A Little Robustness Goes a Long Way: Leveraging Universal Features for Targeted Transfer Attacks

Anonymous Author(s)
Affiliation
Address
email

Abstract

Adversarial examples for neural network image classifiers are known to be transferable: examples optimized to be misclassified by a source classifier are often misclassified as well by classifiers with different architectures. However, *targeted* adversarial examples—optimized to be classified as a chosen target class—tend to be less transferable between architectures. While prior research on constructing transferable targeted attacks has focused on improving the optimization procedure, in this work we examine the role of the source classifier. Here, we show that training the source classifier to be "slightly robust"—that is, robust to small-magnitude adversarial examples—substantially improves the transferability of targeted attacks, even between architectures as different as convolutional neural networks and transformers. We argue that this result supports a non-intuitive hypothesis: on the spectrum from non-robust (standard) to highly robust classifiers, those that are only *slightly* robust exhibit the most *universal* features—ones that tend to overlap with the features learned by other classifiers trained on the same dataset. The results we present provide insight into the nature of adversarial examples as well as the mechanisms underlying so-called "robust" classifiers.

1 Introduction

Neural-network image classifiers are well-known to be susceptible to adversarial examples—images that are perturbed in a way that is largely imperceptable to humans but that cause the neural network to make misclassifications. Much research has gone into methods for constructing adverarial examples in order to understand the nature of neural-network vulnerabilities and to develop methods to make neural-network classifiers more robust to attacks [7, 11, 21, 43, 47, 52, 69].

Untargeted adversarial examples are designed to elicit an unspecified incorrect class, while *targeted* adversarial examples are designed to elicit a specific (incorrect) *target* class. A given adversarial perturbation is designed via optimization with respect to a given trained network; here we call this the *source* network. While untargeted adversarial examples are often transferable—examples designed to attack a source network also successfully attack other trained networks with different parameters and architectures [21, 69]—targeted adversarial examples tend to be less transferable [42].

Prior research on constructing transferable adversarial examples for image classifiers has focused primarily on improving optimization methods for generating successful image perturbations. In this paper, we take a different tack—focusing on the source neural network used in constructing adversarial examples. Specifically, we find that "slightly-robust" convolutional neural networks (CNNs)—ones that have been trained to be robust to small adversarial perturbations—can be leveraged to substantially improve the transferability of targeted adversarial examples to different architectures. Surprisingly, we show that targeted adversarial examples constructed with respect to a slightly robust CNN transfer successfully not only to different CNN architectures but also to transformer architectures such as

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

ViT [16] and even CLIP [54], which is trained on different objective than the source CNN. Such transfers are all "black box" attacks—while generating an adversarial example requires knowledge of the source network's architecture, no such knowledge is required for the black-box architectures that can also be attacked by the same example.

In addition, we leverage the techniques of adversarial transferability to examine which features are learned by neural networks. In accordance with prior work [67], we find that, on the spectrum from non-robust (standard) to highly robust classifiers, those that are only *slightly* robust exhibit the most *universal* features—ones that tend to overlap with the features learned by *all* other classifiers trained on the same dataset [50]. This notion of universality can explain why slightly robust networks give rise to more transferable adversarial attacks and have better weight initializations for downstream transfer-learning tasks [41, 58, 72, 76].

The main contributions of this paper are the following:

1. We demonstrate that adversarial examples generated with respect to slightly robust CNNs are more transferable than those generated with respect to standard (non-robust) networks, This transferability extends not only to other CNNs, but also to transformer architectures.
2. We find that as the robustness of the source network increases, the transferability of targeted adversarial examples to adversarially-defended networks improves substantially.
3. We examine the role of various hyperparameters in the optimization search, including the step-size and the ε-robustness parameter of the adversarially-trained source networks.
4. We argue that the transferability of adversarial examples optimized over slightly-robust source networks can be explained by prior hypotheses that suggest that slightly-robust networks learn more universal features than non-robust networks. In particular, we find that when performing adversarial attacks designed to target specific features (rather than a class label), adversarial examples optimized over slightly-robust networks transfer features substantially better than adversarial examples optimized over non-robust networks.

2 Background

Adversarial Examples. In this paper we are primarily concerned with properties of source networks that facilitate transferability of adversarial examples. Let $f : \mathcal{X} \to \mathcal{Y}$ denote a "white-box" network (i.e., one whose architecture and weights are known to the adversary) and let $g : \mathcal{X} \to \mathcal{Y}$ denote a "black-box" network (weights and architecture are unknown to the adversary). Let $(x, y) \in \mathcal{X} \times \mathcal{Y}$ be an (unperturbed) input-label pair, where $\mathcal{X}$ is the input-space and $\mathcal{Y}$ is the label-space. Given a maximum perturbation size ε, our goal is to optimize over f to construct a transferable adversarial example $x + \delta$ where $\|\delta\|_\infty \leq \varepsilon$ and such that $g(x + \delta) \neq y$ for the *untargeted* case, and $g(x + \delta) = t$ for some target class $t \in \mathcal{Y}$ for the *targeted* case.

Optimizers. Prior research has identified a number of methods for optimizing adversarial examples given a white-box classifier f, many of which are based on the Iterative Fast Gradient Sign Method (I-FGSM) [21, 38], in which a perturbation δ_i is iteratively updated to maximize the loss of the network while obeying an ℓ_∞ norm constraint.

We adopt the state-of-the-art method recently proposed by [87], which combines three variants of I-FGSM and optimizes over many steps:

1. Diverse Input Iterative Fast Gradient Sign Method (DI^2-FGSM), which applies a random affine transformation to the input at each step prior to computing the gradient [83],
2. Translation-Invariant Iterative Fast Gradient Sign Method (TI-FGSM), which convolves the gradient with a Gaussian filter [14],
3. Momentum Iterative Fast Gradient Sign Method (MI-FGSM), in which a momentum term is added to the gradient [13].

