EARLY STOP AND ADVERSARIAL TRAINING YIELD BETTER SURROGATE MODEL: VERY NON-ROBUST FEATURES HARM ADVERSARIAL TRANSFERABILITY

Anonymous authors
Paper under double-blind review

ABSTRACT

The transferability of adversarial examples (AE); known as adversarial transferability, has attracted significant attention because it can be exploited for Transferable Black-box Attacks (TBA). Most lines of works attribute the existence of the non-robust features improves the adversarial transferability. As a motivating example, we test the adversarial transferability on the early stopped surrogate models, which are known to be concentrated on robust features than non-robust features from prior works. We find that the early stopped models yield better adversarial transferability than the models at the final epoch, which leaves non-intuitive interpretation from the perspective of the robust and non-robust features (NRFs). In this work, we articulate a novel Very Non-Robust Feature(VNRF) hypothesis that the VNRFs learned can harm the adversarial transferability to explain this phenomenon. This hypothesis is partly verified through zero-outing some filters with high l_1 norm values. This insight further motivates us to adopt *light adversarial training* that mainly removes the VNRFs for significantly improving the transferability.

1 INTRODUCTION

Deep neural networks (DNNs) are widely known to be vulnerable to adversarial examples (AE) (Szegedy et al., 2013; Goodfellow et al., 2015), i.e. images perturbed by imperceptible noise fooling the network. Specifically, the transferability of AE has drawn significant attention because such property can be used for achieving a Transferable Black-box Attack (TBA). Given full access to the architecture and parameters of the target model, simple I-FGSM (Kurakin et al., 2017) or PGD (Madry et al., 2018) has been proven to achieve sufficiently high attack success even for adversarially trained models. However, TBA does not require access to the target model, thus it constitutes a more practical threat in security-sensitive applications. Arguably, TBA is also more practical than another variant of black-box attack that allows numerous forward query accesses to the target model. A TBA pipeline typically involves two stages: (i) training a surrogate model the same or similar training dataset; (ii) generating transferable AE on the surrogate model.

In the TBA community, the research on the second stage is like a hot-spot, while the first stage remains like a blind spot that attracts very limited attention so far. Though the exact underpinnings of adversarial transferability are not fully understood, numerous works have investigated transferability from various perspectives, but mainly or exclusively focus on the second stage. For example, increasing input diversity(DI^2-FGSM) (Xie et al., 2019) and post-processing the input gradient with momentum (MI-FGSM) (Dong et al., 2018) or smoothing kernel (TI-FGSM) (Dong et al., 2019) improve the transferability. On the contrary, the influence of the first stage in TBA is limited to comparing the different architectures. For example, models with similar architectures are often found to transfer better between each other.

With surrogate models of different architectures, Wu et al. claim that high-accuracy models tend to exhibit stronger adversarial transferability since the decision boundaries of high-accuracy models should be similar. Our work revisits the relationship between surrogate model accuracy and transferability. In contrast to Wu et al. (2018), we fix the same architecture but evaluate the adversarial transferability of the surrogate model saved at different checkpoints. Counter-intuitively, we find that early stop leads to lower model accuracy but it improves the transferability by a non-trivial margin. More

interestingly, we note that this phenomenon also, at first sight, seems to contradict existing claims that explain from the non-robust feature(NRF) perspective. Specifically, It is found in (Ilyas et al., 2019; Nitin, 2021) that adversarial transferability can be attributed to mainly or exclusively non-robust features. Longer training is expected to be beneficial for transferability given that the model learns more NRFs in the latter stage of training (Benz et al., 2020; Nitin, 2021). However, our results show a misaligned phenomenon that after certain epochs, longer training decreases the transferability. This confusion motivates us to revisit the feature perspective on adversarial transferability. Actually, the above confusion can be cleared if the following bold hypothesis statement is true.

VNRF Hypothesis on Transferability. *Even though Non-Robust Features (NRFs) are the main, or exclusive, cause of adversarial transferability, the very non-robust features (VNRFs) are actually harmful to the transferability.*

If true, the above VNRF hypothesis makes the existing feature perspective on transferability more well-rounded. We verify the above hypothesis by identifying those VNRFs in the model by checking the L_1 norm of the convolution filter weights and demonstrate that zeroing out those VNRFs increases the transferability.

Since NRFs are the cause of transferability while VNRFs are harmful to the transferability, it is naturally conjectured that light adversarial training with a very small perturbation budget allows the model to still learn NRFs while excluding VNRFs might be more beneficial for improving the transferability. Note that heavy adversarial training further excludes the general NRFs from the surrogate model, which is expected to reduce the performance. We empirically verify the above conjecture and expectation throuth extensive experiments.

To this end, our work provides both new conceptual insight and strong empirical results for understanding and improving adversarial transferability. Our contributions are summarized as follows.

- Complementary to the existing NRF perspective on transferability, our work provides **new conceptual insight** that VNRFs are harmful to the transferability. This verified new insight well explains the phenomenon that is not readily explainable by the NRF perspective.
- In contrast to numerous works that exclusively study the second stage of TBA, our work fills the gap to investigate its first stage that attracts little attention so far. Our work provides **strong empirical results** for improving the transferability. For example, on ImageNet, switching the surrogate model training procedure from standard training to adversarial training improves transfer rate from 19.8% to higher than 91.8% (from ResNet50 to ViT-B/16. In the more challenging targeted setting, the average transfer rate is increased from 0.4% to 38.4% (from ResNet18 to a wide range of CNN target models).

2 Background & Setup

2.1 Preliminary Knowledge

Adversarial Examples. Adversarial examples $\tilde{x} = x + \delta$ consist of a (clean) sample and a specially crafted small adversarial perturbation δ, and has the objective to fool a classifier, i.e. $f(\tilde{x}) \neq y$. To ensure visual imperceptibility, adversarial perturbations are commonly constrained to be smaller than a certain magnitude ϵ, i.e. $||\delta||_p \leq \epsilon$, where the l_p-norm is a common choice, here indicated as $||\cdot||_p$. A common choice for the objective function for (non-targeted) adversarial examples is to adopt the loss function used for model training, but with an opposite sign. Targeted adversarial perturbations have the objective to fool a model towards a specific target class y_t. Targeted adversarial examples are only considered successful, if a misclassification toward the target class is achieved, i.e. $f(\tilde{x}) = y_t$.

Adversarial Training. Incorporating adversarial examples into the training process is an effective method to defend against adversarial examples and is known under adversarial training (Goodfellow et al., 2015; Madry et al., 2018). The FGSM attack(Goodfellow et al., 2015) was suggested on the fly during the training process to robustify a model. (Madry et al., 2018) posed adversarial training as a min-max optimization problem $\arg\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}}[\max_{\delta\sim\Delta} \mathcal{L}(\theta, x+\delta, y)]$, where Δ indicates the set of possible adversarial perturbations. In the above formula, the inner optimization maximizes the loss to find an adversarial perturbation, and the outer part optimizes the model weights to minimize the loss. For the inner optimization, any adversarial attack technique can be chosen, however, the multi-step projected gradient descent (PGD) is one of the most common choices.

Adversarial Transferability. Despite the capability of adversarial examples to fool a network, adversarial examples do further exhibit a transferability property. (Szegedy et al., 2013) first demonstrated such cross model generalization, where a large fraction of adversarial example generated on one (surrogate) model, here indicated as f_s, will be misclassified by other (target) networks, here indicated as f_t. Possible differences between the target and surrogate models include a different model architecture, training paradigm, initialization, etc. Initially the transferability was investigated in the untargeted setting, i.e. $f_t(x + \delta_s) \neq y$, where the adversarial perturbation δ_s was generated for surrogate network s. Later, also the relatively more difficult targeted transferability has been investigated, i.e. $f_t(x + \delta_s) = y_t$.

2.2 Experimental Setup

TBA First Stage. Our work mainly investigates two training factors: early stop and adversarial training. For investigating early stops, we need to train a model from scratch and save the model parameters at each epoch. We perform experiments on both ImageNet and CIFAR10. On ImageNet, we train the model for 100 epochs with the initial learning rate of 0.1 which is decreased by 10 at the epoch of 30, 60, and 90. Since it is resource-intensive to perform experiments on ImageNet, we choose to adopt a relatively light ResNet18 as the surrogate model architecture. On CIFAR-10, we train the model for 150 epochs with the initial learning rate of 0.1 which is decreases at the epoch of 50 and 100. For investigating the effect of adversarial training, we mainly borrow the models pretrained on ImageNet in (Salman et al., 2020), where they provide robust models adversarially trained with l_2-norm or l_∞-norm bounded perturbations. Additionally, we also use the models pretrained in (Wong et al., 2020) for investigating the effect of FGSM adversarial training.

TBA Second Stage. Following (Dong et al., 2018; 2019; Li et al., 2020a), we use the dataset introduced in the NeurIPS 2017 adversarial challenge[1] for evaluating the transferability performance on ImageNet. This dataset is ImageNet-compatible and composed of 1000 images from each class. For CIFAR10, we use its validation dataset. If not otherwise indicated, for the generating adversarial examples, we set the number of iterations (T) to 20 with a step size of 2/255. Following previous conventions, the maximum allowed perturbation magnitude L_∞ is set to ϵ=16/255. We follow the hyper-parameter settings of previous works and set the momentum to $\mu = 1$ for MI-FGSM as in (Dong et al., 2018). For DI-FGSM (Xie et al., 2019) we set the probability of the stochastic input diversity to $p = 0.7$. For attacks using TI-FGSM (Dong et al., 2019) we adopt a kernel length of 5 as suggested by (Li et al., 2019). We further deploy a strong baseline attack, for which we combine the previously proposed attack MI, DI, and TI-FGSM, which we indicate as MI+DI+TI.

3 Motivation

3.1 Do accurate models transfer better?

An early work (Wu et al., 2018) has concluded that models with higher accuracy are better surrogate models via comparing surrogate models with different architectures. Excluding the influence of architecture, we intend to compare the same surrogate model trained with different epochs. Modern DNNs are typically trained with stage-wise decreasing learning rates, where the accuracy increases in the whole training process, especially with an immediate jump after the learning rate decreases. By default, we adopt other well-trained models as the target model. With their conjecture (Wu et al., 2018) that high-accuracy models learn similar decision boundaries and transfer better to each other, it might be expected that the surrogate trained with higher accuracy achieves better transferability performance. Evaluating with a wide range of model architectures on both CIFAR10 (Krizhevsky et al., 2010) and ImageNet (Russakovsky et al., 2015), we find that the transferability performance indeed increases in the early stage. However, after the learning decrease at first time, there is an overall trend that the transferability decreases when the surrogate model is trained with more epochs, suggesting that higher accuracy does not necessarily indicate higher transferability. The results are detailed in the following.

