3-26-2021

On Generating Transferable Targeted Perturbations

Muzammal Naseer
Australian National University

Salman Khan
Mohamed bin Zayed University of Artificial Intelligence

Munawar Hayat
Monash University

Fahad Shahbaz Khan
Mohamed bin Zayed University of Artificial Intelligence

Fatih Porikli
Qualcomm

On Generating Transferable Targeted Perturbations

Muzammal Naseer*, Salman Khan†, Munawar Hayat§, Fahad Shahbaz Khan†, Fatih Porikli‡
*Australian National University, Australia, §Monash University, Australia, ‡Qualcomm, USA
†Mohamed bin Zayed University of Artificial Intelligence, UAE
muzammal.naseer@anu.edu.au, {salman.khan,fahad.khan}@mbzuai.ac.ae, munawar.hayat@monash.edu.au
fatih.porikli@gmail.com

Abstract

*While the untargeted black-box transferability of adversarial perturbations has been extensively studied before, changing an unseen model's decisions to a specific 'targeted' class remains a challenging feat. In this paper, we propose a new generative approach for highly transferable targeted perturbations (*TTP*). We note that the existing methods are less suitable for this task due to their reliance on class-boundary information that changes from one model to another, thus reducing transferability. In contrast, our approach matches the perturbed image 'distribution' with that of the target class, leading to high targeted transferability rates. To this end, we propose a new objective function that not only aligns the global distributions of source and target images, but also matches the local neighbourhood structure between the two domains. Based on the proposed objective, we train a generator function that can adaptively synthesize perturbations specific to a given input. Our generative approach is independent of the source or target domain labels, while consistently performs well against state-of-the-art methods on a wide range of attack settings. As an example, we achieve* 32.63% *target transferability from (an adversarially weak) VGG19*$_{BN}$ *to (a strong) WideResNet on ImageNet val. set, which is* $4\times$ *higher than the previous best generative attack and* $16\times$ *better than instance-specific iterative attack. Code is available at:* https://github.com/Muzammal-Naseer/TTP.

1. Introduction

We study the challenging problem of *targeted* transferability of adversarial perturbations. In this case, given an input sample from any source category, the goal of the adversary is to change the decision of an *unknown* model to a *specific* target class (e.g., misclassify any painting image to `Fire truck`, see Fig. 1). This task is significantly more difficult than merely changing the decision to a random target class or any similar class (*e.g.*, changing '`cat`' to '`aeroplane`' is more difficult than altering the decision to '`dog`'). Target transferability can therefore lead to *goal-driven* adversarial perturbations that provide desired control over the attacked model. However, target transferability remains challenging for the current adversarial attacks [26, 24, 4, 44, 15, 14, 13, 21, 42] that transfer adversarial noise in a *black-box* setting, where architecture and training mechanism of the attacked model remain unknown, and the attack is restricted within a certain perturbation budget.

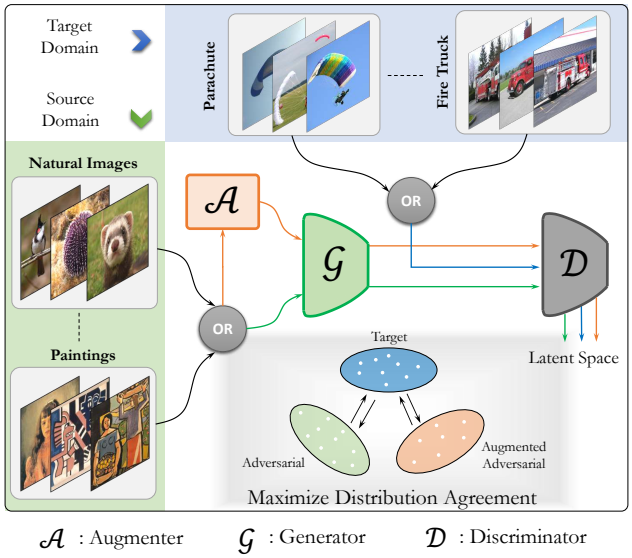


Figure 1: Attack Overview (TTP): Instead of finding perturbations specific to a class-boundary information learned by a model, TTP seeks to match global distribution statistics between the source and the target domains. Specifically, our generator function is trained to maximize agreement between the *perturbed* source distribution, its augmented version and the target distribution in the feature space. Importantly, our attack can function in an unsupervised fashion and does not require source domain to be the same as target (e.g., perturbations can be learned from paintings to transfer on natural images).

We observe that modest performance of existing meth-

ods on targeted transferability is due to their reliance on class-boundary information learned by the model which lacks generalizability. For example, iterative *instance-specific* attacks rely on the classification score information to perturb a given sample, thereby ignoring the global *class-specific* information [26, 4, 44, 14]. Such adversarial directions also vary across different models [24], leading to poor target transferability [24, 4]. On the other hand, although universal and generative perturbations are designed to encode global noise patterns [27, 36, 32], they still exploit the class impressions learned by a neural network which alone are not fully representative of the target distribution, thereby achieving only modest black-box fooling rates [37]. Furthermore, they are dependent on the classification information, necessitating a supervised pretrained model for generator's guidance and therefore cannot directly work with unsupervised features [1, 9]. Another group of techniques exploit intermediate features, but they either find untargeted perturbations by design [29, 38] or are limited in their capacity to transfer targeted perturbations [21, 15, 13, 14].

We introduce a novel generative training framework which maps a given source distribution to a specific target distribution by maximizing the mutual agreement between the two in the latent space of a pretrained discriminator. Our main contributions are:

- ***Generative Targeted Transferability:*** We propose a novel generative approach to learn transferable targeted adversarial perturbations. Our unique training mechanism allows the generator to explore augmented adversarial space during training which enhances the transferability of adversarial examples during inference (Sec. 3.1).
- ***Mutual Distribution Matching:*** Our training approach is based on maximizing the mutual agreement between the given source and the target distribution. Therefore, our method can provide targeted guidance to train the generator without the need of classification boundary information. This allows an attacker to learn targeted generative perturbations from the unsupervised features [1, 9] and eliminate the cost of labelled data (Sec. 3.2).
- ***Neighbourhood Similarity Matching:*** Alongside global distribution matching, we introduce batch-wise neighbourhood similarity matching objective between adversarial and target class samples to maximize the local alignment between the two distributions (Sec. 3.3).

Our extensive experiments on various ImageNet splits and CNN architectures show state-of-the-art targeted transferability against naturally and adversarially trained models, stylized models and input-processing based defenses. The results demonstrate our benefit compared to recent targeted instance-specific as well as other generative methods. Further, our attack demonstrates rapid convergence.

2. Related Work

Iterative Instance-Specific Perturbations: After Szegedy *et al*. [41] highlighted the vulnerability of neural networks, many adversarial attacks have been introduced to study if the adversarial examples are transferable from one model to another, when a target model is unknown. Among these, iterative instance-specific attacks [4, 44, 5] perturb a given sample by iteratively using gradient information. Target transferability of such attacks is very poor [4, 24] (as shown in Sec. 4). Other attacks also use feature space either by maximizing the feature difference [45, 11, 22] or applying attention [43] or avoiding non-linearity while back-propagating gradients [8] or exploiting skip-connections [42]. However, these attacks are mainly designed to enhance non-targeted transferability which is an easier problem. Recently, different instance-specific (transferable) targeted attacks have been proposed including [21] which introduces a triplet loss to push adversarial examples towards the target label while increasing their distance from the original label. Inkawhich *et al*. [13, 14] proposed to exploit feature space [15] along with the classifier information [14] to generate target adversaries that are shown to transfer relatively better than other instance-specific attacks. These attacks [13, 14] have the following limitations. **a)** They need access to a *labeled* dataset *e.g*., ImageNet [39] in order to train one-vs-all binary classifiers for attacked target classes. **b)** They need to identify best performing single layer [13] or a combination of layers [14] which adds further complexity to attack optimization. **c)** Finally, the attack performance degrades significantly with quality of features, *e.g*., it struggles to transfer target perturbations from VGG models [13].
Universal Perturbation: In contrast to instance-specific perturbations, [27] learns a single universal noise pattern which is representative of the entire data distribution and can fool a model on majority of samples. Li *et al*. [23] introduce gradient transformation module to find smooth universal patterns while [29] shows that such patterns can be found without any training data. Although universal perturbations [27, 28, 29, 23] based attacks are efficient (the attacker just needs to add the noise to any given sample at inference), they are limited in their capacity to yield transferable adversaries which can generalize across different data distributions and models [36, 32].
Generative Perturbations: Generative adversarial perturbations perform better than directly optimizing universal noise [38, 36, 32]. Poursaeed *et al*. [36] proposed the first generative approach to adapt perturbations to an input sample. Naseer *et al*. [32] improved this framework with relativistic training objective which also allows cross-domain transferability. Our method belongs to the generative category and can adapt to an input sample with a single forward pass. Unlike [38, 36, 32], we seek to fool the model by matching *distributions* of source and targets with distribu-

tion matching and neighbourhood similarity criteria. Our proposed framework does not require labeled source or target data and can extract target perturbations from a discriminator model trained in an unsupervised manner while previous generative methods are dependent on class-boundary information learned by the model. Further, our method converges faster (Sec. 4) and provides improved targeted transferability owing to its novel loss and training mechanism.

3. Generating Targeted Adversaries

Our goal is to craft adversarial perturbations δ that can fool a model to misclassify any given input to a specific target class t. We assume access to source and target domain data represented by P and Q, from which the source and target class samples are obtained i.e., $\boldsymbol{x}_s \sim P$, $\boldsymbol{x}_t \sim Q$. The source and target domains are likely to be non-aligned i.e., $P \neq Q$, making it challenging to achieve targeted transferability of adversarial perturbations. We also consider a perturbed source data P' that comprises of adversarially manipulated samples $\boldsymbol{x}'_s \sim P'$ where $\boldsymbol{x}'_s = \boldsymbol{x}_s + \delta$. $\boldsymbol{x}_s$, $\boldsymbol{x}'_s$ and $\boldsymbol{x}_t$ represent source, adversarial and target domain samples while $\mathcal{D}_{\boldsymbol{\psi}}(\boldsymbol{x}_s)$, $\mathcal{D}_{\boldsymbol{\psi}}(\boldsymbol{x}'_s)$ and $\mathcal{D}_{\boldsymbol{\psi}}(\boldsymbol{x}_t)$ are their corresponding latent distributions.

3.1. Generative Model

We propose a generative approach to perturb the source domain samples $\boldsymbol{x}_s$ to a specified target class. The framework (see Fig. 2) consists of a generator $\mathcal{G}_{\boldsymbol{\theta}}$ and a discriminator $\mathcal{D}_{\boldsymbol{\psi}}$ parameterized by $\boldsymbol{\theta}$ and $\boldsymbol{\psi}$, respectively. The generator function $\mathcal{G}_{\boldsymbol{\theta}}$ learns a mapping from the source images to the target category such that the input images are minimally changed i.e., adversarial noise δ is strictly constrained under a norm distance $l_\infty \leq \epsilon$. This is ensured by projecting the unbounded adversaries from $\mathcal{G}_{\boldsymbol{\theta}}$ within fixed norm distance of $\boldsymbol{x}_s$ using a differentiable clipping operation,

$$\boldsymbol{x}'_s = \text{clip}\big(\min(\boldsymbol{x}_s + \epsilon, \max(\mathcal{W} * \mathcal{G}_{\boldsymbol{\theta}}(\boldsymbol{x}_s), \boldsymbol{x}_s - \epsilon))\big), \quad (1)$$

where, $\mathcal{W}$ is a smoothing operator with fixed weights that reduces high frequencies without violating the l_∞ distance constraint. The smooth projection in Eq. 1 (denoted by $\mathcal{P}$ in Fig. 2) not only tightly bounds generator's output within l_∞ norm but also encourages avoiding redundant high frequencies [35] during the optimization process. This allows the generator to converge to a more meaningful solution.

The existing generative designs for adversarial attacks [36, 32] leverage the decision space of the discriminator to craft perturbations. In such cases, the *class-boundary* information learned by the discriminator is used to fool DNN models (*e.g.* for ImageNet, discriminator is pretrained on 1k classes). This dependence is problematic since an attacker must have access to a discriminator trained on large-scale labeled dataset [3]. Attacker then tries to learn target class impressions using either cross-entropy (CE) [36] or relativistic CE [32]. Thus, the generated perturbations are directly dependent on the quality of the discriminator's classification space. Furthermore, the generated adversaries are dependent on the input instance-specific features and do not model the global properties of the target distribution, resulting in only limited transferability.