We follow the convention of Zhao et al. [87] and call the combination of these processes *TMDI-FGSM*.

For targeted adversarial examples, the loss function L should be maximized when the target label is predicted with high confidence. For untargeted adversarial examples, this occurs when the predicted label differs from the true label, and when the true label is given a low confidence. A number of adversarial loss functions have been proposed, including standard cross-entropy loss [69], CW loss

[7], and feature-disruption loss [32]. We use the highly-effective logit loss, proposed by [87], which is maximized for targeted examples when the logit score for a target class (i.e., the value of the output neuron associated with the target class prior to the softmax operation) is maximized. Similarly, the untargeted version aims to minimize the logit score associated with the true class.

Constructing Robust Source Networks. We construct robust source networks by performing adversarial training with projected gradient descent [21, 43]. Each source network is trained to be robust to adversarial examples with ℓ_2 norm less than a specified ε parameter, which we call the *robustness parameter*. For this paper, we rely on pre-trained robust ImageNet models [58], which have been released under the MIT License. These source networks, along with many of our experiments, are implemented in PyTorch [53].

Features. In this paper, we will refer to neural network *features* [30, 67, 74]. A feature $f : \mathcal{X} \to \mathbb{R}$ maps input to a real number to describe how "strongly" the feature appears in the image. Every neuron in a neural network computes a feature. However, we are primarily concerned with the *representation-layer* features, i.e. the features computed by the neurons in the penultimate layer [30]. When we are referring to the features of a specific neural network, we are referring to the features described by the neurons in the representation layer. When we say that the features of two different neural networks *overlap*, we mean that the patterns of the input that affect the features of one neural network also affect the features of the other neural network. When features are easily manipulated by small perturbations to the input, they are said to be *non-robust*, and likewise when they are not easily manipulated in this way, they are said to be *robust* [30]. Robust networks, i.e., networks that are less vulnerable to adversarial perturbations, should rely primarily on robust features, though non-robust networks can rely on a mixture of non-robust and robust features [30, 67].

3 Transferability of Adversarial Examples

In this section, we describe the methodology and results of our experiments on the transferability of adversarial examples as a function of the robustness of the source network. We evaluate the targeted and untargeted effectiveness of each constructed adversarial example on a collection of convolutional classifiers, Xception [9], VGG [64], ResNet [23, 24], Inception [70], MobileNet [27], DenseNet [28], NASNetLarge [89], and EfficientNet [71], as well as two transformer-based classifiers, ViT [16] and CLIP [54]. Here we use the term *destination network* to denote the networks on which we will evaluate transferability of adversarial examples that were generated with respect to a source network. For our ImageNet experiments, we rely on pre-trained models [10].

Generating Adversarial Examples. We choose 1000 images randomly from the ImageNet validation dataset such that every image has a different class. We generate target classes randomly for each image such that each class is a target for exactly one image, and no image has a target that is the same as its true class. For each classifier (of different robustness), we generate targeted adversarial examples for each of the 1000 selected images, targeting the associated target class. To optimize each adversarial example, we run the TMDI-FGSM algorithm for 300 iterations. When the exact image input dimensions differs between the source and destination network, we rescale the adversarial example to fit the dimensions required by the destination network using linear interpolation. To generate adversarial examples, we use the Robustness library [17].

Transferability to Convolutional Network Classifiers. We find that across every destination convolutional network, a source network with small robustness parameter improves transfer success rate in both the targeted and untargeted setting compared to the success rate of the non-robust ($\varepsilon = 0$) network (Figure 1). Our untargeted results can be found in the Appendix. The success peak is approximately the same across every destination network ($\varepsilon = 0.1$), suggesting that there is an optimal source-network robustness parameter in order to maximize transferability to convolutional architectures neural networks trained on ImageNet. Note that our attacks do not exclude images that are already misclassified (untargeted case), and misclassified as the target class (targeted case) prior to adversarial perturbation. In Figures1, we plot the baseline performance of the attacks, i.e., what would be measured as the performance of the attacks if the images were unperturbed.

As a practical consideration, this means that attacks can be conducted with limited knowledge of the architecture of the destination classifier. Interestingly, we observe a substantial drop in attack success

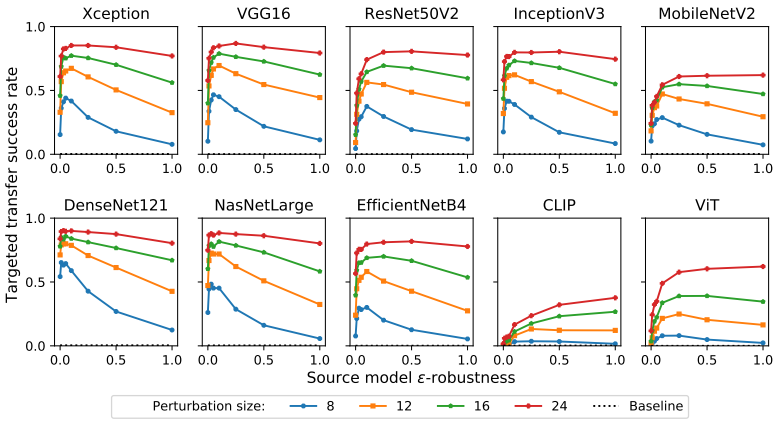

Figure 1: Targeted transfer attack success rate against ImageNet classifiers using ε-robust ResNet50 source models with perturbation size $\|\delta\|_\infty \leq 16/256$. Higher is a more successful attack. Baseline refers the rate at which unperturbed images are classified as the target class. (Best viewed in color.)

as robustness increases past the optimal value. While not shown in Figure 1, this trend continues as ε increases above 1 (see Appendix). We hypothesize that as the robustness of a source network improves substantially, the network begins to entirely ignore many of the non-robust features relied upon by the non-robust destination networks, and thus attacks do not modify these features, reducing attack success. This is consistent with prior literature [74]. We dedicate the second half of this paper to describing a possible mechanism to explain these results.