Early Stop Improves Adversarial Transferability. Here, we investigate the transferability capabilities of a surrogate model saved at intermediate epochs on CIFAR10. We adopt ResNet18 as the surrogate model and evaluate the transferability on multiple well-trained models, including ResNet50, VGG16, DenseNet121. We refer to a well-trained model as a model with a sufficient number of training epochs to guarantee high performance on the validation dataset. The results are presented in

[1] https://github.com/rwightman/pytorch-nips2017-adversarial

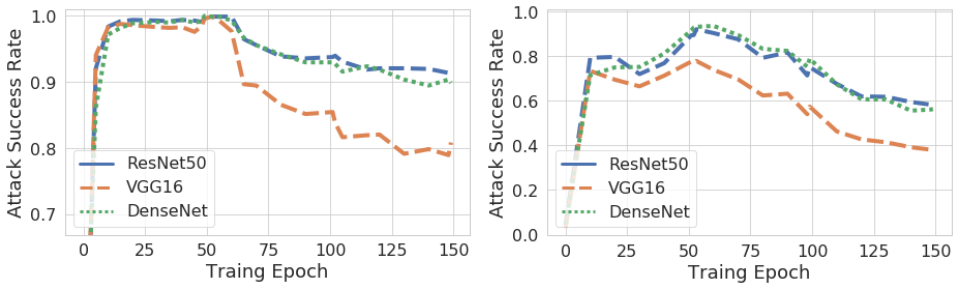

Figure 1: Transferability evaluation of adversarial examples generated with ResNet18 as the surrogate saved at different epochs in untargeted (left) and targeted(right) settings on CIFAR10 dataset.

Figure 1 for both non-targeted and targeted settings. We observe that the transferability performance increases significantly in the first few epochs and then saturates at a high plateau. After the first learning rate decrease at epoch 50, there is an overall trend that the attack success rate on multiple target models decreases. In the targeted setting, a similar trend can be observed. This observation might appear counter-intuitive since models trained with fewer epochs exhibit a lower test accuracy, and it might be tempting to jump to the conclusion that fully-trained models will also exhibit superior transferability capabilities, which might be the reason why existing works mainly adopt a fully-trained model as the surrogate.

Misalignment between existing NRF perspective and observed phenomenon. Ilyas et al. (Ilyas et al., 2019) have found that adversarial transferability arises when models learn similar brittle NRFs of the underlying dataset. One recent work (Nitin, 2021) claims that *"adversarial examples transfer if and only if they exploit predictive NRFs."*. Clearly, adversarial transferability can be attributed to mainly or exclusively NRFs. On the other hand, two works (Benz et al., 2020; Nitin, 2021) have independently shown that the DNNs first learn RFs before NRFs. The reason has been attributed to that the RFs are easier to learn and are important for the network stability in the early stage (Benz et al., 2020). Their finding echos well with the previous finding that DNN learns high-frequency components in the later stage because high-frequency functions converge much slower (Xu et al., 2019; Basri et al., 2019). As training goes on, the feature robustness is expected to decrease, i.e. the model depends more and more on the NRFs. Overall, longer training, which is expected to decrease the model robustness with more NRFs (Benz et al., 2020), *is supposed to* yield a more transferable surrogate model since NRFs account for transferability (Ilyas et al., 2019; Nitin, 2021). However, our results show a misaligned phenomenon that after a certain number of epochs, longer training actually decreases the transferability. Towards explaining the observed phenomenon, we propose VNRF hypothesis that VNRFs can be harmful to adversarial transferability.

3.2 On the influence of VNRFs on adversarial transferability

Ilyas et al. introduce robust features (RFs) and non-robust features (NRFs) which are widely used for understanding adversarial examples and their transferability. In the following, we summarize the definitions of RFs/NRFs and define very non-robust features (VNRFs):

- *RFs* vs*NRFs:* a feature f is robust if a $\gamma > 0$ exists for it to be γ-robustly useful given a certain perturbation budget δ, i.e. $\mathbb{E}_{(\mathrm{x},\mathrm{y})\sim\mathcal{D}}[\inf_{||\delta||\leq\epsilon} y \cdot f(x+\delta)] \geq \gamma$. A feature is NRF when such $\gamma > 0$ does not exist.
- *VNRFs:* Assuming the perturbation budget adopted above is δ_a, a feature is very non-robust if $\gamma > 0$ does not exist even if the perturbation is set to a sufficiently small δ_b, *i.e* $\epsilon_b << \epsilon_a$.

Straightforwardly, VNRF can be referred to the features that are non-robust even under only a small amount of perturbations. The aforementioned VNRF hypothesis naturally explains the above misalignment. Specifically, in the later stage of training, the model is increasingly dependent on the VNRFs, resulting in a lower transferability. In the following, we will discuss the intuition behind this hypothesis and design a toy example to verify this hypothesis.

Intuition behind VNRF hypothesis. We believe there is an opposite correlation between the feature transferability and readiness to be exploited for the attack. Given a dataset that has a set of features with a wide range of robustness, two models are independently trained on the model. We interpret that those features with high robustness tend to be learned by both of the two models, *i.e* they share high overlap between two independent models because RFs are stable and easier to be learned. This interpretation aligns with the finding that the model learns RFs first (Benz et al., 2020; Nitin, 2021). However, those RFs can not be readily exploited for the attack because NRFs are the cause of vulnerability (Ilyas et al., 2019). On the other hand, those VNRFs can be easily exploited for a successful attack on the white-box surrogate model. However, those VNRFs tend to have low overlap between different models. From the optimization point of view, the adversarial examples mainly exploiting the VNRFs are likely to over-fit to the surrogate model and have a low transfer rate. Thus, RFs nor VNRFs are both not beneficial for achieving transferable attacks, and only those NRFs that are not very non-robust exploit are located at a sweet spot. We highlight that RFs do not harm transferability, while VNRFs tend to harm transferability. The reason lies in that RFs are less exploited in the generation of adversarial examples, while VNRFs are readily exploited. Given a fixed perturbation budget, if the adversarial examples are overly dependent on the VNRFs, the exploited transferable NRFs decrease.

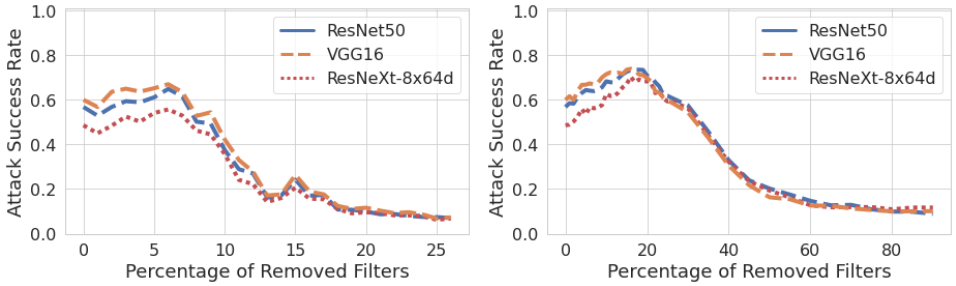

Figure 2: Transferability evaluation of adversarial examples generated with ResNet18 as the surrogate model deleted at every convolution layer(left) and last layer(right), on CIFAR10 dataset with targeted I-FGSM attack.

Verifying the VNRF hypothesis via weight filtering. Our hypothesis-verifying experiment is mainly inspired by (Borkar et al., 2020). Formally, let $\phi_m(u)$ be the output of the mth convolutional filter with weight W_m. The additive perturbation on the activation map $\phi(x)$, denoted as $e_m = \phi(x+r) - \phi(x)$, is caused by applying an additive perturbation r to the input x. The e_m is derived to be bounded as $||e_m||_\infty \leq ||W_m||_1 ||r||_p$ (Equation 2 of (Borkar et al., 2020), applying l_p-norm). In other words, the $||W_m||_1$ can then be the hint for the upper bound of $||e_m||$. The l_1-norm of the convolutional filters can then be used to identify and rank the filter activations regarding their sensitivity to the perturbation on the input. As a consequence, (Borkar et al., 2020) identifies that those filter weights with a small l_1-norm are not sensitive to the additive input perturbation. Empirically, (Borkar et al., 2020) has shown that only retraining those filter weights with a large l_1-norm is sufficient for increasing the model robustness. From the feature robustness perspective, straightforwardly, we can perceive those filter weights with a small (large) l_1-norm as weights representing RFs (NRFs). It is reasonable to conjecture that VNRFs can arise from the convolution filters with very high $||W_m||_1$, for which a very small input perturbation can lead to a large output change on the intermediate activation maps, and thus cause a substantial change in the final logits. To empirically verify the VNRF hypothesis, for each convolutional layer we rank the filters through their l_1-norm and zero out the filters with a large l_1-norm, and the results are shown in Figure 2. We apply the strategy in two ways, for all convolution layers in the whole network (left) and only at the final residual block (right) based on the surrogate model ResNet18. We conduct experiments on CIFAR-10 dataset with I-FGSM for 3 different target models(ResNet50, VGG16, ResNeXt-8x64d). We find that in both setups, zero-outing a certain percentage of filters improves the transferability by a visible margin. By perceiving those filters with very large l_1-norm, the results empirically verify that directly removing the VNRFs via filter removing increases boost the transferability. Note that when the percentage zero-out gets sufficiently large, the transferability is expected to decrease because it starts to remove the transferable NRFs.

4 SIMPLE YET EFFECTIVE TECHNIQUES IN THE FIRST STAGE OF TBA

Our above analysis suggests that only NRFs that are not very non-robust are beneficial for the TBA task, while VNRFs need to be removed due to its readiness to be exploited by the attack yet having low transferability (overlap) between models. To this end, we propose two simple techniques in the first stage of TBA: **early stop** and **light adversarial training**.

4.1 EARLY STOP FOR IMPROVING TRANSFERABILITY

Early stop can be a simple technique for improving transferability. The results on CIFAR10 is shown in Table 1, where we observe that early stop with vanilla I-FGSM is sufficient to achieve an average attack success rate of more than 98% for the surrogate model VGG16 and up to 99.9% for the surrogate model ResNet18. The results in ImageNet are also shown in Table 1, where early stop also non-trivially improves the transferability on both ResNet and VGG16.

Table 1: Early stop for improving the transferability on CIFAR-10 and ImageNet. Each entry indicates the transfer rate without / with early stop.