Algorithm 1 Generating TTP

Require: Source data $\mathcal{X}_s$, Target data $\mathcal{X}_t$, pretrained discriminator $\mathcal{D}_{\boldsymbol{\psi}}$, perturbation budget ϵ, loss criteria $\mathcal{L}_{\mathcal{G}}$.
Ensure: Randomly initialize the generator, $\mathcal{G}_{\boldsymbol{\theta}}$

repeat
  Randomly sample mini-batches $\boldsymbol{x}_s \sim \mathcal{X}_s$ and $\boldsymbol{x}_t \sim \mathcal{X}_t$
  Create augmented copy of the source mini-batch $\tilde{\boldsymbol{x}}_s$.
  Forward-pass $\boldsymbol{x}_s$ and $\tilde{\boldsymbol{x}}_s$ through the generator and generate unbounded adversaries; $\boldsymbol{x}'_s$, $\tilde{\boldsymbol{x}}'_s$.
  Bound the adversaries using Eq. 1 such that:

$$\|\boldsymbol{x}'_s - \boldsymbol{x}_s\|_\infty \leq \epsilon \;\text{ and }\; \|\tilde{\boldsymbol{x}}'_s - \tilde{\boldsymbol{x}}_s\|_\infty \leq \epsilon$$

  Forward pass $\boldsymbol{x}'_s$, $\tilde{\boldsymbol{x}}'_s$ and $\boldsymbol{x}_t$ through $\mathcal{D}_{\boldsymbol{\psi}}$.
  Compute the matching losses; $\mathcal{L}$, $\mathcal{L}^{aug}$ and $\mathcal{L}^{sim}$ using Eq. 3, 4 and 8, respectively.
  Compute the generator loss given in Eq. 9.
  Backward pass and update $\mathcal{G}_{\boldsymbol{\theta}}$
until $\mathcal{G}_{\boldsymbol{\theta}}$ converges.

To address above limitations, our generative design models the target distribution Q and pushes the perturbed source distribution P' closer to Q using the latent space of $\mathcal{D}_{\boldsymbol{\psi}}$,

$$\| \boldsymbol{\delta} \|_\infty \leq \epsilon, \quad s.t., \quad \mathcal{D}_{\boldsymbol{\psi}}(\boldsymbol{x}'_s) \approx \mathcal{D}_{\boldsymbol{\psi}}(\boldsymbol{x}_t). \quad (2)$$

This global objective provides two crucial benefits. *First,* reducing mismatch between perturbed and target distributions provides an improved guidance to the generator. The resulting perturbations well align the input samples with the target distribution, leading to transferable adversaries. *Second,* the distributions alignment task makes us independent of the $\mathcal{D}_{\boldsymbol{\psi}}$'s classification information. In turn, our approach can function equally well with a discriminator trained in a self-supervised manner on unlabelled data [1, 9]. In our case, we simply align the feature distributions from $\mathcal{D}_{\boldsymbol{\psi}}$ to match P' and Q. Thus, for a given sample $\boldsymbol{x}$, n-dimensional features are obtained i.e., $\mathcal{D}_{\boldsymbol{\psi}}(\boldsymbol{x}) \in \mathbb{R}^n$. If $\mathcal{D}_{\boldsymbol{\psi}}$ is trained in a supervised manner on ImageNet then $n = 1000$, and if $\mathcal{D}_{\boldsymbol{\psi}}$ is trained in an unsupervised fashion then n is equal to the output feature dimension.

3.2. Distribution Matching

We measure the mutual agreement between P' and Q using Kullback Leibler (KL) divergence defined on discrimi-

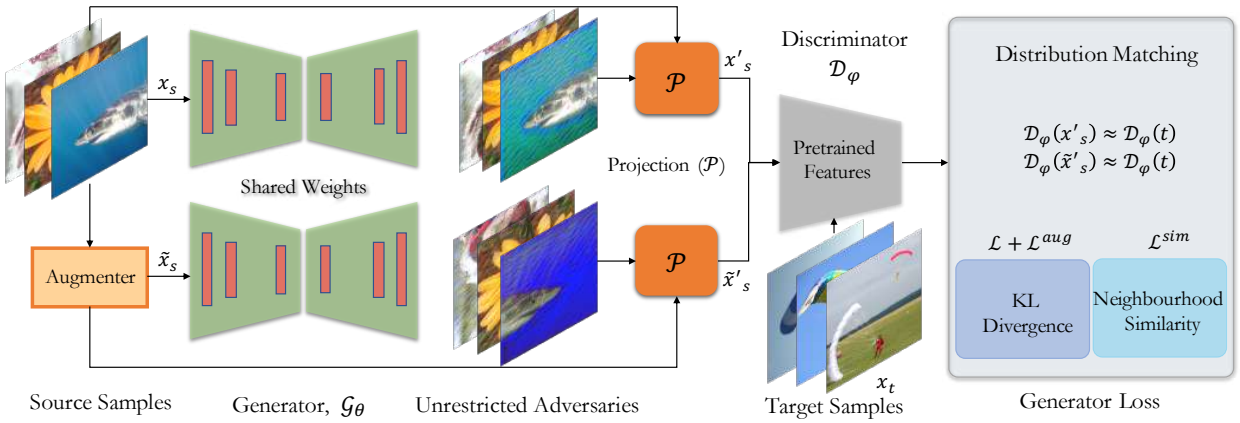


Figure 2: *Targeted Transferable Perturbations:* During training, TTP matches adversarial and augmented adversarial samples to a target domain within discriminator's latent space for improved transferability. The adversarial samples corresponding to original and augmented images are bounded (via projection) around their source samples to explore adversarial space around natural as well as augmented samples.

nator features $\mathcal{D}_\psi(\boldsymbol{x}'_s)$ and $\mathcal{D}_\psi(\boldsymbol{x}_t)$,

$$D_{KL}(P'\|Q) = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{n}\sigma(\mathcal{D}_\psi(\boldsymbol{x}'^{,i}_s))_j \log \frac{\sigma(\mathcal{D}_\psi(\boldsymbol{x}'^{,i}_s))_j}{\sigma(\mathcal{D}_\psi(\boldsymbol{x}^i_t))_j},$$

where N represents the number of samples, n is the discriminator's output dimension, and σ denotes the softmax operation. In simple terms, KL divergence measures the difference between two distributions in terms of the average surprise in experiencing $\boldsymbol{x}_t$ when we expected to see $\boldsymbol{x}'_s$. Since KL divergence is asymmetric *i.e.* $D_{KL}(P'\|Q) \neq D_{KL}(Q\|P')$, and not a valid distance measure, we define our loss function for distribution matching [20] as follows:

$$\mathcal{L} = D_{KL}(P'\|Q) + D_{KL}(Q\|P'). \tag{3}$$

As a regularization measure, we add augmented versions of the source domain samples during distribution matching. This enables the generator to focus specifically on adding target class-specific patterns that are robust to input transformations. To this end, we randomly apply rotation, crop resize, horizontal flip, color jittering or gray scale transformation to create augmented samples $\tilde{\boldsymbol{x}}_s$ from the original $\boldsymbol{x}_s$. The $\tilde{\boldsymbol{x}}_s \sim \tilde{P}$ are passed through the $\mathcal{G}_\theta$ and the perturbed augmented samples $\tilde{\boldsymbol{x}}'_s \sim \tilde{P}'$ are projected using Eq. 1 to stay close to the augmented samples i.e., $\|\tilde{\boldsymbol{x}}'_s - \tilde{\boldsymbol{x}}_s\|_\infty \leq \epsilon$. No augmentation is applied to the target domain samples. We then pass $\tilde{\boldsymbol{x}}'_s$ through the discriminator and compute the mutual agreement between $\mathcal{D}_\psi(\tilde{\boldsymbol{x}}'_s)$ and $\mathcal{D}_\psi(\boldsymbol{x}_t)$ as follows:

$$\mathcal{L}^{aug} = D_{KL}(\tilde{P}'\|Q) + D_{KL}(Q\|\tilde{P}'). \tag{4}$$

The impact of data augmentations and their effectiveness for our proposed targeted attack is studied in Sec. 4.

3.3. Neighbourhood Similarity Matching

The above objective promotes alignment between the distributions but does not consider the local structure e.g., the relationship between a sample and its augmented versions. For a faithful alignment between perturbed source samples and the target class samples, we propose to also match the *neighbourhood similarity distributions* between the two domains. Specifically, consider a batch of target domain samples $\{\boldsymbol{x}^i_t\}_{i=1}^N$ and a batch of perturbed source domain samples $\{\boldsymbol{x}'^i_s\}_{i=1}^N$. For the case of $\boldsymbol{x}'_s$, in a given training batch, we compute a similarity matrix $\mathcal{S}^s$ whose elements encode the cosine similarity between the original sample and its augmented version $\tilde{\boldsymbol{x}}'_s$, i.e.,

$$\mathcal{S}^s_{i,j} = \frac{\mathcal{D}_\psi(\boldsymbol{x}'^{,i}_s)\cdot\mathcal{D}_\psi(\tilde{\boldsymbol{x}}'^{,j}_s)}{\|\mathcal{D}_\psi(\boldsymbol{x}'^{,i}_s)\|\|\mathcal{D}_\psi(\tilde{\boldsymbol{x}}'^{,j}_s)\|}. \tag{5}$$

In contrast, for the case of $\boldsymbol{x}_t$, we compute similarity between only the original target samples (no augmentations) as we need to model the local neighbourhood connectivity in the target domain. This choice is impractical for the source domain case where many categories co-exist, while for the target distribution, we assume a single category. Thus the target similarity matrix $\mathcal{S}^t$ is computed as,

$$\mathcal{S}^t_{i,j} = \frac{\mathcal{D}_\psi(\boldsymbol{x}^i_t)\cdot\mathcal{D}_\psi(\boldsymbol{x}^j_t)}{\|\mathcal{D}_\psi(\boldsymbol{x}^i_t)\|\|\mathcal{D}_\psi(\boldsymbol{x}^j_t)\|}. \tag{6}$$

The resulting similarity matrices are normalized along the row dimension with softmax to obtain probability estimates,

$$\bar{\mathcal{S}}_{i,j} = \frac{\exp(\mathcal{S}_{i,j})}{\sum_k \exp(\mathcal{S}_{i,k})}, \text{ where, } \mathcal{S} \in \{\mathcal{S}^s, \mathcal{S}^t\}. \tag{7}$$

Here, each term shows the probability with which the two sample pairs are related to each other. Given $\bar{\mathcal{S}}^s$ and $\bar{\mathcal{S}}^t$, we

compute the KL divergence to enforce a loss term that seeks to match the local neighbourhood patterns between source and target domains,

$$\mathcal{L}^{sim} = \sum_{i,j} \bar{\mathcal{S}}^t_{i,j} \log \frac{\bar{\mathcal{S}}^t_{i,j}}{\bar{\mathcal{S}}^s_{i,j}} + \sum_{i,j} \bar{\mathcal{S}}^s_{i,j} \log \frac{\bar{\mathcal{S}}^s_{i,j}}{\bar{\mathcal{S}}^t_{i,j}}. \quad (8)$$

3.4. Overall loss function

Finally, the generator parameters are updated by minimizing the following loss (Algorithm 1):

$$\mathcal{L}_{\mathcal{G}} = \mathcal{L} + \mathcal{L}^{aug} + \mathcal{L}^{sim}. \quad (9)$$

This loss encourages the generator to perturb source samples that not only match the global characteristics of the target distribution ($\mathcal{L} + \mathcal{L}^{aug}$), but also the local information based on neighbourhood connectivity ($\mathcal{L}^{sim}$).

4. Experiments

Our generator $\mathcal{G}_{\boldsymbol{\theta}}$ is based on ResNet architecture [17], and outputs an adversarial sample with the same size as of input (Fig. 3). This generator architecture is the same as in the baseline generative attacks [36, 32]. Our discriminator $\mathcal{D}_{\boldsymbol{\psi}}$ is pre-trained in a supervised or self-supervised manner. For training $\mathcal{G}_{\boldsymbol{\theta}}$, we freeze $\mathcal{D}_{\boldsymbol{\psi}}$. We use Adam optimizer [19] with a learning rate of 10^{-4} ($\beta_1 = .5, \beta_2 = .999$) for 20 epochs. For source domain data, we use 50k random images from ImageNet train set. Our method is not sensitive to the choice of source samples since it can learn transferable perturbations even from other domains *e.g.* Paintings. Similar to other generative methods [36, 32], we fix source data. For target domain data, we use 1300 images for each target collected from ImageNet training set (without their original labels). We used default settings or implementations as provided by the authors of baseline attacks. Similarly, we used open-sourced (pretrained) stylized [7], adversarial [40] and purifier (NRP) [30] models to evaluate robustness.

4.1. Evaluation Settings

We perform inference on ImageNet validation set (50k samples). *No augmentations are applied at inference time*. The perturbation budget is tightly bounded and clearly mentioned in each experiment following the standard practices $l_\infty \leq 16$ [4, 15, 14] and $l_\infty \leq 32$ [32, 23]. We perturb all the ImageNet val. samples (except the target samples) to the pre-defined target class. We repeat this process for all the given targets and report Top-1 (%) accuracy averaged across all targets. We compare our method under two main settings (10-Targets and 100-Targets), as described below.