Transformer-Based Classifiers. Few studies have addressed the robustness of transformer-based classifiers to transfer attacks [62]. To our knowledge, our paper is the first to address *targeted* transfer attacks against transformer architectures. Shao et al. [62] finds that transformer-based image classifiers are more robust to transfer attacks than convolutional classifiers, especially when the source model is convolutional. We find an even more striking result: with attack performance near zero, previously published methods are almost entirely ineffective at constructing targeted transferable adversarial examples using convolutional source models (Table 1). This suggests that the features learned by transformer-based models and non-robust convolutional models are largely different.

However, we find that using a *slightly-robust* ResNet50 classifier as a source network dramatically improves the transferability of targeted adversarial examples to transformer-based classifiers. The optimal robustness parameter for the source network is different for destination transformer networks and destination convolutional networks, though we fined that any amount of robustness in the source model (below a critical value) substantially improves transferability. Thus, in a real-world black-box attack setting—in which the destination network's architecture is entirely unknown—an adversary could find a balance between a robustness parameter that optimizes transfer performance for CNN classifiers and for transformer-based classifiers.

CLIP. Radford et al. [54] describes CLIP, a transformer-based classifier based on the ViT architecture. CLIP is trained to simultaneously encode images and short textual descriptions of the images. CLIP can be used for highly effective zero-shot classification by determining which class label, encoded as text, has an encoding most similar to that of the input image. Despite the fact that CLIP has not been explicitly trained with ImageNet labels or to optimize for the classification task, we find that the transfer performance of targeted adversarial examples is improved substantially when the source network (ResNet50) is slightly robust, again suggesting that slightly-robust neural networks rely on features which are universal among non-robust networks. For our experiments, we use the CLIP architecture based on ViT-B/32.

Improvements Upon Existing Attacks. We directly compare the targeted transfer attack success rate to previous state-of-the-art black-box attacks and find that our method substantially outperforms the previous methods under similar constraints (Table 1). In particular, we evaluate our method's attack performace with three different loss functions: standard cross-entropy loss (Xent), Poincaré

distance with a triplet loss term (Po+Trip) [39], and logit loss. We include a comparison with the feature distribution attack (FDA) [31], however, FDA requires that we train multiple supplemental models for each individual target class, which would require thousands of supplemental models to attack all thousand classes of ImageNet. Thus, we do not perform a direct comparison and instead report the targeted transfer attack success rate that is reported in the original FDA paper [31].

Table 1: Direct comparison of targeted transfer attack success rate between our technique (slightly-robust source models, i.e., $\varepsilon > 0$) and previously proposed strong baseline attacks (non-robust source models, i.e., $\varepsilon = 0$). We compare three different loss functions: cross-entropy, Poincaré distance combined with triplet loss, and logit loss. In addition, we report the success rate of FDA from the original paper (see text for discussion). We limit the ℓ_∞ norm of the adversarial examples to the standard value of $16/255$.

		Xcept	VGG16	RN50v2	IncV3	MNv2	DN121	NNL	ENB4	ViT	CLIP
Xent	$\varepsilon = 0$	10.4	9.6	4.6	10.6	6.4	40.5	13.1	6.8	0.8	0.1
Po+Trip	$\varepsilon = 0$	20.8	15.2	10.0	23.0	11.6	59.3	31.2	14.2	1.3	0.3
Logit	$\varepsilon = 0$	45.9	40.0	15.3	43.6	22.9	77.9	60.3	39.6	3.9	0.4
FDA*	$\varepsilon = 0$	–	43.5	–	–	22.9	57.9	–	–	–	–
Xent	$\varepsilon = 0.1$	54.0	59.4	45.8	50.8	32.1	78.8	66.0	41.1	8.6	2.4
Po+Trip	$\varepsilon = 0.1$	59.1	57.9	53.0	56.5	39.2	78.4	72.6	45.1	11.4	3.3
Logit	$\varepsilon = 0.1$	**77.2**	**78.8**	64.5	**73.1**	**52.5**	**84.0**	**81.6**	**68.9**	33.4	11.2
Xent	$\varepsilon = 1$	60.4	69.3	**66.6**	58.2	46.7	69.9	61.3	56.9	29.9	19.9
Po+Trip	$\varepsilon = 1$	48.5	54.4	60.2	49.5	39.9	62.6	53.3	45.0	22.0	12.4
Logit	$\varepsilon = 1$	56.1	62.4	59.5	55.0	47.2	67.0	58.3	53.6	**36.0**	**26.7**

Attacking Adversarially-Trained Models. Adversarial training has been shown to improve robustness to transfer attacks [43]. We evaluate the transferability of adversarial examples on adversarially trained destination networks. Even though the adversarial perturbations which we use to attack each adversarially-trained network are larger than the magnitude for which the destination networks are trained to be robust, the adversarial examples generated using non-robust networks do not transfer to the robust networks. However, as the robustness of the source network increases, the attack success rate increases substantially. Similarly to Figure 1, Figure 2 include the baseline performance of the attacks that would be measured if the images were unperturbed.