Dataset	Surrogate	Attack	ResNet50	DenseNet121	VGG19	WideResNet28-10	Average
CIFAR-10	ResNet18	I	90.8 / **100**	89.9 / **99.7**	80.9 / **99.9**	93.8 / **100.0**	88.8 / **99.9**
		MI	96.1 / **100.0**	93.9 / **99.7**	90.9 / **100.0**	98.2 / **100.0**	94.8 / **99.9**
		DI	98.2 / **100.0**	96.3 / **99.7**	97.3 / **100.0**	97.1 / **100.0**	97.2 / **99.9**
		TI	97.1 / **99.9**	92.9 / **99.4**	94.1 / **100.0**	94.5 / **99.7**	94.7 / **99.8**
		MI+DI+TI	97.9 / **100.0**	95.5 / **99.5**	93.6 / **99.8**	98.4 / **100.0**	96.35 / **99.8**
	VGG16	I	89.0 / **99.1**	86.0 / **95.9**	89.5 / **99.3**	88.9 / **98.1**	88.3 / **98.1**
		MI	93.5 / **98.0**	92.3 / **94.9**	94.1 / **99.0**	93.2 / **96.7**	93.3 / **97.1**
		DI	92.9 / **99.5**	92.4 / **98.1**	94.5 / **99.5**	93.7 / **99.2**	93.4 / **99.1**
		TI	86.5 / **99.0**	83.7 / **96.6**	86.6 / **98.9**	86.1 / **97.9**	85.7 / **98.1**
		MI+DI+TI	96.5 / **98.9**	94.1 / **96.0**	97.6 / **98.5**	96.8 / **97.0**	96.3 / **97.6**
ImageNet	ResNet18	I	79.2 / **84.0**	71.9 / **80.8**	75.9 / **78.0**	78.6 / **84.5**	76.4 / **81.8**
		MI	86.1 / **93.2**	82.5 / **90.6**	83.0 / **88.8**	85.6 / **91.4**	84.3 / **91.0**
		DI	96.4 / **97.5**	94.5 / **96.6**	95.8 / **96.4**	95.2 / **97.1**	95.5 / **96.9**
		TI	83.7 / **85.3**	77.0 / **83.2**	78.4 / **81.0**	82.3 / **86.3**	80.3 / **84.0**
		MI+DI+TI	98.1 / **98.5**	98.1 / **98.7**	98.1 / **98.3**	97.8 / **98.8**	98.0 / **98.5**
	VGG16	I	56.7/**65.9**	43.3/ **53.2**	97.8 / **98.9**	57.2 / **65.5**	63.8 / **70.9**
		MI	71.7/ **81.0**	61.7 / **68.9**	97.9 / **99.3**	69.8 / **76.2**	75.3 / **81.3**
		DI	68.7 / **80.5**	63.8 / **71.5**	99.5 / **99.8**	70.9 / **79.9**	75.7 / **82.9**
		TI	61.7 / **74.2**	55.5 / **64.4**	97.8 / **99.0**	67.1 / **75.3**	70.5 / **78.2**
		MI+DI+TI	86.4 / **91.8**	82.2 / **87.1**	99.6 / **99.8**	88.1 / **91.1**	89.1 / **92.5**

4.2 LIGHT ADVERSARIAL TRAINING FOR IMPROVING TRANSFERABILITY

Our above results show that early stop can significantly increase the transferability, while the performance boost on ImageNet is relatively less satisfactory. The reason can be attributed to that early stop forcing the model to be less dependent on VNRFs also inevitably prevent the model from learning more transferable NRFs. Motivated by this, we further investigate light adversarial training as an advanced alternative since adversarial training explicitly guides the model to not learn VNRFs. We select a standard ($\epsilon_s = 0$) trained and 9 ℓ_2 adversarially trained variants of ResNet18 and ResNet50 with different ϵ_s as the surrogate model and perform 5 FGSM-based attacks on them. We evaluate the transferability on 4 naturally trained target models ($\epsilon_t = 0$) (ResNet, DenseNet121, VGG16, MobileNetV2). For the ResNet, it is set to ResNet18 when ResNet50 is the surrogate model and set to Resnet50 when the surrogate model is ResNet18. We also test the adversarial transferability on two non-CNNs, ViTs with the untargeted attack. For the ViTs, we choose ViT-B/16 and ViT-L/16 where B and L stand for "Base" and "Large", along with the patch size 16. The considered ViT models were pre-trained on ImageNet-21K and fine-tuned on ImageNet-1K. Similar to the ViT models, we also investigated Mixer-B/16 and Mixer-L/16, except that these models were directly trained on the ImageNet-1K without additional pre-training.

Untargeted Attack. The results in untargeted setting are shown in Table 2, from which several observations can be made. First, overall it can be observed that adversarially trained models transfer better than standard models. For I-FGSM, a standard ResNet18 and ResNet50 have a transfer rate of 76.4% and 79.2%, respectively, while their adversarially trained variants with ϵ_s set to 0.5 have a significantly higher transfer-rate of 98.8% and 99.3%, respectively. MI, DI, and TI can non-trivially improve the transferability; however, the transferability improvement through adversarial training is more effective. For example, when the ϵ_s is set to be as small as 0.05, the model accuracy is only slightly lower than its counterparts without adversarial training, i.e. 69.15% vs. 69.79% for ResNet18

Table 2: Attack success rate (%) with a single surrogate model on ImageNet for untargeted attack. The standard ($\epsilon = 0$) and various robust variants of two surrogate models (ResNet18, ResNet50) are evaluated. The presented ASRs are the average over four standard target models: (ResNet50 for ResNet18, ResNet18 for ResNet50), DenseNet121, VGG16 and MobileNetV2. All experiments are performed for 5 different FGSM-based attacks with the negative CE-loss as the objective function. Bold numbers indicate the best ASR for a specific attack. The best result for a surrogate model are indicated by an asterisk (*). More detailed results are shown in the appendix.

Surrogate	Attack	ϵ_s=0	ϵ_s=0.01	ϵ_s=0.03	ϵ_s=0.05	ϵ_s=0.1	ϵ_s=0.25	ϵ_s=0.5	ϵ_s=1	ϵ_s=3	ϵ_s=5
ResNet18	Accuracy	69.79	69.90	69.24	69.15	68.77	67.43	65.49	62.32	53.12	45.59
	I	76.4	89.1	94.5	96.7	97.9	98.7	**98.8**	98.2	94.3	88.8
	MI	84.3	93.0	96.6	97.3	98.5	**98.8**	98.4	97.5	92.1	84.7
	DI	95.5	98.6	98.9	99.2	99.3	99.4	**99.5***	98.7	94.0	87.7
	TI	80.3	88.9	93.8	95.5	97.1	**98.3**	98.2	97.0	91.3	83.5
	MI+DI+TI	98.0	98.9	98.8	**99.4**	99.2	99.3	98.6	97.2	89.0	78.5
ResNet50	Accuracy	75.80	75.68	75.76	75.59	74.78	74.14	73.16	70.43	62.83	56.13
	I	79.2	89.7	93.6	96.4	97.9	99.0	**99.3**	98.9	97.5	94.8
	MI	87.7	93.7	96.1	97.8	98.2	99.0	**99.1**	98.3	95.9	92.0
	DI	97.4	99.0	99.3	99.7	99.7	99.7	**99.8***	99.1	97.4	94.6
	TI	82.6	90.0	93.0	95.7	97.2	98.5	**98.7**	98.1	95.4	91.6
	MI+DI+TI	98.6	99.3	99.2	**99.5**	**99.5**	99.3	99.4	97.9	93.6	88.1

Table 3: Untargeted transfer-based black-box attack with a single surrogate model on ImageNet. Each cell presents the untargeted ASR. The standard ($\epsilon = 0$) and 9 robust variants of two surrogate models (ResNet18, ResNet50) are evaluated. All experiments are performed I-FGSM with the CE-loss. Bold numbers indicate the best ASR.

Surrogate	Attack	ϵ_s=0	ϵ_s=0.01	ϵ_s=0.03	ϵ_s=0.05	ϵ_s=0.1	ϵ_s=0.25	ϵ_s=0.5	ϵ_s=1	ϵ_s=3	ϵ_s=5
ResNet18	ViT-B/16	19.4	28.2	41.7	51.7	63.6	80.5	**87.4**	86.6	79.2	67.7
	ViT-L/16	14.9	23.3	34.4	41.3	55.2	73.5	81.9	**84.0**	74.9	64.6
	Mixer-B16	27.8	38.9	47.6	54.6	65.9	78.4	**83.3**	82.5	74.4	63.2
	Mixer-L/16	34.1	40.7	48.8	54.6	63.8	74.6	80.4	**82.4**	75.1	66.7
ResNet50	ViT-B/16	19.8	29.4	41.2	50.1	65.6	82.4	90.1	**91.8**	87.3	80.0
	ViT-L/16	16.2	23.0	32.1	42.2	59.5	75.3	84.7	**88.2**	85.8	75.8
	Mixer-B/16	30.3	40.9	49.3	58.5	69.4	82.1	87.3	**89.7**	84.8	77.8
	Mixer-L/16	36.1	42.1	50.4	58.1	67.9	78.5	85.8	**87.0**	83.3	78.3

and 75.59% vs. 75.80% for ResNet50, respectively, indicating only a small adversarial strength. While DI-FGSM can improve a standard ResNet18 to 95.5%, an adversarially trained model with $\epsilon_s = 0.05$ can achieve a higher transferability of 96.7% with only I-FGSM. Further, the effectiveness of MI, DI, and TI for improving the transferability are only significant on the standard model. When an appropriate ϵ_s is chosen, e.g. 0.25 or 0.3, they only marginally improve the transferability or even decrease the transferability. Overall, our results suggest that the vanilla I-FGSM is sufficient to achieve close to 100% attack success rate if the surrogate model is adversarially trained with an ϵ_s set to a range from 0.05 to 1. Such a wide range of ϵ_s, indicates the transferability performance is not *very* sensitive to the chosen ϵ_s. We further set the target models to beyond the CNN architectures, such as ViTs (Dosovitskiy et al., 2021) and Mixers (Tolstikhin et al., 2021) (Table 3). We find that when ϵ_s to 0.5 or 1, we observe a very high transfer rate, higher than 80% for all target models. Compared with the standard surrogate model ($\epsilon_s = 0$) the performance has been improved from a significant margin. For instance, from ResNet50 to ViT-B/16, the performance is increased from 19.8% to higher than 90%.

Targeted Attack. The results in the targeted setting are shown in Table 4. Similar to the trend in the untargeted setting, in all attack scenarios (I, MI, DI, TI, MI+DI+TI), adversarially trained surrogate models overall yield more transferable adversarial examples. For I-FGSM, the targeted attack success rate of the surrogate model ResNet18 and ResNet50 is as low as only 0.4% and 0.7%, respectively, while their adversarially trained counterparts with the ϵ_s set to 0.5, achieve a significantly higher transfer rate of 38.4% and 44.4% respectively. DI, among MI, DI, TI, is the most effective technique for boosting targeted transferability. With the assistance of DI, the targeted transfer rate on the adversarially trained model (with $\epsilon_s = 0.5$) is further boosted to 55.8% and 62.8%, for surrogate models ResNet18 and ResNet50, respectively. It is interesting to note that combing MI, DI, and TI achieves a non-trivial transferability boost for a standard surrogate model, however, such a surrogate model still under-performs an adversarially trained surrogate model with the vanilla I-FGSM attack, i.e. 12.4% vs. 38.4% for ResNet18 and 18.6% vs. 44.4% for ResNet50. Similar to the untargeted

setting, we find that MI and TI decrease the transferability performance for an adversarially trained model with ϵ_s set to an appropriate value, like 0.5.

