: A deep learning architecture for visual change detection. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0. Springer, 2018.

[17] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. *arXiv preprint arXiv:1412.6572*, 2014.

[18] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[19] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A Simple and Accurate Method to Fool Deep Neural Networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582. IEEE, 2016.

[20] Omid Poursaeed, Isay Katsman, Bicheng Gao, and Serge Belongie. Generative Adversarial Perturbations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4422–4431. IEEE, 2018.

[21] Muzammal Naseer, Salman H Khan, Harris Khan, Fahad Shahbaz Khan, and Fatih Porikli. Cross-Domain Transferability of Adversarial Perturbations. *arXiv preprint arXiv:1905.11736*, 2019.

[22] Mathieu Salzmann et al. Learning transferable adversarial perturbations. *Advances in Neural Information Processing Systems*, 34, 2021.

[23] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating Adversarial Examples with Adversarial Networks. *arXiv preprint arXiv:1801.02610*, 2018.

[24] Qilong Zhang, Xiaodan Li, YueFeng Chen, Jingkuan Song, Lianli Gao, Yuan He, and Hui Xue'. Beyond imagenet attack: Towards crafting adversarial examples for black-box domains. In *International Conference on Learning Representations*. International Conference on Learning Representations (ICLR), 2022.

[25] Konda Reddy Mopuri, Utkarsh Ojha, Utsav Garg, and R Venkatesh Babu. Nag: Network for adversary generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 742–751, 2018.

[26] Krishna Kanth Nakka and Mathieu Salzmann. Indirect local attacks for context-aware semantic segmentation networks. In *European Conference on Computer Vision*, pages 611–628. Springer, 2020.

[27] Taesung Park, Alexei A Efros, Richard Zhang, and Jun-Yan Zhu. Contrastive learning for unpaired image-to-image translation. In *European Conference on Computer Vision*, pages 319–345. Springer, 2020.

[28] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738. IEEE, 2020.

[29] Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742. IEEE, 2018.

[30] Alex Andonian, Taesung Park, Bryan Russell, Phillip Isola, Jun-Yan Zhu, and Richard Zhang. Contrastive feature loss for image prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1934–1943. IEEE, 2021.

[31] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778. IEEE, 2016.

[33] 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*, pages 740–755. Springer, 2014.

[34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.

[35] Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Analysis of Classifiers' Robustness to Adversarial Perturbations. *Machine Learning*, pages 481–508, 2018.

[36] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*, 2016.

[37] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436. IEEE, 2015.

[38] Aishan Liu, Xianglong Liu, Jiaxin Fan, Yuqing Ma, Anlan Zhang, Huiyuan Xie, and Dacheng Tao. Perceptual-Sensitive GAN for Generating Adversarial Patches. In *Proceedings of the AAAI conference on artificial intelligence*, pages 1028–1035. AAAI, 2019.

[39] Jiangfan Han, Xiaoyi Dong, Ruimao Zhang, Dongdong Chen, Weiming Zhang, Nenghai Yu, Ping Luo, and Xiaogang Wang. Once a MAN: Towards Multi-Target Attack via Learning Multi-Target Adversarial Network Once. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5158–5167. IEEE, 2019.

[40] Chaowei Xiao, Jun-Yan Zhu, Bo Li, Warren He, Mingyan Liu, and Dawn Song. Spatially transformed adversarial examples. *arXiv preprint arXiv:1801.02612*, 2018.

[41] Shasha Li, Abhishek Aich, Shitong Zhu, Salman Asif, Chengyu Song, Amit Roy-Chowdhury, and Srikanth Krishnamurthy. Adversarial attacks on black box video classifiers: Leveraging the power of geometric transformations. *Advances in Neural Information Processing Systems*, 34, 2021.

[42] Yanbo Fan, Baoyuan Wu, Tuanhui Li, Yong Zhang, Mingyang Li, Zhifeng Li, and Yujiu Yang. Sparse adversarial attack via perturbation factorization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 35–50. Springer, 2020.

[43] 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.

[44] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting Adversarial Attacks with Momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193. IEEE, 2018.

[45] Qingquan Song, Haifeng Jin, Xiao Huang, and Xia Hu. Multi-label adversarial perturbations. In *2018 IEEE International Conference on Data Mining (ICDM)*, pages 1242–1247. IEEE, 2018.

[46] Nan Zhou, Wenjian Luo, Xin Lin, Peilan Xu, and Zhenya Zhang. Generating multi-label adversarial examples by linear programming. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020.

[47] Shu Hu, Lipeng Ke, Xin Wang, and Siwei Lyu. Tkml-ap: Adversarial attacks to top-k multi-label learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7649–7657. IEEE, 2021.

[48] Shaohao Lu, Yuqiao Xian, Ke Yan, Yi Hu, Xing Sun, Xiaowei Guo, Feiyue Huang, and Wei-Shi Zheng. Discriminator-free generative adversarial attack. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 1544–1552. ACM, 2021.

[49] Aaron Van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv e-prints*, pages arXiv–1807, 2018.

[50] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. Transferable adversarial perturbations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–467. Springer, 2018.

[51] 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 *NeurIPS*. NeurIPS, 2020.

[52] Nathan Inkawhich, Wei Wen, Hai Helen Li, and Yiran Chen. Feature space perturbations yield more transferable adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7066–7074. IEEE, 2019.

[53] Yuezun Li, Ming-Ching Chang, Pu Sun, Honggang Qi, Junyu Dong, and Siwei Lyu. Transrpn: Towards the transferable adversarial perturbations using region proposal networks and beyond. *Computer Vision and Image Understanding*, 213:103302, 2021.

[54] Shantanu Godbole and Sunita Sarawagi. Discriminative methods for multi-labeled classification. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 22–30. Springer, 2004.

[55] Mohammad S Sorower. A literature survey on algorithms for multi-label learning. *Oregon State University, Corvallis*, 18:1–25, 2010.

[56] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely Connected Convolutional Networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708. IEEE, 2017.

[57] Karen Simonyan and Andrew Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[58] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[59] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

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

[61] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

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

[63] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013.

[64] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. IEEE, 2009.

[65] Aaron Chen. Coarse-grain models and pre-trained weights. GitHub link, 2022.

[66] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141. IEEE, 2018.

[67] Alibaba-AAIG. Fine-grain models and pre-trained weights. GitHub link, 2022.

[68] PyTorch. Imagenet models and pre-trained weights. PyTorch, 2022.

[69] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929. IEEE, 2016.

[70] Abhishek Aich, Akash Gupta, Rameswar Panda, Rakib Hyder, M. Salman Asif, and Amit K. Roy-Chowdhury. Non-adversarial video synthesis with learned priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[71] Akash Gupta, Abhishek Aich, and Amit K Roy-Chowdhury. Alanet: Adaptive latent attention network for joint video deblurring and interpolation. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 256–264, 2020.