10-Targets: We further consider two settings. **(a)** *10-Targets (subset-source)* which is consistent with [13] and has a subset of source classes at inference. **(b)** *10-Targets (all-source)* which is a more challenging large-scale setting as source images can come from all the ImageNet classes except the target class. For consistency and direct comparison, the ten target classes are same as in [13].

−10-Targets (subset-source): Following [13], for each target class, 450 source samples belonging to remaining 9 classes (except target class) become inputs to $\mathcal{G}_{\boldsymbol{\theta}}$ to be transferred to the selected target.

−10-Targets (all-source): For each target class, samples of all 999 source classes (except the target class) in ImageNet val. set are considered i.e., for each target class, 49,950 samples of 999 classes become inputs to $\mathcal{G}_{\boldsymbol{\theta}}$.

100-Targets (all-source): We divide ImageNet 1k classes into 100 mutually exclusive sets. Each set contains 10 classes. We randomly sample 1 target from each set to create 100 targets (see Appendix E for more details). Generators are trained against these targets and evaluated on ImageNet val. set in 100-Targets (all-source) setting with the same protocol as described for 10-Targets (all-source).

Src.	Attack	Naturally Trained (IN) Models				
		$VGG19_{BN}$	Dense121	ResNet50	ResNet152	WRN-50-2
$VGG19_{BN}$	PGD [26]	95.67*	0.31	0.30	0.20	0.25
	MIM [4]	99.91*	0.92	0.68	0.36	0.47
	DIM [44]	99.38*	3.10	2.08	1.02	1.29
	DIM-TI [5]	89.71*	1.08	0.66	0.42	0.45
	Po-TRIP [21]	99.40*	4.61	3.21	1.78	2.01
	GAP [36]	98.23*	16.19	15.83	5.89	7.78
	CDA [32]	98.30*	16.26	16.22	5.73	8.35
	Ours-P	97.38*	45.53	42.90	26.72	31.00
	Ours	98.54*	**45.77**	**45.87**	**27.18**	**32.63**
V_{ens}	**Ours**	97.34*	71.41	71.68	50.78	48.03
Dense121	PGD [26]	1.28	97.40*	1.78	1.01	1.37
	MIM [4]	1.85	99.90*	2.71	1.68	1.88
	DIM [44]	7.31	98.81*	9.06	5.78	6.29
	DIM-TI [5]	0.91	88.59*	1.18	0.77	0.86
	Po-TRIP [21]	8.10	99.00*	11.21	7.83	8.50
	GAP [36]	39.01	97.30*	47.85	39.25	34.79
	CDA [32]	42.77	97.22*	54.28	44.11	46.01
	Ours-P	57.91	97.41*	**71.35**	55.57	53.45
	Ours	**58.90**	97.61*	68.72	**57.11**	**56.80**
D_{ens}	**Ours**	76.96	96.25*	88.81	83.48	81.85
ResNet50	PGD [26]	0.92	1.38	93.74*	1.86	1.89
	MIM [4]	1.58	3.37	98.76*	3.39	3.17
	DIM [44]	9.14	15.47	99.01*	12.45	12.61
	DIM-TI [5]	0.79	2.12	88.91*	1.47	1.45
	Po-TRIP [21]	12.01	19.43	99.22*	14.41	15.10
	GAP [36]	58.47	71.72	96.81*	64.89	61.82
	CDA [32]	64.58	73.57	96.30*	70.30	69.27
	Ours-P	73.09	**84.76**	96.63*	76.27	75.92
	Ours	**78.15**	81.64	97.02*	**80.56**	**78.25**
R_{ens}	**Ours**	90.43	94.39	96.67*	95.48*	92.63

Table 1: ***Target Transferability:*** {10-Targets (all-source)} Top-1 target accuracy (%) averaged across 10 targets with 49.95K ImageNet val. samples. Perturbation budget: $l_\infty \leq 16$. Our method outperforms previous instance-specific as well as generative approaches by a large margin. '*' indicates white-box attack. **Ours-P** represents TTP trained on Paintings.

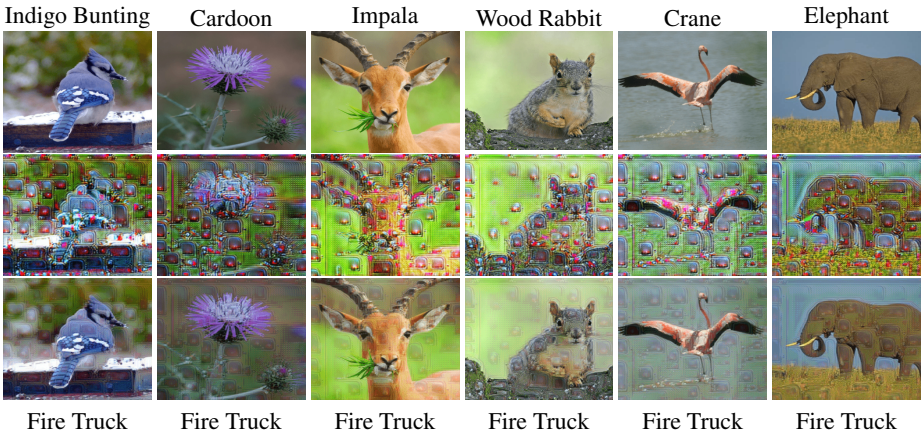


Figure 3: Targeted adversaries produced by a TTP generator learned to maximize the agreement with 'Fire Truck' distribution against Dense121 ImageNet model. 1st and 2nd rows show clean images and unrestricted outputs of the adversarial generator, respectively. 3rd row shows adversaries after valid projection. See Appendix F for more qualitative examples including comparisons between targeted patterns learned by TTP from different source models of a certain family of networks.

Src.	Attack	$VGG19_{BN}$	Dense121	ResNet50
$VGG19_{BN}$	AA [15]	–	0.8	0.6
	FDA-fd [13]	–	3.0	2.1
	FDA^N [14]	–	6.0	5.4
	CDA [32]	–	17.82	17.09
	Ours-P	–	**48.56**	44.47
	Ours	–	48.29	**47.07**
Dense121	AA [15]	0.0	–	0.0
	FDA-fd [13]	34.0	–	34.0
	FDA^N [14]	42.0	–	48.3
	CDA [32]	44.84	–	53.73
	Ours-P	59.81		**71.32**
	Ours	**61.75**	–	69.60
ResNet50	AA [15]	1.1	2.0	–
	FDA-fd [13]	16.0	21.0	–
	FDA^N [14]	32.1	48.3	–
	CDA [32]	68.55	75.68	–
	Ours-P	75.18	**85.71**	
	Ours	**79.04**	84.42	–

Table 2: ***Target Transferability:***{10-Targets (sub-source)} Top-1 accuracy (%) across 10 targets. Our method shows significant improvements in trasfering target perturbations compared to generative as well as feature based instance-specific method [13, 14]. Perturbation budget:$l_\infty \leq 16$. Only black-box attack results are shown. **Ours-P** represents TTP trained on Paintings.

4.2. Attack Protocols and Results

We evaluate black-box target transferability in the following scenarios. **(a)** *Unknown Target Model:* Attacker has access to a pretrained discriminator trained on labeled data but has no knowledge about the architecture of the target model. **(b)** *Unknown Decision Space:* Attacker has access to the pre-trained discriminator trained on unlabeled data in an unsupervised manner but does not know about the architecture and the class-boundary information learned by the target model. **(c)** *Unknown Defense:* Attacker is unaware of the type of defense deployed at the target model, or if any defense is applied at all, e.g., the defense can be an input processing approach or a robust training mechanism such as adversarial training.

4.2.1 Unknown Target Model

Natural Training: We evaluate naturally trained ImageNet models and show strong empirical results in Tables 1, 2 & 3 demonstrating that generative methods are far superior than sample-specific targeted attacks based on boundary information [21, 4, 44] or feature exploitation [15, 13, 14]. Our approach has significantly higher target transferability rates than previous generative methods [32, 36]. To highlight an example from Table 2, our method achieves 47.07% transferability from $VGG19_{BN}$ to ResNet50 which is 175% and 771% better than the previous best generative [32] and sample-specific [14] target attacks, respectively.

Ensemble Effect: We also train generators with our algorithm on the ensembles of same-family discriminators. Specifically, we define the following ensembles: V_{ens}:VGG{11,13,16,19}$_{BN}$,R_{ens}:ResNet{18,50,101,152}, and D_{ens}:DenseNet{121,161,169,201}.

The purpose of such ensembles is to understand if the combination of weak individual models from the same family can provide strong learning for the target distributions. From Table 1, we observe that modeling target distribution from an ensemble provides significantly better tranferability than any individual discriminator (see Appendix A for more analysis). This signifies that an attacker can use multiple variants of the same network to boost the attack.

Target Transferability and Model Disparity: We note that within a specific family, transferring targeted perturbations from a smaller model to a larger one (*e.g.* ResNet18 $\rightarrow$ ResNet152 or $VGG11_{BN} \rightarrow$ VGG19) is difficult as we increase the size discrepancy. Interestingly, this trend remains the same even from larger to smaller models i.e., the attack strength will increase with the disparity between models rather than only depending upon the strength of target model. For example, target transferability ResNet152 $\rightarrow$ ResNet50 is higher than ResNet152 $\rightarrow$ ResNet18 even though ResNet18 is weaker than ResNet50 (Fig. 4). Similar behaviour can be observed within cross-family models *i.e.*,

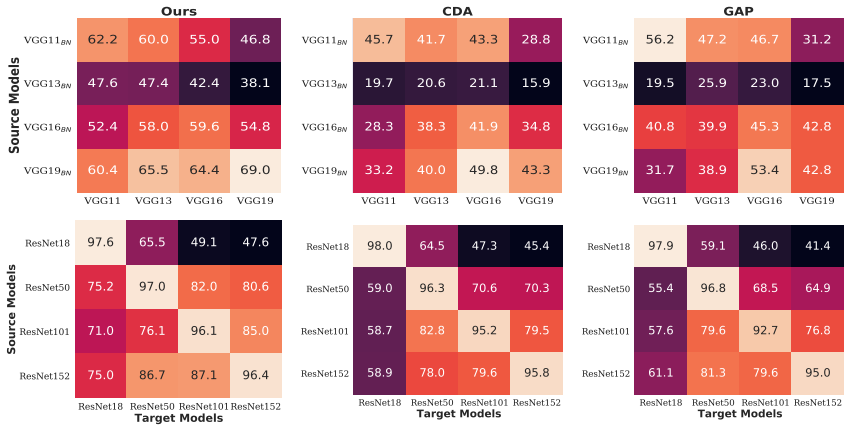


Figure 4: ***Within Family Target Transferability:*** {10-Targets (all source) settings} These results indicate that our approach boosts target transferability within different models of the same family with or without batch-norm and favorably beats the previous generative approaches (GAP [36], CDA [32]) by a large margin. Each value is averaged across 10 targets (Sec. 4) with 49.95k ImageNet val. samples for each target. Perturbation budget is set to $l_\infty = 16$.

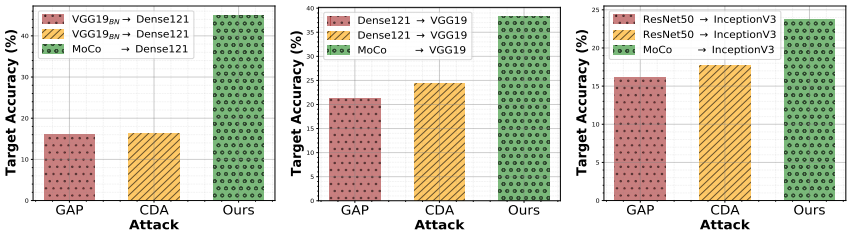


Figure 5: ***Target Transferability of Unsupervised Features:*** {10-Targets (all-source) settings}. Our approach when applied to unsupervised features, MoCo [9], surpasses GAP [36] and CDA [32] that are dependent on classification layer by design. Perturbation budget is $l_\infty = 16$.

target transferability from ResNet50 to Dense121 and vice versa is higher than VGG19_{BN} as both models share skip connections (Table 1). See Appendix B for vulnerability of models with and without batch-norm [16].

4.2.2 Unknown Decision Space

Here we investigate the question, "*Can unsupervised features provide targeted adversarial perturbations?*" A unique property of our proposed approach is that it can be applied to feature space without any class boundary information to achieve target adversarial direction. This allows an attacker to benefit from recently proposed unsupervised feature learning methods [9, 1]. Rather than using a discriminator trained on large scale labelled data, attack can be learned and launched from features of a discriminator trained purely in an unsupervised fashion. Therefore, our attack can eliminate the cost of label annotations. Results in Fig. 5 demonstrate that our method learned from unsupervised features, MoCo [9], not only provides target transferability but surpasses the previous generative methods which are dependent on the discriminator trained on labelled data.