Table 4: Targeted transfer-based black-box attack with a single surrogate model on ImageNet. Each cell presents the targeted ASR. The standard ($\epsilon = 0$) and 9 robust variants of three surrogate models (ResNet18, ResNet50, WRN50-2) are evaluated. The presented ASRs are the average over four standard target models. All experiments are performed for 5 different targeted FGSM variants with the CE-loss. Bold numbers indicate the best ASR for a specific attack. The best result for a surrogate model are indicated by an asterisk (*). Detailed results are available in the appendix.

Surrogate	Attack	ϵ_s=0	ϵ_s=0.01	ϵ_s=0.03	ϵ_s=0.05	ϵ_s=0.1	ϵ_s=0.25	ϵ_s=0.5	ϵ_s=1	ϵ_s=3	ϵ_s=5
ResNet18	I	0.4	1.5	3.6	6.3	13.1	27.0	**38.4**	34.8	16.1	7.2
	MI	1.1	3.1	4.9	8.5	13.3	25.3	**34.6**	32.6	16.5	7.8
	DI	7.9	19.7	31.1	38.9	51.0	**56.4***	55.8	42.6	16.8	5.9
	TI	0.5	1.6	3.4	5.7	11.2	23.5	**32.3**	28.2	10.6	4.2
	MI+DI+TI	12.4	20.2	26.3	30.2	36.3	**37.6**	32.2	21.3	5.2	1.4
ResNet50	I	0.7	2.5	4.5	7.9	14.5	29.4	44.4	**47.2**	31.4	17.9
	MI	2.4	6.1	8.9	13.4	18.7	30.7	**37.3**	33.7	16.0	7.4
	DI	13.3	23.5	30.9	42.1	55.1	61.2	**62.8**	56.5	30.5	16.4
	TI	0.9	2.5	3.9	7.2	14.1	26.6	38.2	**39.0**	22.4	1.5
	MI+DI+TI	18.6	24.9	26.4	33.6	40.0	**41.1**	38.9	30.1	11.4	4.4
WRN50-2	DI	68.1/14.1	76.0/21.7	83.4/34.1	86.3/36.8	90.9/46.1	**93.1/54.7**	92.2/50.8	90.6/46.5	83.0/28.3	76.5/17.2

Other Surrogate Models and Adversarial Training Techniques. We further provide evidence that the above transferability performance improvement is not limited to ResNet-architectures as surrogate models. We evaluate other model architectures for the I-FGSM attack in Table 5. The results further confirm that adversarial examples generated on robust models transfer better in terms of their non-targeted ASR and targeted ASR than those extracted from naturally trained models. Additionally, while the previous robust models were trained with ℓ_2 adversarial training, we test robust models that are trained with ℓ_∞ PGD adversarial training (Madry et al., 2018) as well as fast adversarial training (FAT) (Wong et al., 2020).

Table 5: Average ASR (%) with standard and robust surrogate models for I-FGSM on ImageNet. See appendix for detailed results.

Surrogate	ϵ_s=0	ϵ_s=3
DenseNet160	49.2/0.6	**65.9/5.4**
MNASNet	26.4/0.0	**47.7/0.5**
MobileNet	36.2/0.3	**62.2/4.3**
ResNeXt	41.9/0.3	**67.5/7.8**
ShuffleNet	23.7/0.0	**45.6/0.5**

The results are presented in Table 6. As in our previous observations, robust models for both adversarial training techniques result in significantly higher transferability compared to standard training. The adversarial examples generated from the robust model trained with PGD adversarial training transfer overall better than those for FAT for the same ϵ_s. The adversarial examples from robust models with ϵ_s=0.5 lead to the highest ASR; moreover, we can find that the DI attack method yields the largest ASR under the same epsilon value for both PGD and FAT adversarial training. Overall, the results suggest that the adversarial training type (PGD or FGSM) and perturbation norm type (ℓ_2 or ℓ_∞) are not essential. As long as they can facilitate the model to learn RFs, they all consistently yield more transferable surrogate models.

Transfer to very robust models. We further evaluate the transfer capability of standard and robust models to 4 very robust models, ResNet18/50, DenseNet161, VGG16), MobileNet-V2. Note that they are adversarially trained with PGD adversarial perturbation of ℓ_2-norm ϵ_t=3. The results are shown in Table 7. Here, we perform a targeted attack but report both non-targeted attack success rate and targeted attack success rate. Since the target models are very robust, when ϵ_s=0 or a very small value (lower than 0.1), the targeted attack success rate is always zero even under the strong transferable setup of MI+DI+TI. However, when the ϵ_s is set to 3 or 5, the vanilla I-FGSM already achieves a targeted attack success rate of higher than 10%. The untargeted attack success rate results further demonstrate that there is a clear trend of higher ϵ_s to induce higher transfer rate in this setup of adopting very robust models.

Adversarial Training with Early Stop. We refer the reader to the appendix (Fig. 4) for the transferability performance during adversarial training. Here, we do not intend to boost the performance with an early stop because choosing an appropriate ϵ_s might be more beneficial for the performance boost. Instead, we are mainly interested in whether the phenomenon that early stop improves transferability also applies to adversarial training. The results show that it does not apply to adversarial training. Admittedly, here, it depends on the choice of ϵ_s. Overall, for a sufficiently large ϵ_s, the technique

Table 6: Untargeted success rate (%) with a single surrogate model on ImageNet. We evaluate the surrogate model adversarially trained with PGD adversarial training and FAT with different ϵ_s. All Experiments are performed for I-FGSM, MI, DI, TI, MI-DI-TI untargeted attack with cross entropy loss. It should be noted that ASR values presented here are the average values of 4 target models: ResNet18, DenseNet121, VGG16, MobileNetV2. Detailed results are available in the appendix.

Surrogate	Attack	ϵ_s=0	ϵ_s=0.5 (PGD)	ϵ_s=1 (PGD)	ϵ_s=2 (PGD)	ϵ_s=4 (PGD)	ϵ_s=2 (FAT)	ϵ_s=4 (FAT)
ResNet50	I	79.2	**99.3**	98.7	97.1	91.8	**96.9**	90.3
	MI	87.7	**98.9**	98.1	95.5	89.5	**95.8**	87.8
	DI	97.4	**99.8**	99.1	97.4	92.4	**97.3**	90.1
	TI	82.6	**98.8**	97.7	95.1	87.4	**94.8**	85.0
	MI+DI+TI	98.6	**99.2**	97.7	94.8	87.3	**94.2**	83.9

Table 7: Transferability evaluation on the robust models including ResNet50 ($\epsilon_t = 3$), DenseNet161 ($\epsilon_t = 3$), VGG16 ($\epsilon_t = 3$), MobileNetV2 ($\epsilon_t = 3$). The surrogate model is adversarially trained ResNet18 or ResNet50 with ϵ_s ranging from 0 to 5 (l_2-norm). Each entry indicates non-targeted attack success / targeted attack success rate. It should be noted that ASR values presented here are the average values of 4 target models. More detailed results are shown in the appendix.

Surrogate	Attack	ϵ_s=0	ϵ_s=0.01	ϵ_s=0.03	ϵ_s=0.05	ϵ_s=0.1	ϵ_s=0.25	ϵ_s=0.5	ϵ_s=1	ϵ_s=3	ϵ_s=5
ResNet18	I	37.4/0.0	37.4/0.0	37.7/0.0	37.8/0.0	38.9/0.0	41.1/0.0	45.1/0.2	51.5/2.6	63.7/**12.0**	**65.3**/11.7
	MI	41.4/0.0	41.8/0.0	42.0/0.0	42.3/0.0	42.7/0.0	44.4/0.1	47.9/0.7	52.5/2.8	62.8/**8.4**	**63.6**/7.8
	DI	37.6/0.0	37.9/0.0	38.6/0.0	38.8/0.0	40.2/0.0	42.2/0.0	45.9/0.5	51.5/2.5	61.2/**8.5**	**62.5**/8.1
	TI	37.3/0.0	37.5/0.0	37.8/0.0	38.1/0.0	39.3/0.0	41.8/0.1	45.7/0.3	52.6/2.8	63.9/**12.5**	**65.3**/11.6
	MI+DI+TI	42.0/0.0	42.6/0.0	43.0/0.0	43.1/0.0	43.9/0.0	46.5/0.2	49.6/1.1	54.7/3.2	**63.0/7.2**	62.4/6.1
ResNet50	I	40.5/0.0	40.7/0.0	40.5/0.0	41.0/0.0	41.8/0.0	43.9/0.0	48.1/0.1	54.2/1.6	67.9/11.0	**72.2/13.9**
	MI	44.4/0.0	44.7/0.0	44.9/0.0	45.4/0.0	45.4/0.0	46.9/0.1	51.3/0.3	55.7/1.9	67.5/7.2	**70.7/8.9**
	DI	41.1/0.0	41.4/0.0	41.7/0.0	42.0/0.0	43.0/0.0	44.9/0.0	49.8/0.2	55.1/1.8	67.0/8.0	**69.1/9.1**
	TI	40.6/0.0	41.0/0.0	40.9/0.0	41.3/0.0	42.1/0.0	43.9/0.0	49.9/0.1	55.2/1.6	68.5/11.2	**72.5/13.9**
	MI+DI+TI	45.5/0.0	45.7/0.0	45.7/0.0	47.4/0.0	46.8/0.0	49.7/0.1	53.5/0.7	58.5/2.3	68.4/6.5	**69.6/6.9**

becomes ineffective, which is somewhat expected because early stop improves transferability in standard training because the model mainly learns NRFs/VNRFs in the later stage and adversarial training explicitly discourages learning NRFs/VNRFs.

5 Related works and Discussion

Techniques for Improving Transferability. Early investigations have evaluated the transferability of the white-box attack methods, such as I-FGSM (Kurakin et al., 2017) and PGD (Madry et al., 2018), but with limited success. Generating adversarial examples on an ensemble of models is found to improve transferability (Liu et al., 2017; Tramèr et al., 2018). One line of works extends the I-FGSM with momentum (MI-FGSM) (Dong et al., 2018), input diversity (DI^2-FGSM) (Xie et al., 2019), and translation-invariant property(TI-FGSM) (Dong et al., 2019), to improve adversarial transferability. Fine-tuning adversarial examples with intermediate-level attacks (Huang et al., 2019; Li et al., 2020b) also boosts transferability. Towards transferable targeted attack, (Li et al., 2020a) has proposed a new Po-Trip loss to replace the cross-entropy loss. Another line of works (Inkawhich et al., 2019; 2020a;b) has attempted to perform optimization in feature space for a more transferable targeted attack. These approaches (Inkawhich et al., 2020a;b) require training class-wise and layer-wise auxiliary classifiers. Recently, backpropagating linearly, *e.g*, ignoring some ReLU activations (Guo et al., 2020) or decreasing the weight on residual path (Wu et al., 2020), is also found to be beneficial for transferability enhancement. Most of them mainly or exclusively focus on the attack strategies on the second stage of TBA, while our work investigates simple techniques for achieving a better surrogate model in the first stage.