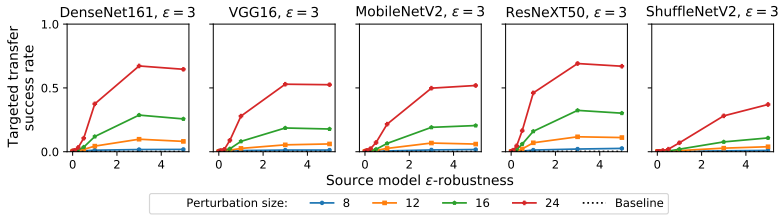

Figure 2: Targeted transfer attack success rate against destination ($\varepsilon = 3$)-robust ImageNet classifiers using ε-robust ResNet50 source networks with perturbation size $\|\delta\|_\infty \leq 16/255$. Higher is a more successful attack. Baseline refers the rate at which unperturbed images are classified as the target class. (Best viewed in color.)

Extending Our Methods to CIFAR-10. We confirm that our results are applicable under multiple datasets; we repeat many of our experiments for the CIFAR-10 dataset [37]. The results are consistent with our ImageNet results. We present and discuss these results in detail in the Appendix.

Computational Limitations. Due to the computational requirement of training multiple robust ImageNet classifiers, we restrict our experiments to those we can compute using pre-trained robust networks. Thus, we only test adversarial attacks with the ResNet50 architecture as a source network, and we only evaluate attacks on destination networks which are easily available to us. Similarly, we do not compute ensemble attacks using slightly-robust source networks despite the fact that we expect this technique to improve our attacks.

Universality of Slightly-Robust Features. Our results suggest that slightly-robust networks learn more *universal* features than non-robust networks [50, 67]—that is, slightly-robust networks learn features that are useful to every other network trained on the same task, regardless of architecture and exact training objective.

4 Hypothesis: Universality Explains Targeted Transferability

We have shown that we can construct *class-targeted* adversarial examples that transfer a specific (incorrectly) predicted class to destination networks. In this section, we aim to address an important question to help us answer *why* class-targeted adversarial examples transfer: *to what extent do class-targeted adversarial examples transfer the representation-layer features across networks?* One could imagine that class-targeted transferability arises from the overlap of a small number of especially vulnerable features where manipulation of these features can change the model classification, or, alternatively, the overlap of many features. To answer this question, we will construct and evaluate *representation-targeted* adversarial examples. We refer to the degree to which adversarial examples of a source model can analogously affect the features that are computed by the *representation layer* of the destination model as the *representation transferability* from source to destination. By contrast, *class transferability* (what is commonly referred to as just *transferability*) refers to the degree to which adversarial examples can analogously affect the output of the destination network.

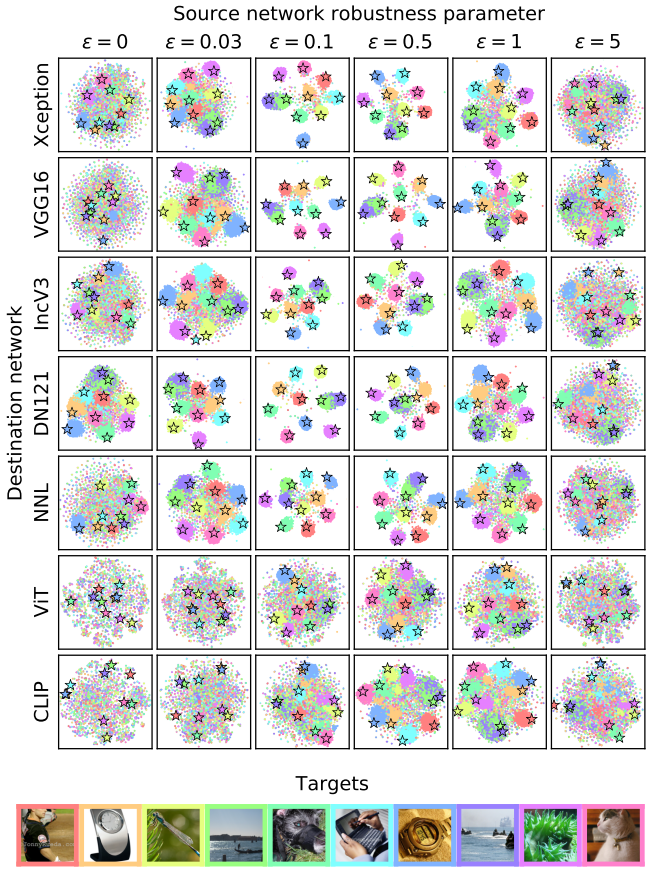

Figure 3: t-SNE plots of destination-network representations of representation-targeted adversarial examples generated by using whitebox ResNet50 models of specified ε-robustness. (Best viewed in color and magnified.)

Representation Transferability. Our goal is to study the representation transferability of source classifiers (with varying degrees of robustness) to non-robust models. We will show that slightly-robust networks exhibit a substantially higher degree of representation transferability than non-robust networks and more-robust networks. This can directly explain why class-targeted adversarial examples of slightly-robust networks are more transferable, as adversarial examples generated with slightly-robust networks will broadly transfer features, and will thus rely less on a small number of highly-vulnerable features that may not be present in every model. When a neural network has substantial representation transferability to *every* other network of a certain type, we say that the neural network has features that are *universal* with respect to these destination networks [50]. In this section, we show that slightly-robust networks have features that are universal with respect to the tested non-robust networks, which means that representation- and class-targeted adversarial examples generated with slightly-robust networks will often transfer even across a substantial difference in network architecture, such as the difference between CNNs and transformer networks.