4.2.3 Unknown Defense Mechanisms

Input Processing as a Defense: We evaluate robustness of different input processing based adversarial defense methods in Fig. 6. We consider the following four representative defenses: a) JPEG with compression quality set to 50% [2], b) DNN-Oriented JPEG compression [25], c) Median Blur with window size set to 5×5 [31], and d) Neural representation purifier (NRP) [30] which is a state-of-the-art defense. Generators are trained against naturally trained ResNet50 and target perturbations are then transferred to VGG19_{BN} and Dense121 which are protected by the input processing defenses. We observe (Fig. 6) that JPEG is the least effec-

Attack	VGG19_{BN}	Dense121	ResNet-152	WRN-50-2	SIN [7]
GAP [36]	47.87	58.10	54.72	49.65	7.1
CDA [32]	53.41	60.34	57.67	51.23	7.6
Ours	**69.55**	**77.48**	**75.74**	**74.61**	**31.0**

Table 3: ***Target Transferability:*** {100-Targets (all-source)} Top-1 target accuracy (%) averaged across 100 targets with 49.95K ImageNet val. samples per target . Generators are trained against ResNet50. Perturbation budget is $l_\infty \leq 16$.

ϵ	Attack	Augmix	Stylized [7]		Adversarial [40]			
		[10]	SIN-IN	SIN	l_∞		l_2	
					ϵ=.5	ϵ=1	ϵ=.1	ϵ=.5
16	GAP [36]	51.57	76.92	12.96	1.88	0.34	23.41	0.92
	CDA [32]	59.79	75.93	9.21	2.10	0.39	23.89	1.18
	Ours	**73.09**	**87.40**	**30.17**	**4.63**	**0.56**	**45.40**	**1.99**
	Ours_{ens}	88.79	92.96	57.75	14.23	1.24	74.95	7.62
32	GAP [36]	54.86	81.15	28.07	26.32	6.36	59.04	16.53
	CDA [32]	63.18	76.81	19.65	27.60	6.74	57.54	16.07
	Ours	**78.66**	**91.27**	**41.52**	**46.82**	**16.35**	**75.97**	**30.94**
	Ours_{ens}	89.96	94.15	70.70	70.22	34.21	90.42	58.25

Table 4: ***Target Transferability:*** {10-Targets (all source) settings} Top-1 (%) target accuracy. Generators are trained against naturally trained ResNet50 or ResNet ensemble. Perturbation are then transferred to ResNet50 trained using different methods including Augmix [10], Stylized [7] or adversarial [40].

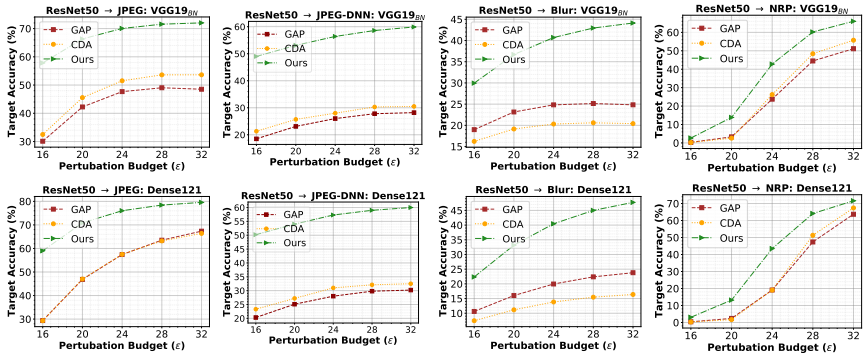


Figure 6: ***Target Transferability against Input Processing Defenses:*** {10-Targets (all-source) settings} Input processing including NRP [30] are broken under targeted black-box attacks. Our method outperforms GAP [36] and CDA [32] on all the considered defenses including JPEG, JPEG-DNN [25], Median Blur and NRP [30]. Each point is an averaged across 10 targets (Sec. 4) with 49.95k ImageNet val. samples for each target. Generators are trained against ResNet50.

tive method against target attacks while JEPG-DNN [25] performs relatively better than JPEG. Compared to JPEG, JPEG-DNN and Median blur, NRP shows better resistance to target attacks at $l_\infty \leq 16$ but quickly breaks as perturbation is increased. Median blur shows more resistance than JPEG, JPEG-DNN and NRP at higher perturbation rates ($l_\infty \leq 32$)[1]. Success rate of our method is much better than previous generative attacks [36, 32] even when the target model and the input processing remain unknown (Fig. 6).

Robust Training Mechanism: Here we study the transferability of our approach against various robust training methods (augmented vs. stylized vs. adversarial) based defense strategies. Augmentation based training can make the model robust to natural corruptions [10] while training on stylized ImageNet [7] improves shape bias and training on adversarial examples can improve robustness against adversarial attacks at the cost of computation, clean accuracy, and generalization to global changes [6]. We evaluate the vulnerability of these training methods in Table 4. Generators are trained against naturally trained ResNet50 or ResNet ensemble and adversarial perturbations are then transferred to ResNet50 trained using Augmix [10], Stylized ImageNet (SIN) [7], mixture of Stylized and natural ImageNet (SIN-IN) and adversarial examples [40]. Target transferability can easily be achieved against models trained on mixture (SIN-IN), however, the model trained on stylized images (SIN) shows higher resistance but remains vulnerable as our target attack (ensemble) achieves ≈71% success at perturbation of $l_\infty = 32$ (Table 4). Adversarially trained models using Madry's method [26] are more robust to target attacks.

4.3. Ablative Analysis

In order to understand the effect of each component of our approach, we present an ablative study in Fig. 7. Target perturbations are transferred from ResNet50 to VGG16 (SIN) trained on stylized ImageNet which is a much harder task than transferring to naturally trained VGG16. We observe that training TTP on only distribution matching loss (Eq. 3) increases the transferability by more than 100% in comparison to GAP [36] (with cross-entropy) or CDA [32] (with relativistic cross-entropy). Adding smoothing operator $\mathcal{W}$ enhances the efficiency of TTP. $\mathcal{W}$ is a differentiable Gaussian kernel with size 3×3. We then noticed a significant jump in transferability when augmentations are introduced and TTP is trained using both distribution matching losses (Eq. 3 & 4) which is further complemented by neighbor similarity loss (Eq. 8). Our generator trained for only one epoch outperforms GAP and CDA trained for 20 epochs (Fig. 7) which highlights our rapid convergence rate.

[1]Blur defense causes large drop in clean accuracy (see Appendix D).

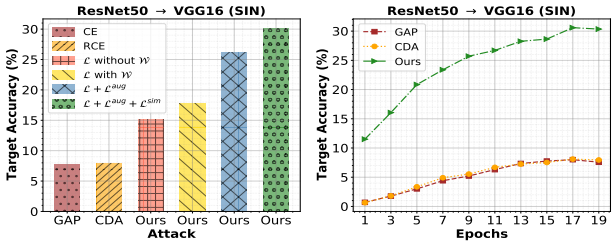


Figure 7: ***Ablation:*** We dissect effect of each component of our method including novel losses, augmentation, smooth projection and epochs. Results are presented with 10-Target (all source) settings. Perturbation budget is set to $l_\infty = 16$.

5. Conclusion

We proposed a new generative approach that can learn to model transferable target-specific perturbations. Given an image from any source class, our approach can synthesize perturbations that lead to its misclassification on a variety of black-box target models. The core of our approach is an instance-adaptive generator function that is learned using a novel loss formulation. Our loss focuses on matching the distribution-level statistics of perturbed source and target samples. By its design, our approach can work with both supervised and unsupervised representations. We demonstrate impressive transferability rates across a range of attack settings compared to state-of-the-art. Our results advocate for the use of global loss functions defined over distributions to craft highly transferable adversarial patterns. In future work, we plan to extend proposed method to other model families such as vision transformers [34, 33, 18].

References

[1] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 2, 3, 7

[2] Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E Kounavis, and Duen Horng Chau. Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. 2018. 7

[3] 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. 3

[4] 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*, 2018. 1, 2, 5, 6, 13

[5] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019. 2, 5

[6] Nic Ford, Justin Gilmer, Nicolas Carlini, and Dogus Cubuk. Adversarial examples are a natural consequence of test error in noise. *arXiv preprint arXiv:1901.10513*, 2019. 8

[7] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. 5, 7, 8, 11, 12, 13

[8] Yiwen Guo, Qizhang Li, and Hao Chen. Backpropagating linearly improves transferability of adversarial examples. *arXiv preprint arXiv:2012.03528*, 2020. 2, 11, 13

[9] 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, 2020. 2, 3, 7

[10] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020. 7, 8, 13

[11] Qian Huang, Isay Katsman, Horace He, Zeqi Gu, Serge Belongie, and Ser-Nam Lim. Enhancing adversarial example transferability with an intermediate level attack. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4733–4742, 2019. 2

[12] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 11

[13] Nathan Inkawhich, Kevin Liang, Lawrence Carin, and Yiran Chen. Transferable perturbations of deep feature distributions. In *International Conference on Learning Representations*, 2020. 1, 2, 5, 6, 13

[14] 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. *arXiv preprint arXiv:2004.14861*, 2020. 1, 2, 5, 6, 13

[15] 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, 2019. 1, 2, 5, 6

[16] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 7, 11, 12

[17] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 5

[18] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *arXiv preprint arXiv:2101.01169*, 2021. 8

[19] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[20] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951. 4

[21] Maosen Li, Cheng Deng, Tengjiao Li, Junchi Yan, Xinbo Gao, and Heng Huang. Towards transferable targeted attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 641–649, 2020. 1, 2, 5, 6, 13

[22] Qizhang Li, Yiwen Guo, and Hao Chen. Yet another intermediate-level attack. In *European Conference on Computer Vision*, pages 241–257. Springer, 2020. 2

[23] Yingwei Li, Song Bai, Cihang Xie, Zhenyu Liao, Xiaohui Shen, and Alan L Yuille. Regional homogeneity: Towards learning transferable universal adversarial perturbations against defenses. *arXiv preprint arXiv:1904.00979*, 2019. 2, 5

[24] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016. 1, 2

[25] Zihao Liu, Qi Liu, Tao Liu, Yanzhi Wang, and Wujie Wen. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. *arXiv preprint arXiv:1803.05787*, 2018. 7, 8

[26] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 1, 2, 5, 8, 13

[27] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturba-

tions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017. 2

[28] Konda Reddy Mopuri, Aditya Ganeshan, and R Venkatesh Babu. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE transactions on pattern analysis and machine intelligence*, 41(10):2452–2465, 2018. 2

[29] 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. 2

[30] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 5, 7, 8, 13

[31] Muzammal Naseer, Salman Khan, and Fatih Porikli. Local gradients smoothing: Defense against localized adversarial attacks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1300–1307. IEEE, 2019. 7, 13

[32] Muzammal Naseer, Salman H Khan, Harris Khan, Fahad Shahbaz Khan, and Fatih Porikli. Cross-domain transferability of adversarial perturbations. *Advances in Neural Information Processing Systems*, 2019. 2, 3, 5, 6, 7, 8

[33] Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *arXiv preprint arXiv:2105.10497*, 2021. 8

[34] Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Fahad Shahbaz Khan, and Fatih Porikli. On improving adversarial transferability of vision transformers. *arXiv preprint arXiv:2106.04169*, 2021. 8

[35] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. https://distill.pub/2017/feature-visualization. 3

[36] 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, 2018. 2, 3, 5, 6, 7, 8

[37] Konda Reddy Mopuri, Phani Krishna Uppala, and R Venkatesh Babu. Ask, acquire, and attack: Data-free uap generation using class impressions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 19–34, 2018. 2

[38] 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. 2

[39] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 2

[40] Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? In *ArXiv preprint arXiv:2007.08489*, 2020. 5, 7, 8, 13

[41] 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. 2

[42] Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. In *ICLR*, 2020. 1, 2, 11, 13

[43] Weibin Wu, Yuxin Su, Xixian Chen, Shenglin Zhao, Irwin King, Michael R Lyu, and Yu-Wing Tai. Boosting the transferability of adversarial samples via attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1161–1170, 2020. 2

[44] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan Yuille. Improving transferability of adversarial examples with input diversity. In *Computer Vision and Pattern Recognition*. IEEE, 2019. 1, 2, 5, 6, 13

[45] 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, 2018. 2

Supplementary: On Generating Transferable Target Perturbations

We study the effect of augmentations and ensemble learning by analysing class-wise transferability in Appendix A. We further discuss on why augmentations and ensemble learning leads to more transferable targeted patters in Appendix A.1 & Appendix A.2. We then present the vulnerability of batchnorm to black-box targeted perturbations in Appendix B. In Appendix C, we analyze the effect of linear back-propagation of gradients [8] and using more gradients from skip connections [42] on the targeted attack transferability. For the sake of completeness, we report the drop in clean accuracy caused by different defenses including input processing methods (JPEG, Median Blur, and NRP), adversarial training, and stylized training in Appendix D. Names of 100 target classes are provided in Appendix E. Finally, we present visual illustrations to showcase different targeted adversarial patterns found by our method, TTP (Transferable Targeted Perturbations), in Appendix F.