Transfer learning. Several studies (Liang et al., 2020; Terzi et al., 2020; Utrera et al., 2021; Salman et al., 2020) have found that adversarial training improves model generalization for downstream tasks in transfer learning. Their finding refutes a widely held belief in transfer learning that models with higher accuracy tend to transfer better. In essence, their works show adversarial training improves transferability to a new *dataset*, while our work shows it improves the transferability to a new *model*. Our finding in the TBA task might also provide new insight into the phenomenon observed in transfer learning. For example, VNRFs might also harm the transferability in transfer learning, and we leave such investigation to future work.

REFERENCES

Ronen Basri, David Jacobs, Yoni Kasten, and Shira Kritchman. The convergence rate of neural networks for learned functions of different frequencies. *NeurIPS*, 2019.

Philipp Benz, Chaoning Zhang, and In So Kweon. Batch normalization increases adversarial vulnerability: Disentangling usefulness and robustness of model features. *arXiv preprint arXiv:2010.03316*, 2020.

Tejas Borkar, Felix Heide, and Lina Karam. Defending against universal attacks through selective feature regeneration. In *CVPR*, 2020.

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

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

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Yiwen Guo, Qizhang Li, and Hao Chen. Backpropagating linearly improves transferability of adversarial examples. *arXiv preprint arXiv:2012.03528*, 2020.

Qian Huang, Isay Katsman, Horace He, Zeqi Gu, Serge Belongie, and Ser-Nam Lim. Enhancing adversarial example transferability with an intermediate level attack. In *ICCV*, 2019.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *NeurIPS*, 2019.

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

Nathan Inkawhich, Kevin J Liang, Lawrence Carin, and Yiran Chen. Transferable perturbations of deep feature distributions. *ICLR*, 2020a.

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. *NeurIPS*, 2020b.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). *URL http://www. cs. toronto. edu/kriz/cifar. html*, 2010.

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

Maosen Li, Cheng Deng, Tengjiao Li, Junchi Yan, Xinbo Gao, and Heng Huang. Towards transferable targeted attack. In *CVPR*, 2020a.

Qizhang Li, Yiwen Guo, and Hao Chen. Yet another intermediate-level attack. In *ECCV*, 2020b.

Yuanzhi Li, Colin Wei, and Tengyu Ma. Towards explaining the regularization effect of initial large learning rate in training neural networks. *arXiv preprint arXiv:1907.04595*, 2019.

Kaizhao Liang, Jacky Y Zhang, Oluwasanmi Koyejo, and Bo Li. Does adversarial transferability indicate knowledge transferability? *arXiv preprint arXiv:2006.14512*, 2020.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *ICLR*, 2017.

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

Vikram Nitin. Sgd on neural networks learns robust features before non-robust. *ICLR Review*, 2021.

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. doi: 10.1007/s11263-015-0816-y.

Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? In *NeurIPS*, 2020.

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.

Matteo Terzi, Alessandro Achille, Marco Maggipinto, and Gian Antonio Susto. Adversarial training reduces information and improves transferability. *arXiv preprint arXiv:2007.11259*, 2020.

Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *ICLR*, 2018.

Francisco Utrera, Evan Kravitz, N Benjamin Erichson, Rajiv Khanna, and Michael W Mahoney. Adversarially-trained deep nets transfer better. *ICLR*, 2021.

Eric Wong, Leslie Rice, and J Zico Kolter. Fast is better than free: Revisiting adversarial training. *ICLR*, 2020.

Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. *arXiv preprint arXiv:2002.05990*, 2020.

Lei Wu, Zhanxing Zhu, Cheng Tai, et al. Understanding and enhancing the transferability of adversarial examples. *arXiv preprint arXiv:1802.09707*, 2018.

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

Zhi-Qin John Xu, Yaoyu Zhang, and Yanyang Xiao. Training behavior of deep neural network in frequency domain. In *ICONIP*, 2019.

A Appendix

A.1 Transferability of Early-stopped robust models

We further evaluate the early stopped model's transferability on the adversarially trained models. As mentioned in the main manuscript, we do not intend to boost the performance with early stopping because choosing an appropriate ϵ_s might be more beneficial for the performance boost. Instead, we are mainly interested in whether the phenomenon that early stop improves transferability also applies to adversarial training. The results in Figure **??** show that the trend is different from that in Figure 1 in the main manuscript even though we also observe a decrease in transferability in the final stage of training of Figure **??**. For the normally trained model on CIFAR10, the increase in the transferability performance through early stopping is more obvious, while that for the adversarially trained model (Figure **??** (left)) fluctuates more with a less obvious transferability in the final stage. This is somewhat expected because adversarial training already encourages the model to learn robust features and early stop is less significant in adversarial training for improving the transferability. The transferability capabilities of adversarial examples extracted from an adversarially trained ImageNet model are presented in Figure **??** (right), where nearly no positive effect on the transferability can be observed for the early stop.

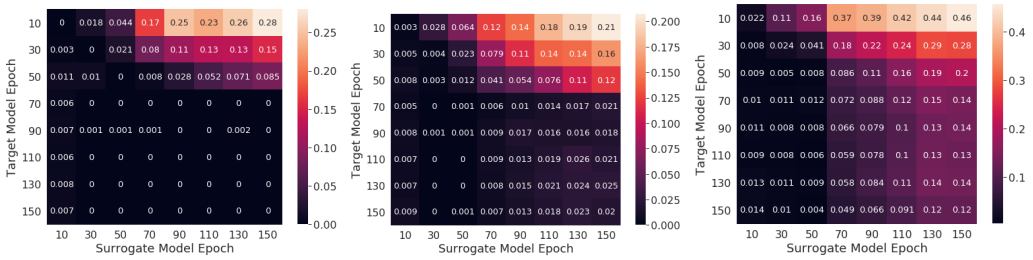

Figure 3: Transferability evaluation for adversarial examples generated on early-stopped surrogate models S_n and evaluated on early-stopped target models T_n. Left: The same ResNet18 with the same model parameters as surrogate and target model. Center: Two separately trained ResNet18 architectures(ResNet18-A, ResNet18-B) for surrogate and target model. Right: ResNet18 as the surrogate model and VGG16 as the target model. Each entry represents the target model accuracy evaluated on the obtained adversarial examples from S_n. The experiments were performed for CIFAR10 and the I-FGSM attack.

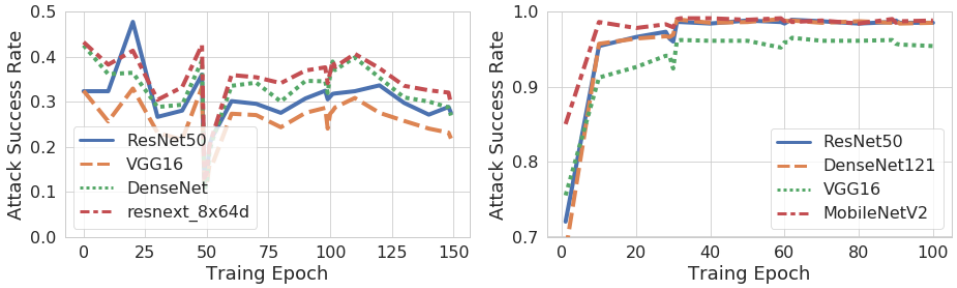

Figure 4: Transferability of adversarial examples generated for a robust ResNet18 as the surrogate model. Left: Transferability for an adversarially trained model with l_∞-PGD with ϵ=8/255 for CIFAR10. Right: Transferability for an adversarially trained model with l_∞-FGSM with ϵ=1/255 for ImageNet. The transferability is evaluated for the adversarial examples obtained from the models during different adversarial training stages of the surrogate model. The adversarial examples were generated with an untargeted I-FGSM. The results are reported in the untargeted ASR.

Table 8: Full results of Table 2 of the main manuscript showing the Attack success rate (%) with a single surrogate model on ImageNet. The standard ($\epsilon = 0$) and various robust variants of two surrogate models (ResNet18, ResNet50) are evaluated. The detailed ASR, as well as the average, are reported. All experiments are performed for 5 different FGSM-based attacks with the negative CE-loss as the objective function. Bold numbers indicate the best ASR for a specific attack.