Measuring Representation Transferability and Universality. To measure representation transferabililty, we rely on a simple but powerful test to assess the similarity between the representations produced by two different networks. Let x and y be two inputs that produce identical (or very similar) patterns of activation in the source network's representation (penultimate) layer. If the source network has a high degree of representation transferability to the destination network, then the responses to x and y will be very similar in the destination network as well. By contrast, if x and y do not share similar representations in the destination network, then the source network has a low degree of representation transferability. We measure the universality of source networks by testing the representation transferability to many destination networks: a source network which has a high degree of representation transferability to every tested non-robust network has learned universal features with respect to those networks.

This method allows us to test the representation transferability of a source network to a destination network by constructing images x and y with similar representations in a source network and measuring the similarity of the representations in the destination network. To construct these images, we select two images, x and y_0 from the ImageNet testset. We construct a representation-targeted adversarial example $y = y_0 + \delta$ targeting the representation of x. More precisely, we run the TMDI-FGSM algorithm to minimize the the ℓ_2 distance between the representations of x and y:

$$\delta = \underset{\|\delta\|_\infty \leq \varepsilon}{\arg\min} \|F(x) - F(y_0 + \delta)\|_2$$

where F represents the representation layer of the source network. Since we want to observe how well the representations transfer under the conditions of typical adversarial examples, we limit the perturbation to have an ℓ_∞ norm of ε, which, for this paper, we set to be the standard $16/255$. To limit the computational requirements of this experiment, we randomly select 10 images whose representations we use as targets $x$and 990 images to use as initial images y_0. Of these 1000 total images, no two share the same ImageNet class. For each source network, and for each initial image y_0, we construct ten representation-targeted adversarial examples—one for each target x—for a total of 9900 representation-targeted adversarial examples per source classifier. For each target x, the 990 representation-targeted adversarial examples y will have similar representations to x in the source network. By measuring the similarity of the (x, y) representations in the destination network, we can measure the representation transferability of the source network.

Table 2: Cosine similarity between representations (in the destination network) of representation-targeted adversarial examples y and the corresponding target image x, as a function of robustness parameter ε of the source network. Each value is an average over the 9900 (x, y) pairs of representation vectors.

	Source network robustness parameter (ε)									
Destination	0	0.01	0.03	0.05	0.1	0.25	0.5	1	3	5
Xception	0.462	0.505	0.531	0.563	**0.594**	0.585	0.572	0.543	0.449	0.404
VGG16	0.333	0.401	0.417	0.494	**0.528**	0.520	0.520	0.486	0.383	0.333
ResNet50V2	0.284	0.348	0.379	0.432	0.497	0.496	**0.510**	0.484	0.380	0.321
InceptionV3	0.577	0.612	0.627	0.644	**0.673**	0.662	0.655	0.636	0.572	0.539
MobileNetV2	0.431	0.459	0.460	0.493	**0.517**	0.513	0.513	0.504	0.455	0.425
DenseNet121	0.672	0.689	0.685	0.713	**0.726**	0.714	0.706	0.679	0.616	0.584
NasNetLarge	0.356	0.422	0.452	0.488	**0.541**	0.513	0.482	0.437	0.315	0.271
EfficientNetB4	0.085	0.111	0.137	0.144	**0.237**	0.220	0.226	0.202	0.112	0.074
ViT	0.066	0.087	0.109	0.129	0.195	**0.206**	**0.206**	0.203	0.120	0.086
CLIP	0.529	0.541	0.550	0.563	0.585	0.599	0.606	**0.613**	0.581	0.566

Here we present two different similarity metrics for comparing representations of x and y. First, we plot the t-distributed stochastic neighbor embedding (t-SNE) of the representation vectors of each representation-targeted adversarial example y in the destination network (Figure 3). Each color corresponds to one of the 10 target images (x). The 10 stars in each plot correspond to the t-SNE embedding of the destination-network representation of each target x. If the destination-network representation of each y associated with a particular color is near to its corresponding star, the representation transferability is high.

When representation transferability is low, the destination-network representations of representation-targeted adversarial examples that share the same initial image (y_0) should be similar. By contrast,

when representation-targeted adversarial examples transfer successfully, we observe clusters grouped by target image (in Figure 3, by color). The visual tightness of each cluster represents the similarity between the representations associated with each point and its neighbors. We find that across every tested destination model, clustering by target image is limited when the source model is not robust ($\varepsilon = 0$) in comparison to when the source model is slightly robust. The tightness of clustering by target image appears to peak, for convolutional models, when $\varepsilon \in \{0.1, 0.5\}$ which is coincident with the optimal source robustness for class-targeted transferability (Figure 1). For ViT and CLIP, the clustering by target image appears tightest when $\varepsilon \in \{0.5, 1\}$, which similarly occurs close to the robustness for optimal standard transferability (Figure 1). DenseNet, to which the non-robust source classifier exhibits the highest standard transferability (Figure 1), has the tightest clusters when the source classifier is non-robust. The presence of these clusters across every destination network when the source network is slightly-robust confirms that slightly-robust networks have a high degree of representation transferability to every tested non-robust model. In addition, we observe tight clusters by target image associated with the non-robust source models when the destination network has a DenseNet architecture. The representation transferability of the non-robust classifier to DenseNet can likely explain the effectiveness of class-targeted transferable adversarial examples to DenseNet. The fact that the representation-targeted adversarial examples are not as strongly clustered by target image for non-robust destination networks suggests that non-robust source classifiers do not transfer representations well, providing some evidence that the limited class-targeted transferability of non-robust source classifiers may arise from feature transfer of a small number of vulnerable features, as opposed to the feature transfer of many features. In addition, as robustness increases past the peak, the representation transferability to every destination network degrades. We hypothesize that as robustness increases, many smaller features on which non-robust classifiers rely are thrown out. Note that Figure 3 omits certain source and destination networks due to space constraints. We present the results for all combinations of source and destination networks in the Appendix.