Appendix A. Effect of Augmentations and Ensemble Learning

We proposed a mechanism to explore augmented adversarial space and ensemble learning to boost transferability of the targeted adversarial perturbations found by TTP. A per-class analysis for 10 targets presented in Table 1 reveals that augmentations and ensemble learning increase the adversarial effect for every target. TTP is trained against naturally trained ResNet50 and ResNet ensemble $\mathtt{R}_{ens}:\mathtt{ResNet}\{18,50,101,152\}$ and perturbations are transferred to naturally trained VGG16 and stylized VGG16 [7]. In some cases, such as `Hippopotamus`, augmented learning maximizes the transferability from ResNet50 to naturally trained VGG16 by more than 100% (Table 1). Similarly, we observe that ensemble learning proves to be effective e.g., see `Grey-Owl` in Table 1. VGG16 trained on stylized ImageNet showed higher resistance against targeted adversarial attacks. For example, transferability of perturbations found by TTP for `French Bulldog` distribution is around 11% on VGG16 (SIN) as compared to 63% on VGG16 trained on ImageNet (IN) (Table 1).

Appendix A.1. Why Augmentations boost Transferability?

Ilyas etal [12] showed that adversarial examples can be explained by features of the attacked class label. In our targeted attack case, we wish to imprint the features of the target class distribution onto the source samples within an allowed distance (*e.g.* $l_\infty \leq 16$). However, black-box (unknown) model might apply different set of transformations (from one layer to another) to process such features and reduce the target transferability. Training on adversarial augmented samples allows the generator to capture such targeted features that are robust to transformations that may vary from one model to another.

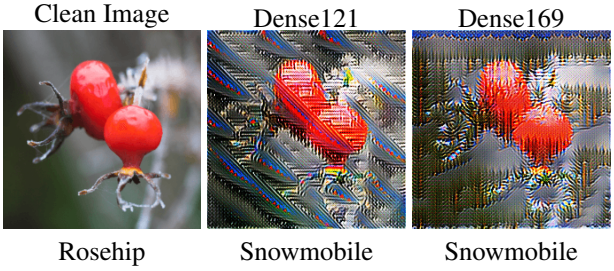


Figure 1: Unconstrained targeted patterns for Snowmobile are shown to demonstrate how discriminators (models) from the same family can capture different information to classify a certain class. Thus, TTP when trained against ensemble of same family models show higher transferability than any of the individual model.

Appendix A.2. Why Ensemble of week Models maximize Transferability?

Different models of the same family of networks can exploit different information to make prediction. One such example is shown in Fig 1. Generators are trained against Dense121 and Dense169 to target Snowmobile distribution. Unrestricted generator outputs reveal that Dense121 is more focused on Snowmobile's blades while Dense169 emphasis background pine tree patterns to discriminate Snowmobile samples. This complementary information from different models of the same family helps the generator to capture more generic global patterns for a given target distribution.

Appendix B. The Vulnerability of Batchnorm

Batchnorm [16] helps in optimization of neural networks as well as increases their clean accuracy. However, our empirical cross-family (Dense $\rightarrow$ VGG$_{BN}$, Dense $\rightarrow$ VGG, ResNet $\rightarrow$ VGG$_{BN}$, ResNet $\rightarrow$ VGG) analysis presented in Fig. 2 suggests that batchnorm makes the model more vulnerable to the targeted adversarial attacks. Adversarial perturbations found by TTP transfer better against models trained using batchnorm as compared to models trained without it (Fig. 2).

Appendix C. Skip Connections and Linear Back-Propagation of Gradients

Dongxian *et al.* [42] observed that while back-propagating, giving more importance to the gradients coming from skip connections can enhance adversarial transferabililty. Similarly, Guo *et al.* [8] showed that encouraging

Source	Augmentations	Target Model: VGG16										
		Grey-Owl	Goose	Bulldog	Hippopotamus	Cannon	Fire-Truck	Model-T	Parachute	Snowmobile	Street-Sign	Average
ResNet50	✗	56.5	80.9	49.0	43.9	61.9	82.9	56.5	89.4	41.3	72.9	63.5
ResNet50	✓	56.7	84.1	63.7	94.9	79.5	91.5	76.5	89.8	70.4	80.8	78.8
R_{ens}	✓	85.1	94.5	63.3	97.8	90.5	95.8	90.7	96.1	89.6	90.4	89.1
		Target Model: VGG16 (SIN)										
ResNet50	✗	1.61	43.1	0.50	40.9	14.9	9.6	5.8	36.2	6.2	19.2	17.8
ResNet50	✓	1.30	69.6	11.6	68.7	17.0	15.2	20.5	33.2	35.4	30.9	30.3
R_{ens}	✓	17.6	77.7	11.4	77.0	59.7	48.4	56.1	72.8	74.1	41.2	53.6

Table 1: ***Per Target Transferability of our Method (*TTP*):*** Top-1 target accuracy (%) with 49.95K ImageNet val. samples for each target. Perturbation budget: $l_\infty \leq 16$. Adversarial perturbations are transferred from naturally trained ResNet50 and ResNet ensemble to naturally trained VGG16 and stylized VGG16 [7]. Augmentations as well as ensemble learning improves efficiency of TTP.

Figure 2: ***Batchnorm Vulnerability to Targeted Transferability :*** {10-Targets (all source) settings}. TTP (Algorithm 1 in the paper) strength is higher against models trained naturally with batchnorm as compared to without batchnorm. Batchnorm [16] provides better optimization and increase model clean accuracy but these empirical results indicate that it also make the model more vulnerable to blackbox targeted attacks. Each value is averaged across 10 targets (see Section 4 in the paper for details) with 49.95k ImageNet val. samples for each target. Perturbation budget is $l_\infty = 16$.

Source	Attack	Natural Training					Augs.	Stylized		Adversarial	
		$VGG19_{BN}$	Dense121	ResNet152	WRN-50-2	VGG16	Augmix	SIN	VGG16 (SIN)	Adv. ($l_\infty = 0.5$)	Adv. ($l_\infty = 1.0$)
ResNet50	PGD [26]	0.8/2.1	1.9/3.7	3.0/4.7	2.5/4.4	0.3/1.5	0.4/1.3	0.1/0.4	0.1/0.3	0.0/0.0	0.0/0.0
	MI [4]	1.5/1.8	3.2/6.2	3.1/5.6	3.0/4.6	1.1/1.4	1.0/1.6	0.3/0.9	0.2/0.4	0.0/0.1	0.0/0.0
	DIM [44]	10.4/14.4	16.2/26.0	13.4/20.9	13.4/19.8	6.4/6.7	4.8/7.7	1.7/3.2	0.5/1.2	0.2/0.5	0.1/0.1
	Po-TRIP [21]	12.5/15.0	18.2/30.0	15.9/23.7	14.2/22.3	7.3/8.9	5.5/9.0	2.1/3.7	0.8/2.0	0.3/0.7	0.1/0.1
	FDA-fd [13]	16.0/25.3	21.0/33.1	19.7/32.9	17.1/28.4	12.0/18.7	15.3/19.3	3.1/6.3	1.2/3.0	0.1/1.9	0.1/0.3
	FDA-N [14]	32.1/38.6	48.3/52.3	37.5/39.0	35.5/40.7	19.0/28.3	20.3/30.3	5.0/16.6	3.0/10.7	0.6/4.7	0.2/0.8
	SGM [42]	19.2/26.3	25.9/40.6	19.7/31.1	21.6/30.4	13.5/13.7	10.5/15.9	2.6/6.1	1.3/2.8	0.5/1.2	0.1/0.3
	SGM [42] + LinBP [8]	22.0/27.1	34.5/40.0	30.5/32.9	25.1/21.0	14.8/15.0	17.3/25.3	4.6/14.3	2.4/8.0	0.3/2.9	0.1/0.3
	Ours (TTP)	**79.0/81.4**	**84.4/87.0**	**81.9/86.6**	**80.2/81.2**	**79.4/78.2**	**72.7/81.2**	**30.5/42.4**	**29.3/36.9**	**5.5/50.1**	**0.4/17.1**

Table 2: ***Target Transferability:*** {10-Targets (sub-source)} Top-1 target accuracy (%) averaged across 10 targets. Perturbation budget: $l_\infty \leq 16/32$. SIN [7] and Adv (l_∞=0.5), and Adv (l_∞=1.0) [40] are ResNet50 models trained using stylized and adversarial examples, respectively. Augs. represents augmentation based training [10] of ResNet50.

Model	Defense	Accuracy	Difference
$VGG19_{BN}$	–	74.24	0.0
	JPEG	67.34	-6.90
	Blur	53.86	-20.38
	NRP	72.00	-2.24
Dense121	–	74.65	0.0
	JPEG	68.92	-5.73
	Blur	61.27	-13.38
	NRP	72.01	-2.63
ResNet50	–	76.15	0.0
	JPEG	70.82	-5.33
	Blur	61.30	-14.85
	NRP	73.21	-2.94

Table 3: ***Effect of Input Processing on Clean Accuracy:*** Top-1 (%) accuracy on ImageNet val. set (50k images). Median Blur with window size 5×5 causes large drop in clean accuracy while NRP [30] has the least effect on the model's clean accuracy.

linearity while back-propagating gradients improve transferability. Here, we analyze target transferability of both of these techniques [42, 8] and present a holistic comparison between all the considered iterative and generative attacks in Table 2. Our approach sets new state-of-the-art.

Appendix D. Clean Accuracy vs. Defenses

We evaluate the effect of different defenses on model's clean accuracy. We study the input processing methods including JPEG with quality 50% [31], Median Blur with kernel size 5×5 [31] and NRP [30] as well as different training mechanisms including Augmix [10], stylized [7] and adversarial training methods [26, 40]. Results are presented in Tables 3 & 4. We observe that Median Blur causes a significant drop in clean accuracy (Table 3) while among training methods, stylized training (SIN) [7] has the most negative effect on the clean accuracy.

Model	Training Type	Accuracy	Difference
ResNet50	IN	76.15	0.0
	SIN	60.18	-15.97
	SIN-IN	74.59	-1.56
	Augmix	77.53	+1.38
	Adv. ($l_\infty, \epsilon = .5$)	73.73	-2.42
	Adv. ($l_\infty, \epsilon = 1$)	72.05	-4.10
	Adv. ($l_2, \epsilon = .1$)	74.78	-1.37
	Adv. ($l_2, \epsilon = .5$)	73.16	-2.99
VGG16	IN	71.59	0.0
	SIN	52.26	-19.33

Table 4: ***Effect of Robust Training on Clean Accuracy:*** Top-1 (%) accuracy on ImageNet val. set (50k images). Every training mechanism with the exception of Augmix [10] reduces model's clean accuracy. Stylized training [7] causes significant drop in accuracy in comparison to other types of training methods.

Appendix E. 100 Targets Names

The performance of TTP is evaluated against the following randomly selected 100 targets (see Sec. 4.1 of the paper). We divide ImageNet classes into 100 mutually exclusive sets. Each set contains 10 classes. We randomly selected one target from each set.