Source	Attack	Target	ϵ_s=0	ϵ_s=0.01	ϵ_s=0.03	ϵ_s=0.05	ϵ_s=0.1	ϵ_s=0.25	ϵ_s=0.5	ϵ_s=1	ϵ_s=3	ϵ_s=5
ResNet18	I	ResNet50	79.2	91.3	96.4	98.4	98.6	99.4	**99.5**	99.2	95.5	89.4
		DenseNet121	71.9	88.2	94.5	97.1	98.6	**99.3**	99.2	98.9	94.9	88.0
		VGG16	75.9	86.1	92.2	93.9	96.5	96.7	**97.2**	95.9	89.7	82.7
		MobileNetV2	78.6	90.8	95.0	97.5	97.8	**99.4**	99.3	98.9	97.2	95.0
		Average	76.4	89.1	94.5	96.7	97.9	98.7	**98.8**	98.2	94.3	88.8
	MI	ResNet50	86.1	94.7	97.9	98.3	98.7	**99.6**	99.2	98.7	92.8	84.6
		DenseNet121	82.5	93.3	97.1	97.8	98.9	**99.0**	98.8	98.5	92.5	83.7
		VGG16	83.0	90.7	94.6	95.6	**97.4**	97.1	96.6	94.4	87.1	78.1
		MobileNetV2	85.6	93.5	96.7	97.7	98.8	**99.3**	98.8	98.4	96.1	92.2
		Average	84.3	93.0	96.6	97.3	98.5	**98.8**	98.4	97.5	92.1	84.7
	DI	ResNet50	96.4	99.0	98.9	99.5	99.5	99.7	99.7	99.4	94.9	87.4
		DenseNet121	94.5	98.7	99.2	99.6	99.5	99.6	**99.7**	99.2	94.9	87.4
		VGG16	95.8	98.4	98.3	98.8	98.9	98.7	**99.2**	97.0	89.3	83.0
		MobileNetV2	95.2	98.4	99.1	99.1	99.4	**99.7**	99.5	99.0	97.1	93.0
		Average	95.5	98.6	98.9	99.2	99.3	99.4	**99.5**	98.7	94.0	87.7
	TI	ResNet50	83.7	91.3	94.9	97.2	98.2	**99.4**	98.7	98.0	92.0	82.8
		DenseNet121	77.0	88.3	94.6	96.1	97.6	**99.4**	98.9	97.9	92.4	83.3
		VGG16	78.4	86.8	90.8	92.5	95.2	95.1	**96.1**	93.4	85.4	77.0
		MobileNetV2	82.3	89.3	94.7	96.3	97.3	**99.2**	98.9	98.7	95.5	90.8
		Average	80.3	88.9	93.8	95.5	97.1	**98.3**	98.2	97.0	91.3	83.5
	MI+DI+TI	ResNet50	98.1	99.3	99.2	**99.9**	99.6	99.6	99.3	97.9	89.2	77.2
		DenseNet121	98.1	99.1	98.9	99.5	**99.6**	99.4	99.0	98.2	89.6	76.6
		VGG16	98.1	**98.9**	98.4	98.8	98.5	98.6	97.0	93.9	82.4	72.2
		MobileNetV2	97.8	98.4	98.7	99.3	99.1	**99.5**	98.9	98.6	94.7	88.0
		Average	98.0	98.9	98.8	**99.4**	99.2	99.3	98.6	97.2	89.0	78.5
ResNet50	I	ResNet18	85.4	94.0	96.2	98.0	99.4	99.7	99.5	99.7	99.1	98.5
		DenseNet121	80.1	90.7	94.4	98.0	98.7	99.6	**99.7**	99.4	98.1	95.4
		VGG16	75.1	85.8	91.0	93.4	95.4	97.5	**98.4**	97.5	94.6	88.7
		MobileNetV2	76.0	88.2	92.9	96.3	98.0	99.3	**99.5**	99.0	98.2	96.8
		Average	79.2	89.7	93.6	96.4	97.9	99.0	99.3	98.9	97.5	94.8
	MI	ResNet18	92.4	96.6	97.9	99.1	99.4	99.5	99.7	99.3	98.2	97.2
		DenseNet121	89.1	94.8	97.0	98.7	99.1	**99.8**	99.7	98.7	96.5	91.7
		VGG16	83.2	90.4	93.6	96.1	96.3	**98.0**	97.9	96.7	91.3	84.5
		MobileNetV2	86.2	93.1	95.8	97.4	98.0	98.7	**99.2**	98.7	97.6	94.7
		Average	87.7	93.7	96.1	97.8	98.2	99.0	99.1	98.3	95.9	92.0
	DI	ResNet18	98.5	99.5	99.6	99.8	**99.9**	**99.9**	99.8	99.7	98.9	98.2
		DenseNet121	98.1	99.5	99.6	**99.9**	99.8	**99.9**	**99.9**	99.3	98.1	94.5
		VGG16	96.9	98.5	99.2	99.4	99.3	99.3	**99.5**	98.0	94.3	89.0
		MobileNetV2	96.1	98.4	99.0	99.5	99.7	99.7	**99.8**	99.5	98.2	96.9
		Average	97.4	99.0	99.3	99.7	99.7	99.7	**99.8**	99.1	97.4	94.6
	TI	ResNet18	86.9	93.7	95.3	97.3	99.1	99.4	**99.7**	99.2	98.2	97.1
		DenseNet121	78.2	86.5	89.5	92.4	93.7	**96.4**	96.3	95.5	89.8	83.0
		VGG16	78.2	86.5	89.5	92.4	93.7	**96.4**	96.3	95.5	89.8	83.0
		MobileNetV2	80.5	88.6	92.9	95.8	97.1	98.6	**99.3**	98.6	97.3	94.5
		Average	82.6	90.0	93.0	95.7	97.2	98.5	**98.7**	98.1	95.4	91.6
	MI+DI+TI	ResNet18	99.3	99.8	99.8	**99.9**	**99.9**	99.8	99.8	99.2	97.4	94.7
		DenseNet121	99.1	99.6	99.5	99.8	**99.9**	99.5	99.8	98.5	94.8	87.1
		VGG16	98.3	98.8	98.3	**99.1**	99.0	98.9	98.3	95.6	87.1	78.8
		MobileNetV2	97.9	98.9	99.1	99.2	99.3	99.1	**99.6**	98.4	95.0	91.8
		Average	98.6	99.3	99.2	**99.5**	**99.5**	99.3	99.4	97.9	93.6	88.1

A.2 FULL EXPERIMENT RESULTS

The majority of the results presented in the main manuscript report the average ASR over different models. Here, we present the detailed results for each target model. Specifically, the full results corresponding to Table 2, Table 4, Table 6, Table 7, Table 5 in the main manuscript are shown in Table 8, Table 9, Table 10, Table 11, Table 12, respectively.

Table 9: Full results of Table 4 of the main manuscript showing the targeted transfer-based black-box attack with a single surrogate model on ImageNet. In each cell, the untargeted and targeted ASR are presented. The standard ($\epsilon = 0$) and 9 robust variants of three surrogate models (ResNet18, ResNet50, WRN50-2) are evaluated. All experiments are performed for 5 different targeted FGSM variants with the CE-loss. Bold numbers indicate the best ASR for a specific attack. Each entry shows non-targeted ASR / targeted ASR. Note that we can also report non-targeted ASR in the targeted setting.