As a second similarity metric, we report the mean cosine similarity of the representations (in destination networks) between (x, y) pairs, averaged over all 9900 such pairs for each source network (Table 2). This similarity peaks for destination CNNs when the source network has $\varepsilon = 0.1$, except for ResNet50v2, which peaks at $\varepsilon = 0.5$. Destination network ViT peaks at $\varepsilon \in \{0.25, 0.5\}$ and CLIP when $\varepsilon = 1$. These peaks are consistent with our t-SNE results (Figure 3), and are included primarily for the purpose of quantitatively confirming the visual results of our t-SNE plots.

In summary, slightly-robust source networks exhibit a high degree of representation transferability across *every* tested model. This suggests that slightly-robust source networks rely on universal features, i.e., features which are useful to *every* tested non-robust destination model, regardless of architecture, and, strikingly, useful even to CLIP, which is trained with a training objective that seeks to associate images to textual descriptions, rather than to a discrete set of labels. By contrast, while the non-robust source models have a high degree of representation transferability to a few architectures (most prominently, to DenseNet), non-robust source models fail to effectively transfer representations to many of the architectures, especially the transformer architectures, ViT and CLIP, and are thus not universal. The extent to which universal features exist in neural networks, as well as characterizing their behavior has recently been addressed in limited settings [36, 40, 50, 55], but is still very much an open area of research. Our results provide empirical evidence that universal features exist, at least among the tested non-robust networks and at the penultimate layer of the neural network. Furthermore, we demonstrate that networks can use adversarial training with a small robustness parameter to learn these universal features. We leave it to future work to validate whether similar universal features can be found at intermediate layers in the network.

5 Related Work

The vulnerability and defense of neural networks to adversarial examples have been studied extensively [1, 3, 5–8, 11, 18, 19, 21, 25, 35, 43, 46, 47, 52, 56, 59, 60, 68, 69, 75, 79].

First proposed by Goodfellow et al. [21], transferable adversarial examples are generally constructed by optimizing a perturbation to fool a white-box (source) network with hopes that it will transfer to black-box (destination) networks. Recent transferability research has focused on improving the optimizer to add generalization priors [38, 51, 84, 88]: researchers have added momentum to the gradient descent process [13], introduced transformations to the input [14, 63, 83], modified the adversarial loss functions [7, 39, 87], attacked intermediate feature representations [29, 31–33, 57, 88],

linearizing the source network [22], and additional manipulations to the gradient computation [81]. Additionally, some research has examined the role of source classifier(s): many of the aforementioned papers test the difference between source classifier architectures and researchers have proposed using generative networks [66, 86], and ensemble attacks [42, 73].

Robust networks have been shown to have a number of valuable properties, including serving as a good starting point for transfer learning [41, 58, 72, 76] and gradient interpretability [18]. The vulnerable features learned by neural networks have been studied both empirically [4, 15, 20, 30, 34, 44, 45, 49, 65, 67, 78, 80, 85] and theoretically [1, 2, 12, 26, 48, 61, 77, 82]. Similarly, there has been some research related to the universality hypothesis [36, 40, 50, 55], which speculates that under the right conditions, all neural networks may learn analogous features.

In prior work, the similarity of the feature spaces of different neural networks has been compared via linear transformations; however, these techniques are either anecdotal or do not account for possible non-linear relationship between neural networks [36, 40, 50, 55].

Our research draws inspiration from recent work [67] that proposes that slightly-robust features exhibit the universality principle, and presents a limited experiment that slightly-robust networks can be used to increase transferability. By contrast, our work provides a comprehensive study demonstrating that the technique can be used to achieve state-of-the-art transferability across a wide variety of architectures. In addition, we present a novel experiment to verify that universality is the underlying cause of this success.

6 Conclusion

We have demonstrated that slightly-robust networks learn features that can be exploited to construct highly transferable targeted adversarial examples. These adversarial examples, constructed with convolutional networks (ResNets), can attack other convolutional networks with state-of-the-art performance, as well as networks with substantially different architectures, such as the transformer-based network ViT, and with different learning objective, such as CLIP. In fact, this work is the first we are aware of that constructs targeted transferable attacks against transformer-based networks such as ViT and CLIP. We propose that the effectiveness of slightly-robust networks can be explained by the universality of the features learned by these networks. We find this to be true by showing that representation-targeted adversarial attacks generated with slightly-robust networks are highly transferable, even to networks such as ViT and CLIP. Furthermore, since most previous transferable adversarial generation techniques rely on optimizing adversarial examples over a non-robust source network, our technique can be combined with virtually any previously existing optimization technique by replacing the non-robust source network with a slightly-robust network.

As discussed, our paper is important to the field of adversarial machine learning, as we improve transfer attacks and study the mechanism of adversarial examples. More generally, our paper reveals a phenomenon that is significant for the broader field of deep learning: we find that different non-robust networks, even when trained with similar convolutional architectures, do not necessarily have many features that substantially overlap. This can have important implications for the reliability of neural networks; when different networks rely on different features, they are susceptible to different types of errors. In addition, we present an argument that, for a given task, there are features that are useful to every tested neural network, and that these features can be learned with small-ε adversarial training, even when the source network architecture and learning objective are dissimilar to those of the destination network. Thus, by studying the features of a single slightly-robust network, we can empirically discover properties that will be applicable across all non-robust networks. We speculate that this phenomenon can explain why slightly-robust networks are successful at transfer-learning tasks [41, 58, 72, 76]. With applications across the field of machine learning, we expect that our contributions in this paper provide an important stepping stone towards discovering a general understanding of the features learned by neural networks.