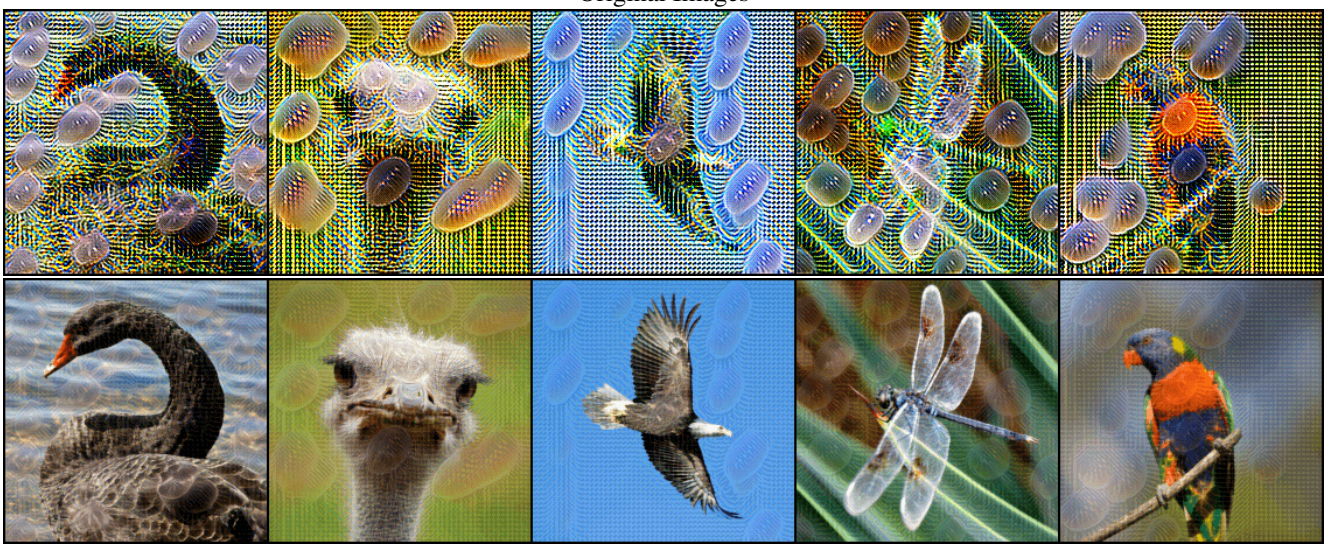
`Tiger-Shark`, `Bulbul`, `Grey-Owl`, `Terrapin`, `Komodo-Dragon`, `Thunder-Snake`, `Trilobite`, `Scorpion`, `Quail`, `Goose`, `Jellyfish`, `Slug`, `Flamingo`, `Bustard`, `Dowitcher`, `Chihuahua`, `Beagle`, `Weimaraner`, `Lakeland-Terrier`, `Australian-Terrier`, `Golden-Retriever`,

English-Setter, Komondor, Appenzeller, French-Bulldog, Chow, Keeshond, Hyaena, Egyptian-Cat, Lion, Bee, Leafhopper, Sea-Urchin, Zebra, Hippopotamus, Polecat, Gorilla, Langur, Eel, Anemone-Fish, Airliner, Banjo, Bassinet, Beaker, Bell-Cote, Bookcase, Buckle, Cannon, CD-Player, Chain-Saw, Coil, Cornet, Crutch, Dome, Electric-Guitar, Fire-Truck, Garbage-Truck, Greenhouse, Grocery-Store, Honeycomb, iPod, Jigsaw-Puzzle, Lipstick, Maillot, Maze, Military-Uniform, Model-T, Neck-Brace, Overskirt, Parachute, Pay-Phone, Pickup, Pirate-Ship, Poncho, Purse, Rain-Barrel, Rotisserie, School-Bus, Sewing-Machine, Shopping-Cart, Snowmobile, Spatula, Stove, Sunglass, Teapot, Toaster, Tractor, Umbrella, Velvet, Wallet, Whiskey-Jug, Street-Sign, Ice-Lolly, Pretzel, Cardoon, Hay, Pizza, Volcano, Rapeseed, Agaric

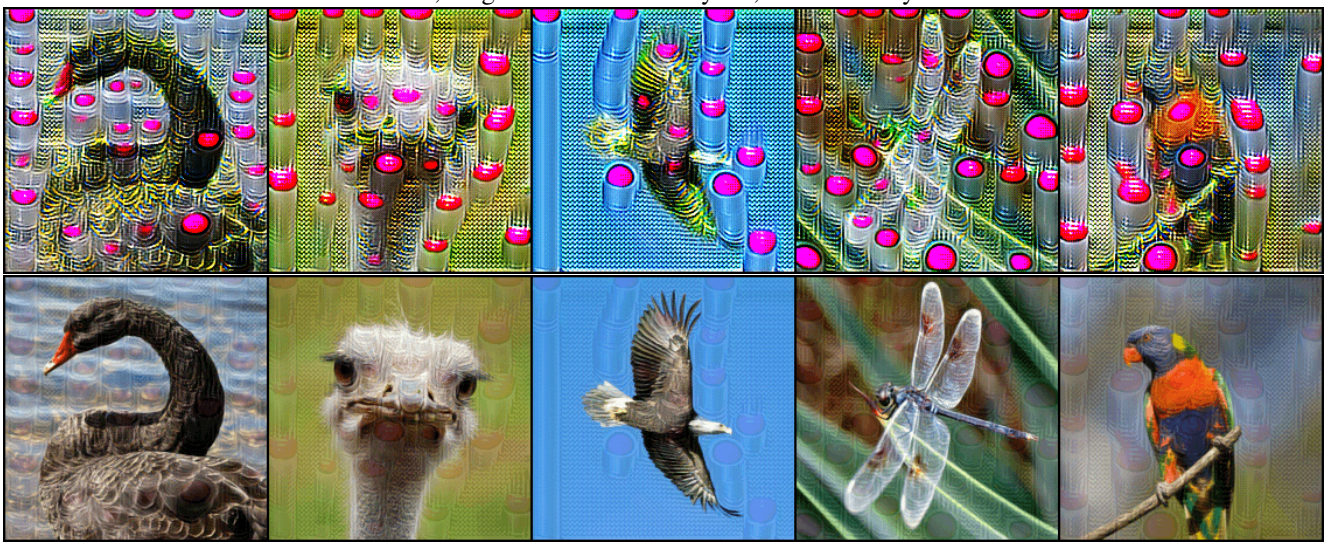
Appendix F. Visual Demos

Figures 3, 4, 5, 6, 7 and 8 show different targeted patterns produced by TTP trained against naturally trained ResNet50. We demonstrate how adversarial patterns evolve as TTP learns to model a certain target distribution from different networks of the same family in Figures 9 and 10.

Original Images



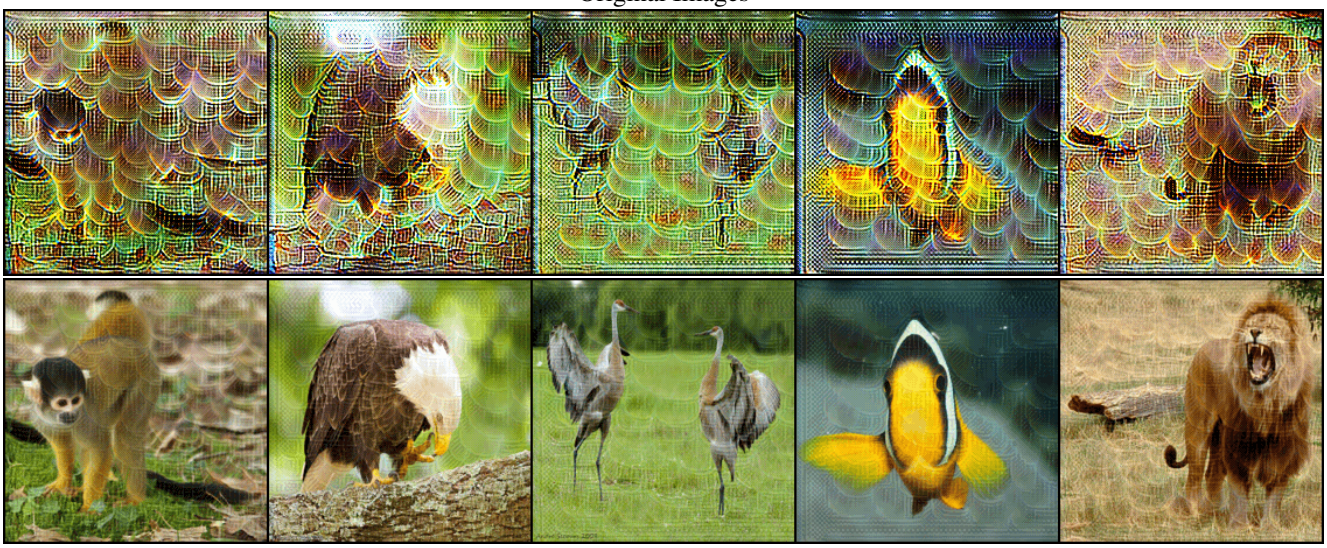
Source model: ResNet50, Target Distribution: Jellyfish, Transferabiliy to Dense121: 90.05 %



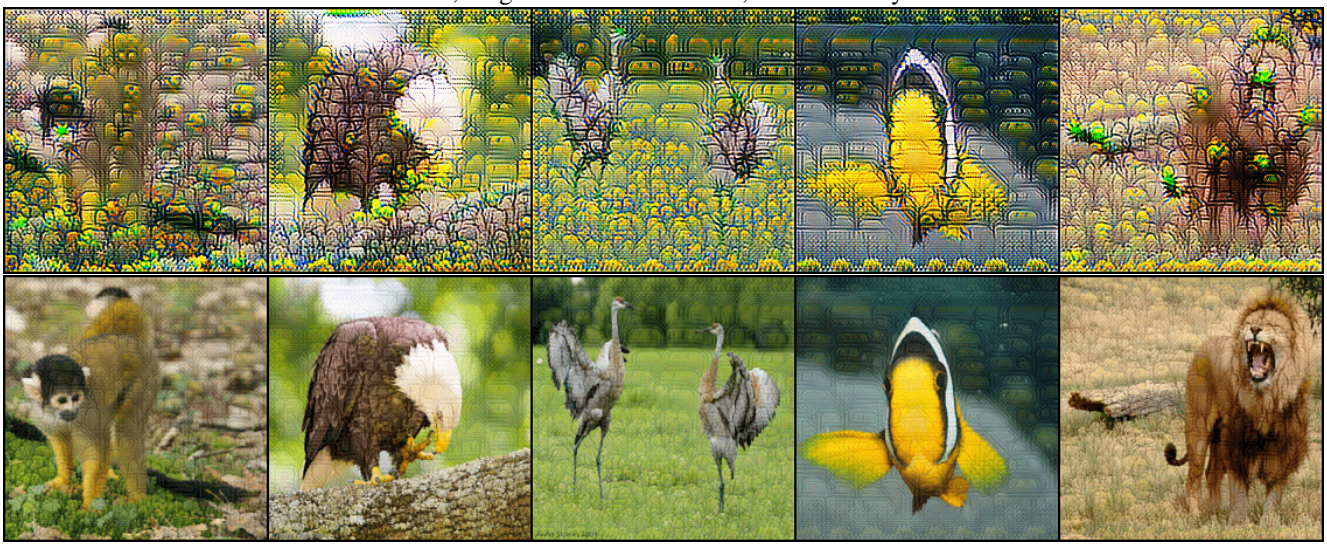
Source model: ResNet50, Target Distribution: Lipstick, Transferability to Dense121: 95.20 %

Figure 3: Targeted adversaries produced by TTP (before and after valid projection) trained against ResNet50. Observe that adversarial patterns are not constant rather TTP adapts to the input sample and adds different patterns to different samples to achieve maximum transferability. Transferability is measured as Top-1 target accuracy on the ImageNet val. set (49.95k samples excluding the target images).

Original Images



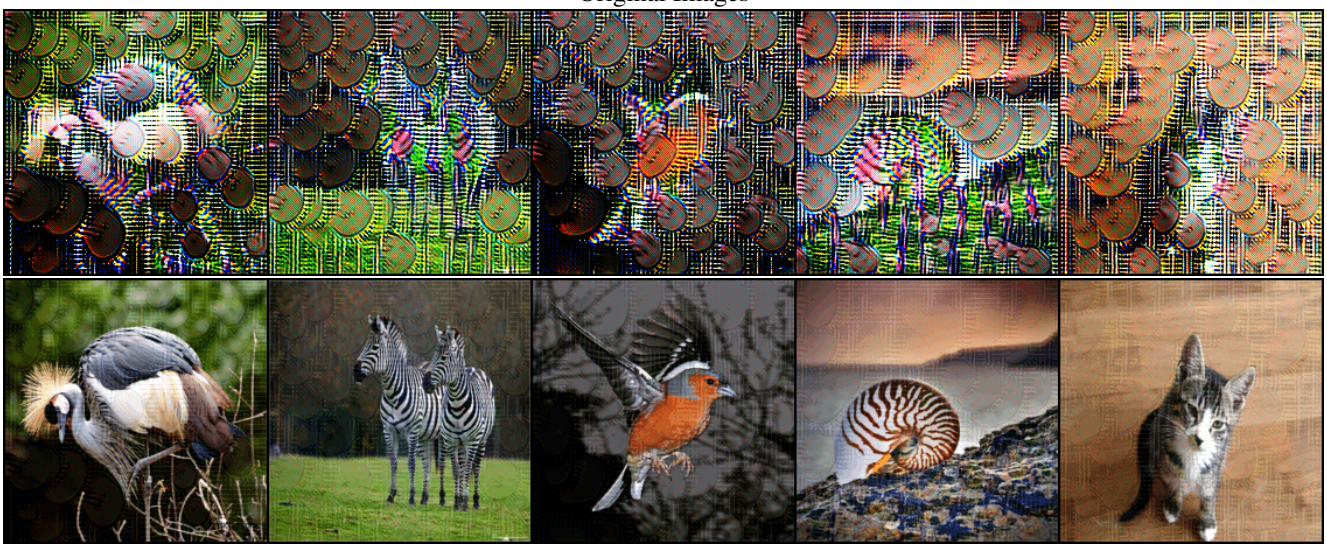
Source model: ResNet50, Target Distribution: Stove, Transferabiliy to Dense121: 36.86%



Source model: ResNet50, Target Distribution: Rapeseed, Transferabiliy to Dense121: 49.59%

Figure 4: Targeted adversaries produced by TTP (before and after valid projection) trained against ResNet50. Observe that adversarial patterns are not constant rather TTP adapts to the input sample and adds different patterns to different samples to achieve maximum transferability. Transferability is measured as Top-1 target accuracy on the ImageNet val. set (49.95k samples excluding the target images).