Source	Attack	Target	ϵ_s=0	ϵ_s=0.01	ϵ_s=0.03	ϵ_s=0.05	ϵ_s=0.1	ϵ_s=0.25	ϵ_s=0.5	ϵ_s=1	ϵ_s=3	ϵ_s=5
ResNet18	I	ResNet50	37.1/0.7	44.9/2.9	55.0/5.3	59.8/9.6	69.2/17.6	83.9/34.1	**87.2/46.9**	85.0/40.6	70.8/16.2	60.7/8.9
		DenseNet121	31.5/0.2	41.0/1.1	51.0/4.1	56.9/6.6	69.1/14.7	79.8/30.8	**85.7/44.0**	83.3/39.8	70.2/19.0	59.9/7.4
		VGG16	41.0/0.5	49.4/0.9	55.8/1.5	58.8/3.1	67.7/7.6	77.4/13.4	**78.6/18.6**	78.1/16.0	66.6/6.4	57.4/2.4
		MobileNetV2	43.2/0.1	52.9/1.2	61.0/3.5	67.2/5.9	75.3/12.7	85.9/29.8	**90.6/44.1**	88.9/42.6	81.4/22.7	74.7/10.3
		Average	38.2/0.4	47.1/1.5	55.7/3.6	60.7/6.3	70.3/13.1	81.8/27.0	**85.5/38.4**	83.8/34.8	72.2/16.1	63.2/7.2
	MI	ResNet50	50.5/0.4	63.5/3.3	70.6/6.1	74.5/9.8	79.9/16.4	**84.6**/32.2	83.9/**35.8**	79.8/27.2	63.1/9.4	52.1/3.0
		DenseNet121	43.4/0.4	59.2/1.6	67.3/5.4	72.7/9.0	78.7/16.5	**83.2**/29.7	81.6/**34.8**	77.8/27.6	63.3/9.2	52.2/2.7
		VGG16	53.3/0.2	64.1/1.0	68.1/2.0	72.6/2.8	75.5/4.2	**78.5**/10.0	77.3/**11.7**	72.8/9.2	60.4/2.5	53.0/0.7
		MobileNetV2	61.8/0.3	70.7/1.8	75.5/3.8	79.4/5.0	83.6/9.5	86.5/25.0	**87.7/32.5**	84.6/29.1	76.0/11.1	65.9/3.9
		Average	52.2/0.3	64.4/1.9	70.4/4.3	74.8/6.7	79.4/11.7	**83.2**/24.2	82.6/**28.7**	78.8/23.3	65.7/8.1	55.8/2.6
	DI	ResNet50	58.7/9.0	73.0/24.9	81.2/36.5	86.6/45.8	90.2/58.0	**92.0/62.8**	90.6/61.4	85.9/46.8	71.8/17.5	57.9/6.4
		DenseNet121	52.4/8.0	68.0/21.7	79.1/32.2	85.4/41.5	90.2/56.2	**91.9/62.1**	91.6/60.9	86.1/46.1	71.6/18.4	56.8/5.4
		VGG16	64.7/9.1	73.4/18.4	79.9/27.5	82.7/32.2	**87.7**/40.7	87.5/**41.6**	85.4/37.6	81.3/25.2	66.2/7.4	55.6/2.1
		MobileNetV2	63.3/5.3	75.3/13.8	82.5/28.1	85.8/36.2	90.5/49.3	92.9/59.0	**93.5/63.2**	91.1/52.1	82.3/23.9	71.6/9.8
		Average	59.8/7.9	72.4/19.7	80.7/31.1	85.1/38.9	89.7/51.0	**91.1/56.4**	90.3/55.8	86.1/42.6	73.0/16.8	60.5/5.9
	TI	ResNet50	38.6/0.9	44.0/2.5	50.0/4.5	54.7/7.9	63.5/14.5	76.0/29.6	**80.0/38.7**	77.8/30.7	62.4/11.4	51.6/4.0
		DenseNet121	34.4/0.2	41.6/1.8	49.5/3.8	53.0/6.1	65.9/13.4	75.4/26.7	**80.4/38.4**	78.4/33.4	63.4/12.3	52.1/4.8
		VGG16	44.5/0.7	48.5/0.9	54.2/2.1	56.3/3.4	60.7/5.9	70.3/11.8	**72.6/14.0**	71.1/12.0	59.0/3.5	51.0/1.3
		MobileNetV2	45.6/0.3	51.3/1.2	58.6/3.0	60.6/5.5	70.5/11.1	81.1/26.1	**86.8/38.2**	84.6/36.9	73.8/15.2	66.1/6.5
		Average	40.8/0.5	46.4/1.6	53.1/3.4	56.1/5.7	65.2/11.2	75.7/23.5	**80.0/32.3**	78.0/28.2	64.7/10.6	55.2/4.2
	MI+DI+TI	ResNet50	75.5/14.8	80.8/25.1	84.3/31.2	85.6/37.9	**88.0**/43.6	86.9/**45.1**	82.5/37.4	76.6/24.2	58.8/5.7	45.2/1.2
		DenseNet121	71.8/13.0	79.5/24.0	84.3/31.8	86.8/37.5	86.7/46.0	**87.4/46.6**	84.3/40.1	77.7/24.9	57.7/5.9	44.9/1.2
		VGG16	77.8/13.2	81.8/16.6	**83.2**/19.7	83.1/20.0	81.8/**21.8**	80.2/18.4	76.0/14.8	70.8/9.1	54.7/1.4	48.7/0.5
		MobileNetV2	81.2/8.7	82.9/15.3	86.0/22.6	87.5/25.3	87.6/33.8	**88.6/40.3**	88.1/36.6	82.8/26.9	70.7/7.7	59.4/2.7
		Average	76.6/12.4	81.2/20.2	84.5/26.3	85.8/30.2	**86.0**/36.3	85.8/**37.6**	82.7/32.2	77.0/21.3	60.5/5.2	49.5/1.4
ResNet50	I	ResNet18	53.2/1.3	61.9/3.7	62.7/5.8	70.4/10.2	81.7/19.6	88.3/36.9	93.2/54.7	**93.8/59.3**	89.9/41.5	83.4/25.0
		DenseNet121	35.6/0.6	45.1/3.0	51.0/5.7	61.7/12.1	71.3/19.9	84.8/39.2	**90.8/56.4**	90.7/55.3	83.8/38.3	72.7/20.1
		VGG16	42.3/0.2	49.5/1.3	55.5/2.8	62.9/4.5	66.8/6.7	79.3/15.1	**83.9**/23.8	81.9/**26.1**	74.9/12.4	65.9/6.2
		MobileNetV2	44.5/0.6	54.8/2.2	57.8/3.5	67.3/5.0	76.4/11.8	85.7/26.6	90.7/42.9	**91.0/48.0**	87.3/33.4	83.4/20.2
		Average	43.9/0.7	52.8/2.5	56.8/4.5	65.6/7.9	74.1/14.5	84.5/29.4	**89.6**/44.4	89.3/**47.2**	84.0/31.4	76.3/17.9
	MI	ResNet18	71.6/1.3	79.2/4.5	79.6/5.9	83.5/10.4	87.9/17.9	89.8/31.2	**92.4/43.6**	91.3/41.4	85.6/23.1	78.1/11.3
		DenseNet121	50.6/2.0	60.0/4.9	64.2/8.1	75.7/14.2	80.0/21.8	87.8/38.8	**90.1/47.5**	87.8/41.5	74.5/20.8	64.7/9.3
		VGG16	57.2/0.7	64.7/1.2	65.3/1.9	73.8/3.8	75.6/5.7	81.5/11.2	**83.4/16.1**	77.3/13.4	68.1/4.5	59.3/2.0
		MobileNetV2	63.1/0.4	72.1/1.8	74.6/3.6	80.6/5.4	82.7/7.8	87.0/19.9	**88.8**/31.2	88.3/**34.2**	82.1/17.5	77.1/8.8
		Average	60.6/1.1	69.0/3.1	70.9/4.9	78.4/8.5	81.5/13.3	86.5/25.3	**88.7/34.6**	86.2/32.6	77.6/16.5	69.8/7.8
	DI	ResNet18	73.4/12.3	82.5/25.2	86.0/30.5	91.7/43.5	95.7/60.2	96.3/66.1	**97.2/69.7**	95.6/64.0	89.1/38.6	81.6/23.2
		DenseNet121	65.8/20.7	77.3/34.2	84.5/42.5	90.2/55.7	94.4/66.6	95.9/**72.8**	**96.2**/72.7	93.8/63.6	82.5/34.6	72.4/17.6
		VGG16	71.7/14.5	77.5/21.0	83.4/28.2	88.0/36.6	90.6/46.1	91.6/**48.8**	**92.1**/46.4	88.6/39.5	76.9/14.1	64.8/6.2
		MobileNetV2	67.2/5.8	77.1/13.7	82.5/22.3	88.5/32.7	92.7/47.6	94.0/57.3	**94.4/62.5**	93.1/58.9	86.9/34.6	81.9/18.7
		Average	69.5/13.3	78.6/23.5	84.1/30.9	89.6/42.1	93.3/55.1	94.5/61.2	**95.0/62.8**	92.8/56.5	83.8/30.5	75.2/16.4
	TI	ResNet18	54.1/1.6	60.4/3.5	63.1/5.4	69.5/9.7	77.8/18.6	84.5/34.4	**90.3**/49.0	89.6/**51.4**	85.0/32.5	75.7/17.3
		DenseNet121	39.5/0.7	44.5/2.7	51.5/5.6	59.7/9.8	68.4/18.5	80.9/37.8	**88.2/51.2**	87.5/48.6	75.4/24.6	64.9/13.1
		VGG16	47.0/0.8	50.1/1.5	53.7/2.0	58.6/3.7	63.5/7.1	72.4/11.8	**77.0**/16.5	75.0/**18.1**	67.5/7.7	58.5/2.6
		MobileNetV2	49.4/0.6	54.1/2.4	58.6/2.6	64.0/5.4	71.1/12.1	81.6/22.4	87.5/36.2	**87.5/38.1**	81.1/24.7	73.4/12.9
		Average	47.5/0.9	52.3/2.5	56.7/3.9	63.0/7.2	70.2/14.1	79.8/26.6	**85.8**/38.2	84.9/**39.0**	77.2/22.4	68.1/11.5
	MI+DI+TI	ResNet18	6.2/17.7	89.1/27.4	89.5/29.7	92.2/36.0	93.9/48.2	92.8/47.9	92.4/46.8	89.7/38.7	79.1/17.3	70.1/7.4
		DenseNet121	79.3/31.0	83.7/37.2	86.3/41.3	91.7/52.2	**92.0**/54.6	90.9/**56.1**	91.6/52.5	84.4/39.9	69.0/14.1	56.9/4.4
		VGG16	80.7/16.2	82.6/19.3	84.0/18.2	**86.4**/23.2	86.0/**23.7**	84.6/**23.7**	82.3/19.4	76.2/11.6	62.2/2.5	54.1/1.2
		MobileNetV2	81.3/9.5	86.1/15.7	85.9/16.4	90.0/23.0	**90.2**/33.3	87.8/**36.8**	89.1/36.7	86.3/30.3	75.1/11.7	67.2/4.6
		Average	81.9/18.6	85.4/24.9	86.4/26.4	90.1/33.6	90.5/40.0	89.0/41.1	88.8/38.9	84.2/30.1	71.3/11.4	62.1/4.4
WRN50-2	DI	ResNet50	75.1/27.1	81.1/38.8	90.0/59.6	92.7/60.6	93.9/67.9	**95.9/74.0**	94.9/64.5	92.8/56.0	83.5/33.6	76.4/20.6
		DenseNet121	64.6/16.5	75.1/25.1	83.8/38.5	85.2/42.2	92.2/53.3	**94.8/64.9**	93.6/60.0	91.1/55.4	83.9/32.0	76.6/21.2
		VGG16	69.0/9.6	77.4/14.8	80.4/21.8	84.5/25.0	88.3/32.3	**90.3/37.7**	87.9/32.4	86.5/29.5	77.0/15.7	70.3/7.9
		MobileNetV2	63.6/3.0	70.6/8.0	79.3/16.6	82.8/19.4	89.0/30.7	91.4/42.2	**92.6/46.1**	92.0/45.1	87.6/31.9	82.7/19.3
		Average	68.1/14.1	76.0/21.7	83.4/34.1	86.3/36.8	90.9/46.1	**93.1/54.7**	92.2/50.8	90.6/46.5	83.0/28.3	76.5/17.2

Table 10: Full results of Table 6 in the main manuscript showing the transferability evaluation on the robust models including ResNet50 ($\epsilon_t = 3$), DenseNet161($\epsilon_t = 3$), VGG16($\epsilon_t = 3$), MobileNetV2($\epsilon_t = 3$). The surrogate model is adversarially trained ResNet18 or ResNet50 with ϵ_s ranging from 0 to 5 (l_2-norm).

Source	Attack	Target	ϵ_s=0 (Normal)	ϵ_s=0.5 (PGD)	ϵ_s=1 (PGD)	ϵ_s=2 (PGD)	ϵ_s=4 (PGD)	ϵ_s=2 (FAT)	ϵ_s=4 (FAT)
ResNet50	I	ResNet18	85.4	**99.8**	99.6	98.9	95.3	**99.1**	96.7
		DenseNet121	80.1	**99.7**	99.1	97.7	92.0	**97.5**	89.4
		VGG16	75.1	**98.4**	97.5	94.1	86.2	**92.5**	81.7
		MobileNetV2	76.0	**99.3**	98.6	97.6	93.8	**98.3**	93.6
		Average	79.2	**99.3**	98.7	97.1	91.8	**96.9**	90.3
	MI	ResNet18	92.4	**99.6**	99.2	97.9	94.2	**98.7**	95.1
		DenseNet121	89.1	**99.5**	98.3	95.8	88.6	**96.0**	85.4
		VGG16	83.2	**97.7**	96.5	92.0	83.5	**91.1**	79.3
		MobileNetV2	86.2	**98.8**	98.4	96.5	91.7	**97.6**	91.4
		Average	87.7	**98.9**	98.1	95.5	89.5	**95.8**	87.8
	DI	ResNet18	98.5	**99.9**	99.6	98.8	96.2	**99.3**	96.2
		DenseNet121	98.1	**99.9**	99.4	97.7	92.3	**97.2**	88.7
		VGG16	96.9	**99.7**	98.1	95.1	87.1	**94.6**	81.7
		MobileNetV2	96.1	**99.5**	99.3	97.9	94.0	**98.0**	93.7
		Average	97.4	**99.8**	99.1	97.4	92.4	**97.3**	90.1
	TI	ResNet18	86.9	**99.6**	99.2	98.0	92.9	**98.7**	93.8
		DenseNet121	78.2	**99.7**	98.4	95.7	87.6	**95.4**	82.8
		VGG16	78.2	**97.1**	94.8	90.0	78.6	**88.2**	73.8
		MobileNetV2	80.5	**99.0**	98.3	96.6	90.3	**96.8**	89.6
		Average	82.6	**98.8**	97.7	95.1	87.4	**94.8**	85.0
	MI+DI+TI	ResNet18	99.3	**99.8**	98.9	97.3	93.0	**98.1**	92.7
		DenseNet121	99.1	**99.8**	98.4	95.8	86.9	**94.8**	81.3
		VGG16	98.3	**98.2**	95.6	90.6	79.4	**88.0**	73.2
		MobileNetV2	97.9	**98.9**	97.9	95.4	90.0	**96.1**	88.3
		Average	98.6	**99.2**	97.7	94.8	87.3	**94.2**	83.9

Table 11: Full results of Table 7 showing the transferability evaluation on robust models. The surrogate model is an adversarially trained ResNet18 or ResNet50 with ϵ_s ranging from 0 to 5 (l_2-norm). Each entry indicates non-targeted attack success / targeted attack success rate.