Of course, our research has potential negative implications. Firstly, we propose a method to improve the generation of targeted transferable adversarial examples. While we hope that our research leads to the development of more robust and interpretable machine learning systems, in principle, an adversary could use our technique to attack existing systems. Secondly, we advocate for the use of adversarial training, which can be computationally intensive and could lead to excessive energy consumption.

References

[1] Z. Allen-Zhu and Y. Li. Feature purification: How adversarial training performs robust deep learning. *arXiv preprint arXiv:2005.10190*, 2020.

[2] D. Arpit, S. Jastrzębski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, Y. Bengio, et al. A closer look at memorization in deep networks. *arXiv preprint arXiv:1706.05394*, 2017.

[3] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018.

[4] M. Aubry and B. C. Russell. Understanding deep features with computer-generated imagery. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2875–2883, 2015.

[5] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli. Evasion attacks against machine learning at test time. In *Joint European conference on machine learning and knowledge discovery in databases*, pages 387–402. Springer, 2013.

[6] N. Carlini and D. Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14, 2017.

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

[8] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

[9] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

[10] F. Chollet et al. Keras. `https://keras.io`, 2015.

[11] J. Cohen, E. Rosenfeld, and Z. Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning*, pages 1310–1320. PMLR, 2019.

[12] G. De Palma, B. Kiani, and S. Lloyd. Random deep neural networks are biased towards simple functions. In *Advances in Neural Information Processing Systems*, pages 1964–1976, 2019.

[13] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

[14] Y. Dong, T. Pang, H. Su, and J. Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019.

[15] A. Dosovitskiy and T. Brox. Inverting visual representations with convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4829–4837, 2016.

[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[17] L. Engstrom, A. Ilyas, H. Salman, S. Santurkar, and D. Tsipras. Robustness (python library), 2019. URL `https://github.com/MadryLab/robustness`.

[18] L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, B. Tran, and A. Madry. Adversarial robustness as a prior for learned representations. *arXiv preprint arXiv:1906.00945*, 2019.

[19] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017.

[20] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann. Shortcut learning in deep neural networks. *arXiv preprint arXiv:2004.07780*, 2020.

[21] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[22] Y. Guo, Q. Li, and H. Chen. Backpropagating linearly improves transferability of adversarial examples. *arXiv preprint arXiv:2012.03528*, 2020.

[23] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[24] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.

[25] D. Hendrycks, K. Zhao, S. Basart, J. Steinhardt, and D. Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019.

[26] K. Hermann, T. Chen, and S. Kornblith. The origins and prevalence of texture bias in convolutional neural networks. *Advances in Neural Information Processing Systems*, 33, 2020.

[27] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.

[28] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[29] Q. Huang, I. Katsman, H. He, Z. Gu, S. Belongie, and S.-N. Lim. Enhancing adversarial example transferability with an intermediate level attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4733–4742, 2019.

[30] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Adversarial examples are not bugs, they are features. In *Advances in Neural Information Processing Systems*, pages 125–136, 2019.

[31] N. Inkawhich, W. Wen, H. H. Li, and Y. 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.

[32] N. Inkawhich, K. J. Liang, L. Carin, and Y. Chen. Transferable perturbations of deep feature distributions. *arXiv preprint arXiv:2004.12519*, 2020.

[33] N. Inkawhich, K. J. Liang, B. Wang, M. Inkawhich, L. Carin, and Y. Chen. Perturbing across the feature hierarchy to improve standard and strict blackbox attack transferability. *arXiv preprint arXiv:2004.14861*, 2020.

[34] J. Jo and Y. Bengio. Measuring the tendency of cnns to learn surface statistical regularities. *arXiv preprint arXiv:1711.11561*, 2017.

[35] S. Kaur, J. Cohen, and Z. C. Lipton. Are perceptually-aligned gradients a general property of robust classifiers? *arXiv preprint arXiv:1910.08640*, 2019.

[36] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

[37] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[38] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016.

[39] M. Li, C. Deng, T. Li, J. Yan, X. Gao, and H. Huang. Towards transferable targeted attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 641–649, 2020.

[40] Y. Li, J. Yosinski, J. Clune, H. Lipson, and J. E. Hopcroft. Convergent learning: Do different neural networks learn the same representations? In *FE@ NIPS*, pages 196–212, 2015.

[41] K. Liang, J. Y. Zhang, O. Koyejo, and B. Li. Does adversarial transferability indicate knowledge transferability? *arXiv preprint arXiv:2006.14512*, 2020.

[42] Y. Liu, X. Chen, C. Liu, and D. Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

[43] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[44] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

[45] R. T. McCoy, E. Pavlick, and T. Linzen. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. *arXiv preprint arXiv:1902.01007*, 2019.

[46] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff. On detecting adversarial perturbations. *arXiv preprint arXiv:1702.04267*, 2017.

[47] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. 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, 2016.

[48] P. Nakkiran, D. Kalimeris, G. Kaplun, B. Edelman, T. Yang, B. Barak, and H. Zhang. Sgd on neural networks learns functions of increasing complexity. In *Advances in Neural Information Processing Systems*, pages 3496–3506, 2019.

[49] C. Olah, A. Mordvintsev, and L. Schubert. Feature visualization. *Distill*, 2(11):e7, 2017.

[50] C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. https://distill.pub/2020/circuits/zoom-in.

[51] N. Papernot, P. McDaniel, and I. Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

[52] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[53] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`.