Original Images

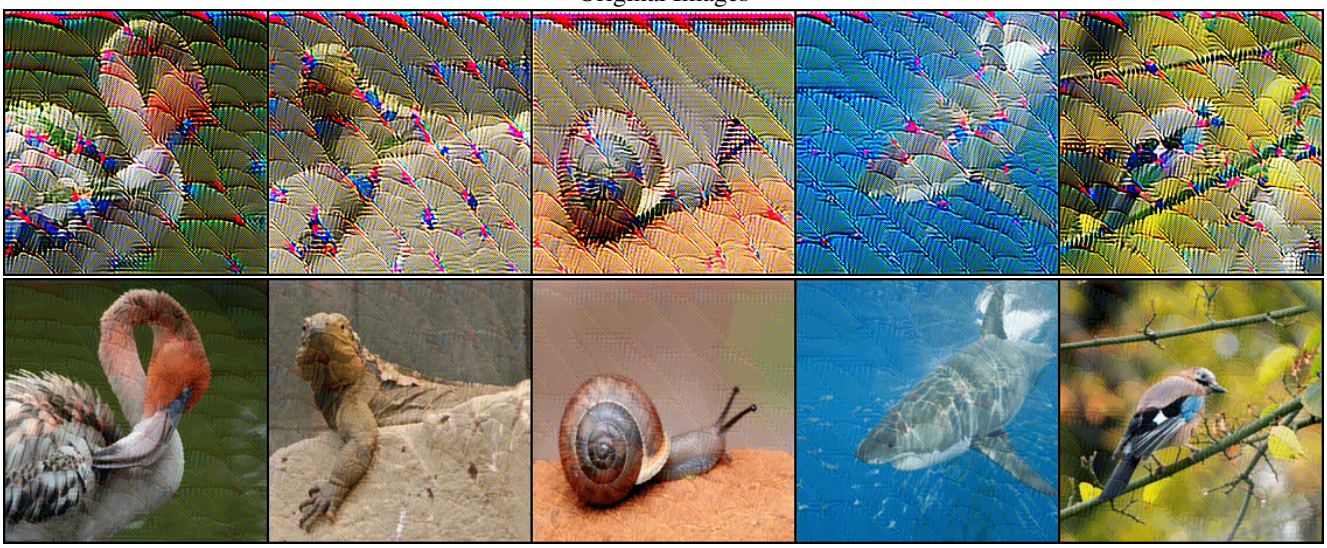


Source model: ResNet50, Target Distribution: Banjo, Transferabiliy to Dense121: 82.95%

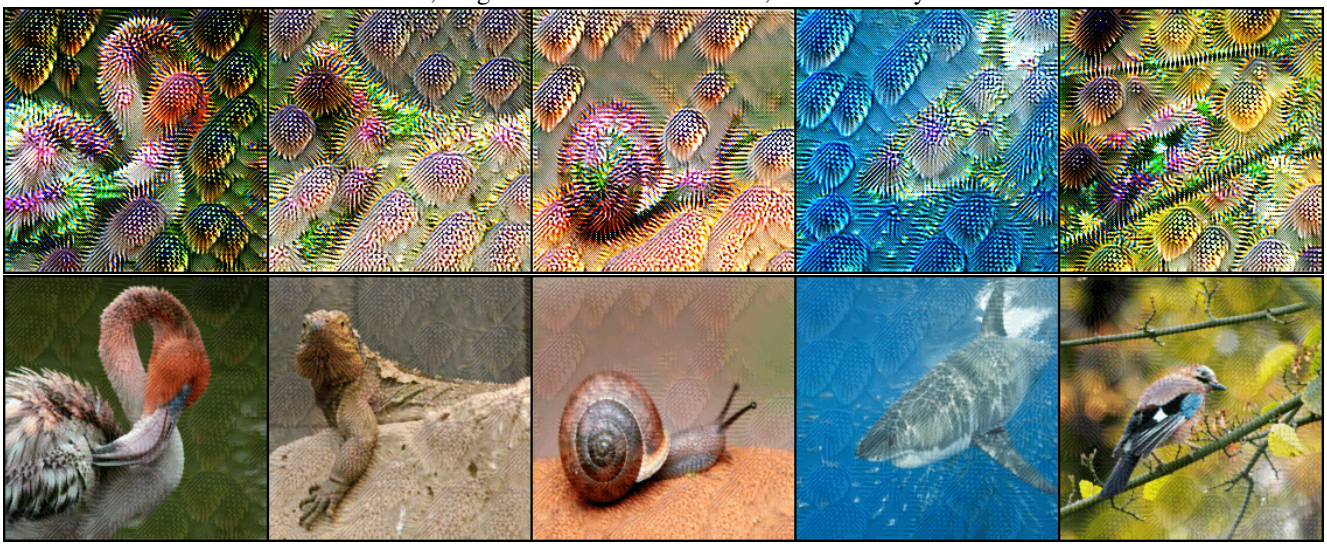
Source model: ResNet50, Target Distribution: Anemone Fish, Transferabiliy to Dense121: 74.45%

Figure 5: Targeted adversaries produced by TTP (before and after valid projection) trained against ResNet50. Observe that adversarial patterns are not constant rather TTP adapts to the input sample and adds different patterns to different samples to achieve maximum transferability. Transferability is measured as Top-1 target accuracy on the ImageNet val. set (49.95k samples excluding the target images).

Original Images



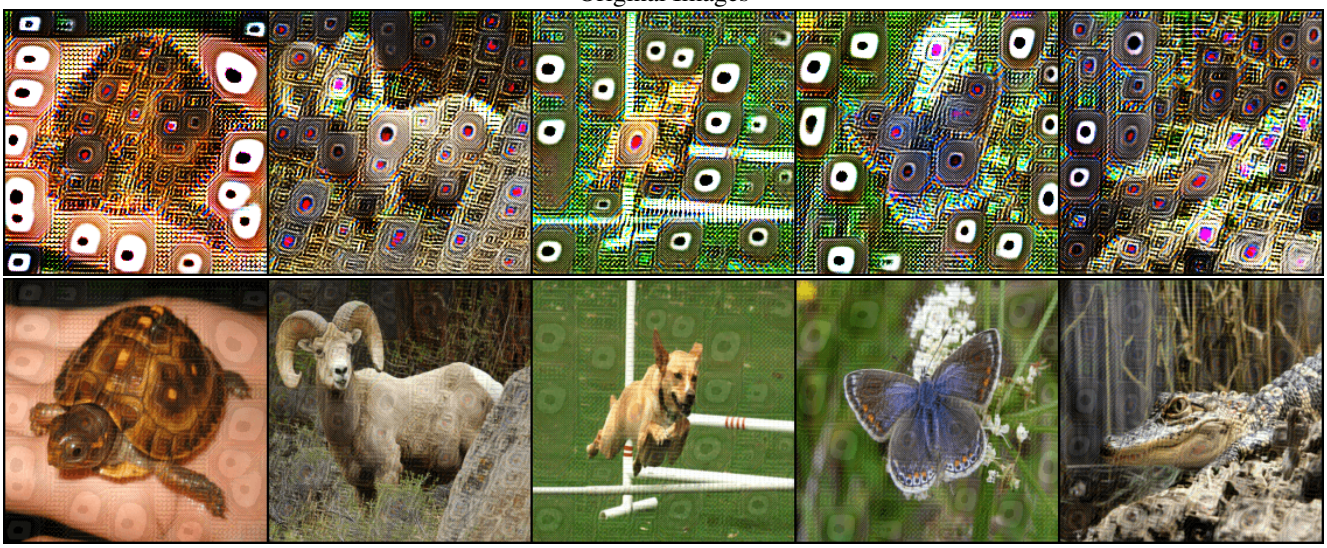
Source model: ResNet50, Target Distribution: Parachute, Transferabiliy to Dense121: 95.30%



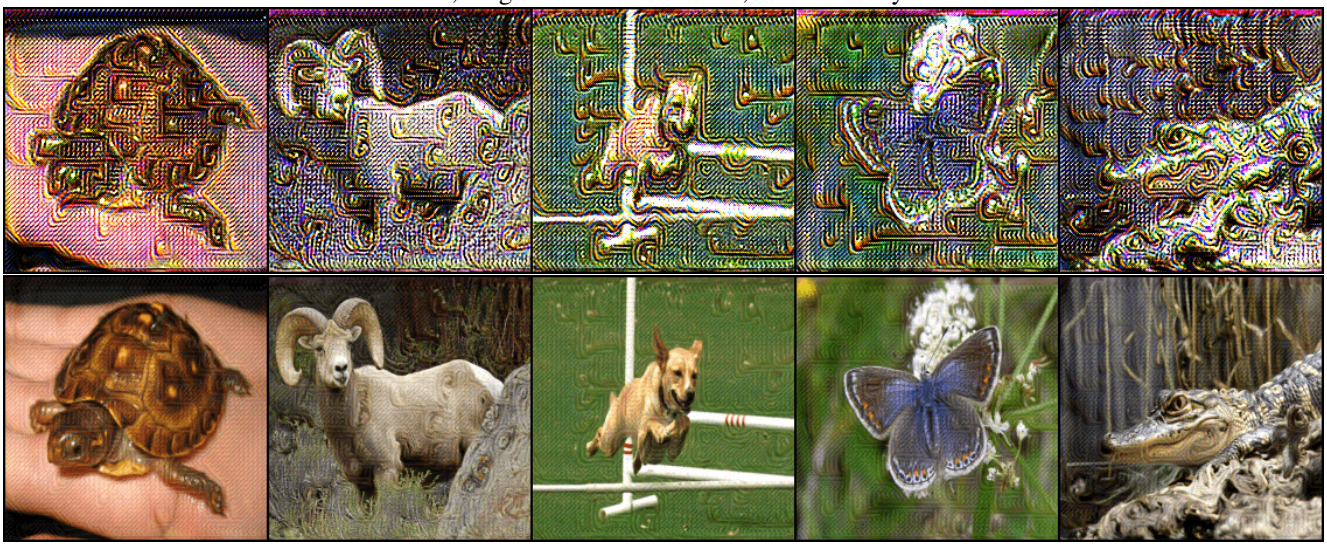
Source model: ResNet50, Target Distribution: Sea Urchin, Transferabiliy to Dense121: 89.10%

Figure 6: Targeted adversaries produced by TTP (before and after valid projection) trained against ResNet50. Observe that adversarial patterns are not constant rather TTP adapts to the input sample and adds different patterns to different samples to achieve maximum transferability. Transferability is measured as Top-1 target accuracy on the ImageNet val. set (49.95k samples excluding the target images).

Original Images

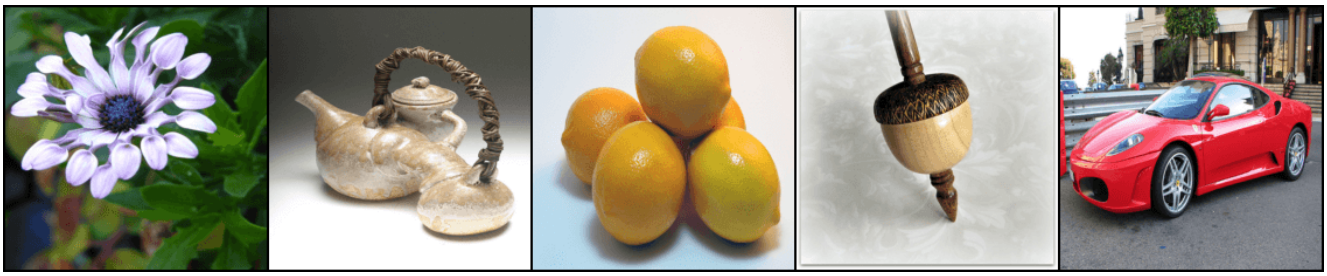


Source model: ResNet50, Target Distribution: iPOD, Transferabiliy to Dense121: 69.86%

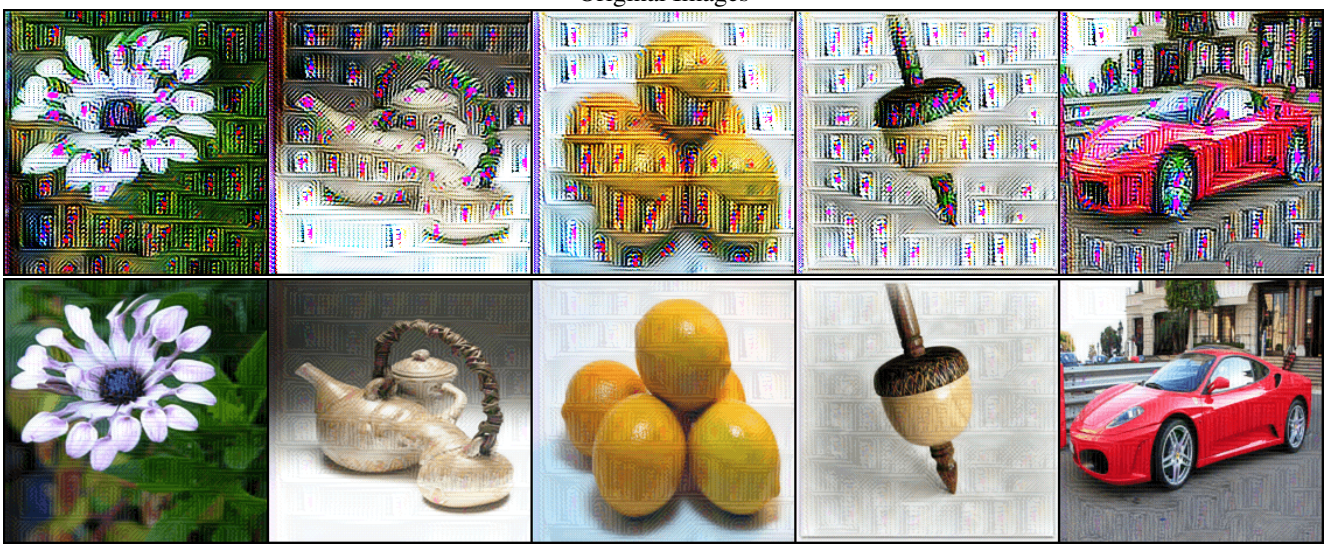


Source model: ResNet50, Target Distribution: Buckle, Transferabiliy to Dense121: 77.06%

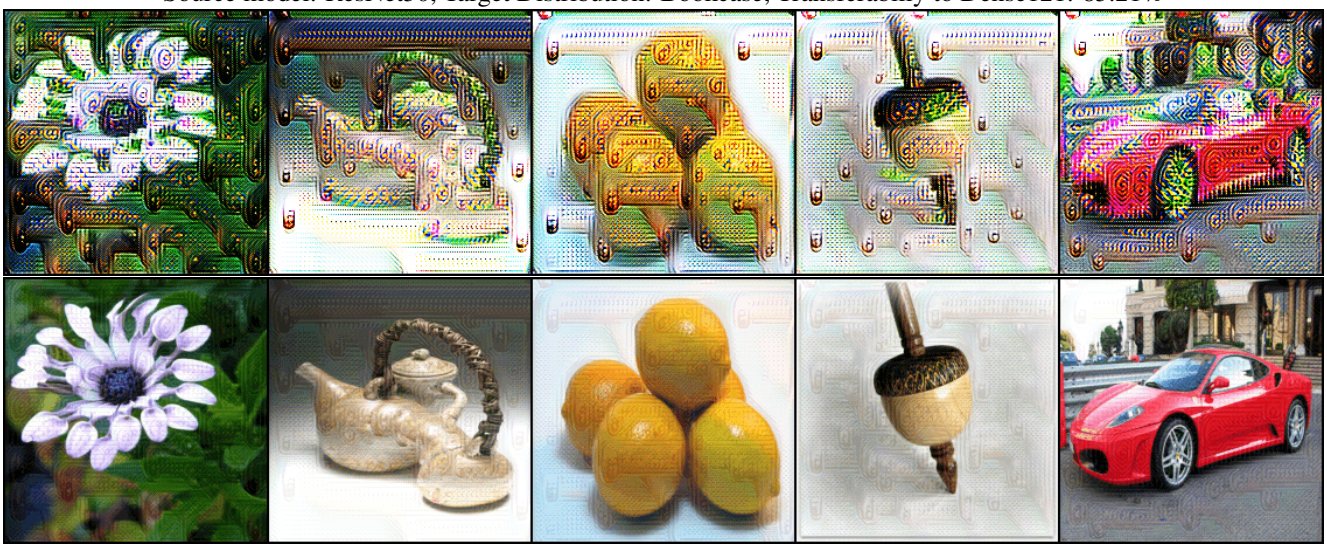
Figure 7: Targeted adversaries produced by TTP (before and after valid projection) trained against ResNet50. Observe that adversarial patterns are not constant rather TTP adapts to the input sample and adds different patterns to different samples to achieve maximum transferability. Transferability is measured as Top-1 target accuracy on the ImageNet val. set (49.95k samples excluding the target images).



Original Images



Source model: ResNet50, Target Distribution: Bookcase, Transferabiliy to Dense121: 85.21%



Source model: ResNet50, Target Distribution: Sewing Machine, Transferabiliy to Dense121: 67.26%

Figure 8: Targeted adversaries produced by TTP (before and after valid projection) trained against ResNet50. Observe that adversarial patterns are not constant rather TTP adapts to the input sample and adds different patterns to different samples to achieve maximum transferability. Transferability is measured as Top-1 target accuracy on the ImageNet val. set (49.95k samples excluding the target images).

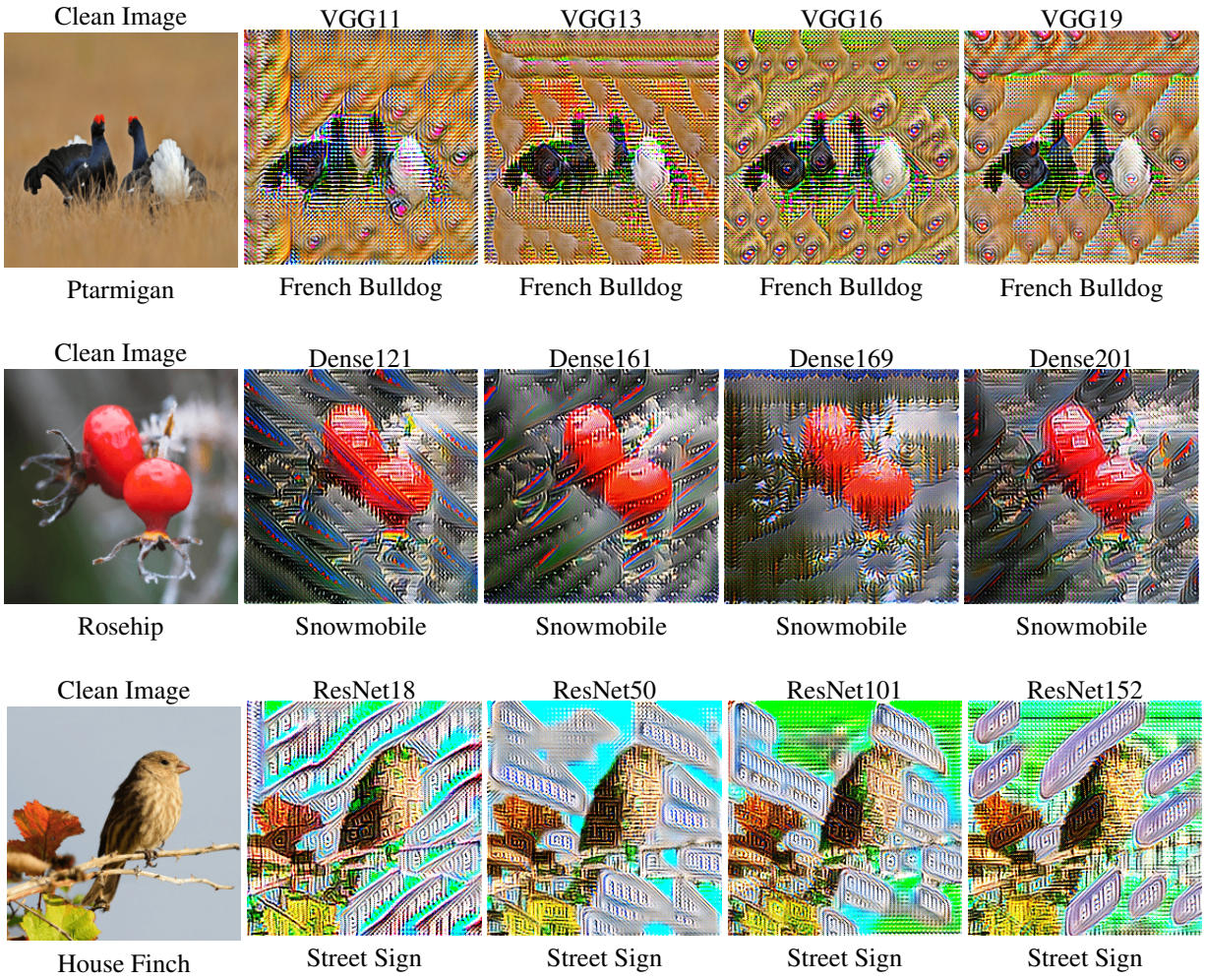


Figure 9: ***Evolution of* TTP*:*** Unconstrained targeted adversarial patterns generated by TTP are shown to demonstrate how TTP evolves as it learns perturbations from different source models of a certain family of networks.

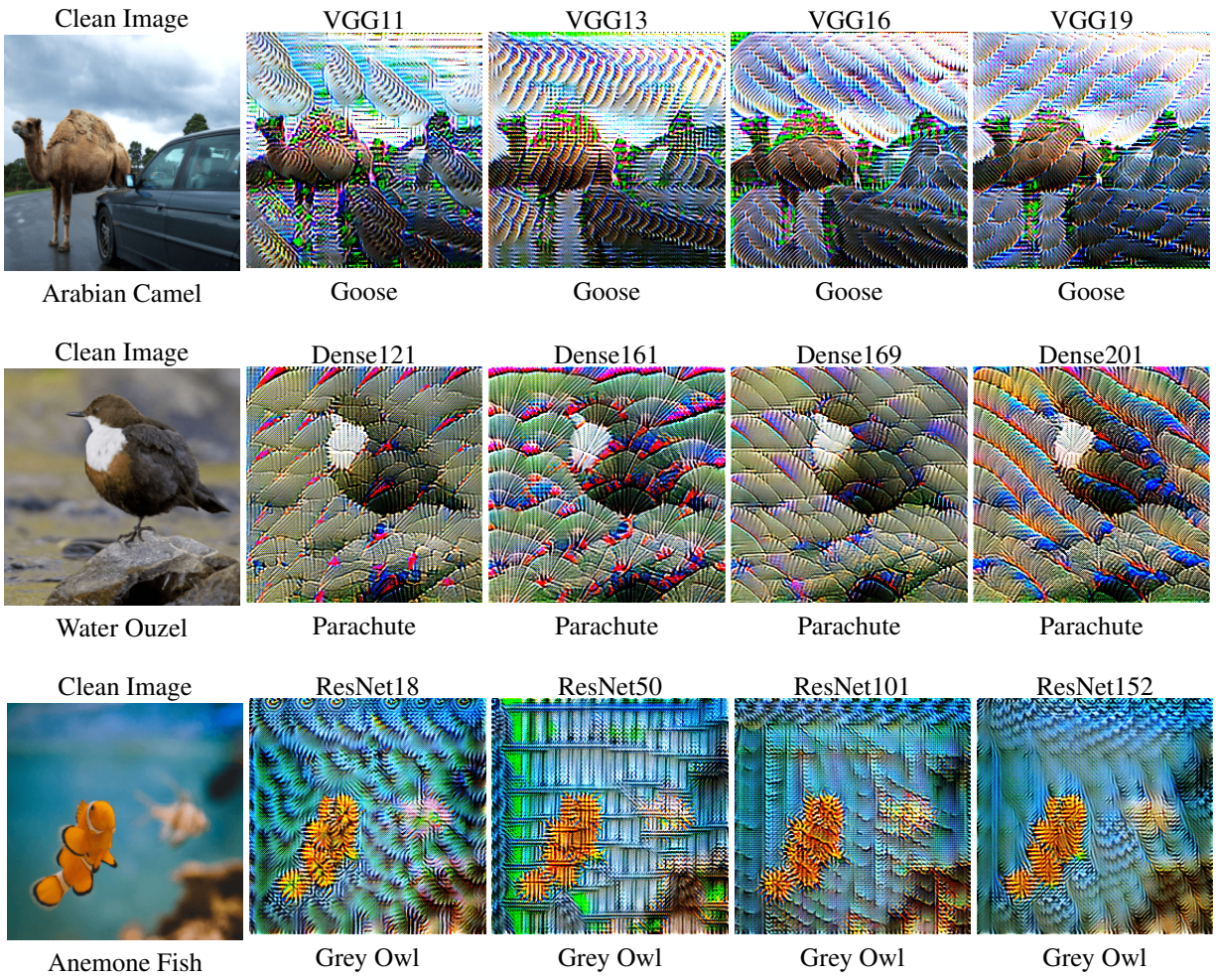


Figure 10: ***Evolution of* TTP*:*** Unconstrained targeted adversarial patterns generated by TTP are shown to demonstrate how TTP evolves as it learns perturbations from different source models of a certain family of networks.