[54] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al. Learning transferable visual models from natural language supervision. *arXiv preprint arXiv:2103.00020*, 2021.

[55] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. *arXiv preprint arXiv:1706.05806*, 2017.

[56] A. Raghunathan, J. Steinhardt, and P. Liang. Certified defenses against adversarial examples. *arXiv preprint arXiv:1801.09344*, 2018.

[57] A. Rozsa, M. Günther, and T. E. Boult. Lots about attacking deep features. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 168–176. IEEE, 2017.

[58] H. Salman, A. Ilyas, L. Engstrom, A. Kapoor, and A. Madry. Do adversarially robust imagenet models transfer better? *arXiv preprint arXiv:2007.08489*, 2020.

[59] S. Santurkar, A. Ilyas, D. Tsipras, L. Engstrom, B. Tran, and A. Madry. Image synthesis with a single (robust) classifier. In *Advances in Neural Information Processing Systems*, pages 1262–1273, 2019.

[60] A. Shafahi, W. R. Huang, C. Studer, S. Feizi, and T. Goldstein. Are adversarial examples inevitable? *arXiv preprint arXiv:1809.02104*, 2018.

[61] H. Shah, K. Tamuly, A. Raghunathan, P. Jain, and P. Netrapalli. The pitfalls of simplicity bias in neural networks. *arXiv preprint arXiv:2006.07710*, 2020.

[62] R. Shao, Z. Shi, J. Yi, P.-Y. Chen, and C.-J. Hsieh. On the adversarial robustness of visual transformers. *arXiv preprint arXiv:2103.15670*, 2021.

[63] Y. Sharma, G. W. Ding, and M. Brubaker. On the effectiveness of low frequency perturbations. *arXiv preprint arXiv:1903.00073*, 2019.

[64] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[65] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[66] Y. Song, R. Shu, N. Kushman, and S. Ermon. Constructing unrestricted adversarial examples with generative models. *arXiv preprint arXiv:1805.07894*, 2018.

[67] J. M. Springer, M. Mitchell, and G. T. Kenyon. Adversarial perturbations are not so weird: Entanglement of robust and non-robust features in neural network classifiers. *arXiv preprint arXiv:2102.05110*, 2021.

[68] D. Stutz, M. Hein, and B. Schiele. Disentangling adversarial robustness and generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6976–6987, 2019.

[69] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations, ICLR 2014*, 2014.

[70] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[71] M. Tan and Q. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR, 2019.

[72] M. Terzi, A. Achille, M. Maggipinto, and G. A. Susto. Adversarial training reduces information and improves transferability. *arXiv preprint arXiv:2007.11259*, 2020.

[73] F. Tramèr, D. Boneh, A. Kurakin, I. Goodfellow, N. Papernot, and P. McDaniel. Ensemble adversarial training: Attacks and defenses. In *6th International Conference on Learning Representations, ICLR 2018-Conference Track Proceedings*, 2018.

[74] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry. Robustness may be at odds with accuracy. In *International Conference on Learning Representations*, 2019.

[75] J. Uesato, B. O'donoghue, P. Kohli, and A. Oord. Adversarial risk and the dangers of evaluating against weak attacks. In *International Conference on Machine Learning*, pages 5025–5034. PMLR, 2018.

[76] F. Utrera, E. Kravitz, N. B. Erichson, R. Khanna, and M. W. Mahoney. Adversarially-trained deep nets transfer better. *arXiv preprint arXiv:2007.05869*, 2020.

[77] G. Valle-Pérez, C. Q. Camargo, and A. A. Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. *arXiv preprint arXiv:1805.08522*, 2018.

[78] H. Wang, X. Wu, Z. Huang, and E. P. Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8684–8694, 2020.

[79] D. Warde-Farley and I. Goodfellow. Adversarial perturbations of deep neural networks. *Perturbations, Optimization, and Statistics*, 311, 2016.

[80] K.-A. A. Wei. *Understanding non-robust features in image classification*. PhD thesis, Massachusetts Institute of Technology, 2020.

[81] D. Wu, Y. Wang, S.-T. Xia, J. Bailey, and X. Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. *arXiv preprint arXiv:2002.05990*, 2020.

[82] L. Wu, Z. Zhu, et al. Towards understanding generalization of deep learning: Perspective of loss landscapes. *arXiv preprint arXiv:1706.10239*, 2017.

[83] C. Xie, Z. Zhang, Y. Zhou, S. Bai, J. Wang, Z. Ren, and A. L. Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019.

[84] Z. Yang, L. Li, X. Xu, S. Zuo, Q. Chen, B. Rubinstein, C. Zhang, and B. Li. Trs: Transferability reduced ensemble via encouraging gradient diversity and model smoothness. *arXiv preprint arXiv:2104.00671*, 2021.

[85] Q.-s. Zhang and S.-C. Zhu. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering*, 19(1):27–39, 2018.

[86] Z. Zhao, D. Dua, and S. Singh. Generating natural adversarial examples. *arXiv preprint arXiv:1710.11342*, 2017.

[87] Z. Zhao, Z. Liu, and M. Larson. On success and simplicity: A second look at transferable targeted attacks. *arXiv preprint arXiv:2012.11207*, 2020.

[88] W. Zhou, X. Hou, Y. Chen, M. Tang, X. Huang, X. Gan, and Y. Yang. Transferable adversarial perturbations. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–467, 2018.

[89] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le. Learning transferable architectures for scalable image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8697–8710, 2018.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]

2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] See supplemental material
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] See supplemental material
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We are unable to train additional adversarially-trained models due to computational limitations.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] See supplemental material
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you mention the license of the assets? [Yes]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]