Source	Attack	Target	ϵ_s=0	ϵ_s=0.01	ϵ_s=0.03	ϵ_s=0.05	ϵ_s=0.1	ϵ_s=0.25	ϵ_s=0.5	ϵ_s=1	ϵ_s=3	ϵ_s=5
ResNet18	I	ResNet50 (ϵ_t=3)	23.6/0.0	23.6/0.0	24.2/0.0	24.4/0.0	25.5/0.0	28.7/0.0	35.7/0.1	43.0/3.9	57.5/15.0	58.8/14.9
		DenseNet161 (ϵ_t=3)	25.9/0.0	26.1/0.0	26.5/0.0	26.4/0.0	26.7/0.0	28.9/0.0	32.4/0.1	37.8/1.9	50.1/9.6	52.3/8.9
		VGG16 (ϵ_t =3)	45.2/0.0	44.9/0.0	45.1/0.0	45.3/0.0	46.3/0.0	48.6/0.0	50.8/0.3	58.2/2.3	70.1/11.1	71.3/10.4
		MobileNetV2 (ϵ_t=3)	54.7/0.0	54.9/0.0	55.1/0.0	55.2/0.0	57.1/0.0	58.4/0.0	61.4/0.5	67.0/2.3	77.1/12.2	78.7/12.7
		Average (ϵ_t =3)	37.4/0.0	37.4/0.0	37.7/0.0	37.8/0.0	38.9/0.0	41.1/0.0	45.1/0.2	51.5/2.6	63.7/12.0	65.3/11.7
	MI	ResNet50 (ϵ_t=3)	36.6/0.0	37.5/0.0	37.7/0.0	36.7/0.0	36.3/0.0	36.5/0.2	39.7/1.0	45.3/3.8	55.4/9.5	55.7/9.9
		DenseNet161 (ϵ_t=3)	26.8/0.0	27.2/0.0	27.2/0.0	28.4/0.0	27.6/0.0	30.3/0.0	34.7/0.4	38.9/2.2	49.5/6.3	50.9/5.2
		VGG16 (ϵ_t=3)	46.1/0.0	46.2/0.0	46.5/0.0	47.2/0.0	48.0/0.0	49.4/0.0	53.6/0.7	58.1/2.9	69.1/8.5	69.4/7.2
		MobileNetV2 (ϵ_t=3)	56.1/0.0	56.2/0.0	56.8/0.0	56.8/0.0	58.8/0.0	61.2/0.2	63.7/0.7	67.5/2.1	77.2/9.5	78.5/8.7
		Average (ϵ_t=3)	41.4/0.0	41.8/0.0	42.0/0.0	42.3/0.0	42.7/0.0	44.4/0.1	47.9/0.7	52.5/2.8	62.8/8.4	63.6/7.8
	DI	ResNet50 (ϵ_t=3)	24.1/0.0	24.4/0.0	24.9/0.0	25.8/0.0	27.7/0.0	30.3/0.0	37.2/0.6	45.0/4.8	59.4/14.9	59.8/13.5
		DenseNet161 (ϵ_t=3)	26.4/0.0	26.9/0.0	27.8/0.0	26.8/0.0	27.7/0.0	29.7/0.0	32.2/0.1	37.5/1.2	44.8/5.2	47.3/5.0
		VGG16 (ϵ_t=3)	45.4/0.0	44.8/0.0	45.7/0.0	45.8/0.0	47.7/0.0	48.1/0.0	51.8/0.7	56.5/2.1	66.1/6.8	67.4/6.4
		MobileNetV2 (ϵ_t=3)	54.7/0.0	55.5/0.0	56.2/0.0	56.6/0.0	57.8/0.0	60.7/0.1	62.2/0.7	67.2/2.1	74.7/7.1	75.3/7.4
		Average (ϵ_t=3)	37.6/0.0	37.9/0.0	38.6/0.0	38.8/0.0	40.2/0.0	42.2/0.0	45.9/0.5	51.5/2.5	61.2/8.5	62.5/8.1
	TI	ResNet50 (ϵ_t=3)	23.3/0.0	23.1/0.0	23.6/0.0	24.2/0.0	25.7/0.0	29.1/0.0	35.3/0.2	44.1/3.8	57.3/15.1	58.7/14.8
		DenseNet161 (ϵ_t=3)	26.1/0.0	26.6/0.0	27.1/0.0	26.3/0.0	26.9/0.0	29.5/0.0	33.0/0.2	39.5/2.0	51.1/10.2	52.6/8.6
		VGG16 (ϵ_t=3)	44.9/0.0	45.4/0.0	45.5/0.0	46.1/0.0	46.7/0.0	48.4/0.0	51.8/0.5	58.7/2.8	70.3/11.9	71.1/10.3
		MobileNetV2 (ϵ_t=3)	54.8/0.0	55.0/0.0	55.0/0.0	55.8/0.0	57.8/0.0	60.0/0.2	62.6/0.5	68.3/2.7	77.0/12.9	78.8/12.5
		Average (ϵ_t=3)	37.3/0.0	37.5/0.0	37.8/0.0	38.1/0.0	39.3/0.0	41.8/0.1	45.7/0.3	52.6/2.8	63.9/12.5	65.3/11.6
	MI+DI+TI	ResNet50 (ϵ_t=3)	36.2/0.0	37.4/0.0	37.6/0.1	36.9/0.1	36.8/0.0	39.7/0.4	42.3/1.8	47.7/5.4	56.8/10.0	56.8/8.3
		DenseNet161 (ϵ_t=3)	28.0/0.0	28.1/0.0	28.8/0.0	28.8/0.0	29.7/0.0	32.0/0.1	35.1/0.5	41.0/1.8	48.5/4.7	47.9/4.0
		VGG16 (ϵ_t=3)	47.0/0.0	47.3/0.0	47.7/0.0	48.1/0.0	49.9/0.0	51.0/0.3	55.4/1.0	60.4/3.0	69.1/6.6	67.8/5.2
		MobileNetV2 (ϵ_t=3)	56.9/0.0	57.5/0.0	58.0/0.0	58.7/0.0	59.4/0.0	63.3/0.1	65.8/1.0	69.6/2.8	77.7/7.5	76.9/6.9
		Average (ϵ_t=3)	42.0/0.0	42.6/0.0	43.0/0.0	43.1/0.0	43.9/0.0	46.5/0.2	49.6/1.1	54.7/3.2	63.0/7.2	62.4/6.1
ResNet50	I	ResNet18 (ϵ_t=3)	36.2/0.0	36.5/0.0	36.6/0.0	36.7/0.0	37.3/0.0	40.3/0.0	45.6/0.2	52.8/1.9	67.6/13.5	73.0/19.0
		DenseNet161 (ϵ_t=3)	26.1/0.0	26.3/0.0	26.1/0.0	26.5/0.0	27.4/0.0	29.1/0.0	33.5/0.0	38.7/1.7	54.6/13.7	60.1/15.3
		VGG16 (ϵ_t=3)	44.6/0.0	44.9/0.0	45.3/0.0	45.5/0.0	46.5/0.0	48.2/0.0	50.9/0.1	58.8/1.5	72.6/8.5	75.3/10.6
		MobileNetV2 (ϵ_t=3)	54.9/0.0	55.1/0.0	54.2/0.0	55.5/0.0	56.0/0.0	58.0/0.0	62.4/0.0	66.4/1.1	76.9/8.4	80.3/10.9
		Average (ϵ_t=3)	40.5/0.0	40.7/0.0	40.5/0.0	41.0/0.0	41.8/0.0	43.9/0.0	48.1/0.1	54.2/1.6	67.9/11.0	72.2/13.9
	MI	ResNet18 (ϵ_t=3)	48.6/0.0	49.1/0.0	48.2/0.0	48.0/0.0	46.8/0.0	46.9/0.0	51.1/0.4	54.7/2.3	66.5/8.6	71.1/12.0
		DenseNet161 (ϵ_t=3)	26.9/0.0	26.8/0.0	27.9/0.0	28.6/0.0	29.4/0.0	30.9/0.0	36.3/0.3	40.6/1.9	54.4/8.7	57.6/8.7
		VGG16 (ϵ_t=3)	46.4/0.0	46.7/0.0	47.0/0.0	47.8/0.0	47.5/0.0	49.4/0.0	53.5/0.4	60.2/2.1	72.1/6.1	74.0/7.2
		MobileNetV2 (ϵ_t=3)	55.7/0.0	56.1/0.0	56.6/0.0	57.1/0.0	58.0/0.0	60.5/0.2	64.3/0.3	67.4/1.2	77.2/5.6	80.0/7.8
		Average (ϵ_t=3)	44.4/0.0	44.7/0.0	44.9/0.0	45.4/0.0	45.4/0.0	46.9/0.1	51.3/0.3	55.7/1.9	67.5/7.2	70.7/8.9
	DI	ResNet18 (ϵ_t=3)	37.0/0.0	37.1/0.0	37.3/0.0	37.9/0.0	39.5/0.0	42.1/0.0	48.7/0.5	55.2/2.5	70.3/13.9	74.4/16.4
		DenseNet161 (ϵ_t=3)	26.7/0.0	26.6/0.0	27.1/0.0	27.7/0.0	27.7/0.0	29.4/0.0	34.3/0.0	39.2/1.6	51.4/7.9	53.1/7.6
		VGG16 (ϵ_t=3)	45.3/0.0	45.6/0.0	46.2/0.0	46.2/0.0	47.2/0.0	48.7/0.0	52.4/0.3	58.6/1.7	71.1/5.3	71.0/6.5
		MobileNetV2 (ϵ_t=3)	55.4/0.0	56.4/0.0	56.2/0.0	56.3/0.0	57.4/0.0	59.5/0.1	63.9/0.2	67.3/1.2	75.1/4.8	77.9/5.9
		Average (ϵ_t=3)	41.1/0.0	41.4/0.0	41.7/0.0	42.0/0.0	43.0/0.0	44.9/0.0	49.8/0.2	55.1/1.8	67.0/8.0	69.1/9.1
	TI	ResNet18 (ϵ_t=3)	36.1/0.0	36.7/0.0	36.4/0.0	36.4/0.0	37.9/0.0	39.5/0.0	47.1/0.2	53.6/1.6	67.4/13.2	72.6/17.4
		DenseNet161 (ϵ_t=3)	26.0/0.0	26.1/0.0	26.8/0.0	27.4/0.0	27.5/0.0	29.8/0.0	34.8/0.0	40.1/1.6	55.9/14.7	61.2/16.9
		VGG16 (ϵ_t=3)	45.5/0.0	45.0/0.0	45.2/0.0	45.5/0.0	46.9/0.0	48.8/0.0	53.1/0.1	59.8/1.7	73.0/8.6	75.2/10.3
		MobileNetV2 (ϵ_t=3)	54.9/0.0	56.1/0.0	55.1/0.0	55.8/0.0	56.2/0.0	57.6/0.0	64.4/0.1	67.4/1.3	77.9/8.5	80.8/11.1
		Average (ϵ_t=3)	40.6/0.0	41.0/0.0	40.9/0.0	41.3/0.0	42.1/0.0	43.9/0.0	49.9/0.1	55.2/1.6	68.5/11.2	72.5/13.9
	MI+DI+TI	ResNet18 (ϵ_t=3)	48.2/0.0	48.9/0.0	48.0/0.0	49.0/0.0	48.3/0.0	49.3/0.0	53.8/1.0	58.5/3.3	70.3/9.0	72.4/10.5
		DenseNet121 (ϵ_t=3)	28.5/0.0	27.8/0.0	28.8/0.0	30.3/0.0	30.5/0.0	33.6/0.0	37.9/0.2	42.8/2.1	53.8/7.2	54.6/6.4
		VGG16 (ϵ_t=3)	47.8/0.0	47.9/0.0	48.1/0.0	49.9/0.0	48.8/0.0	52.6/0.1	55.9/0.8	61.6/2.2	71.7/4.8	72.7/5.2
		MobileNetV2 (ϵ_t=3)	57.6/0.0	58.3/0.0	57.8/0.0	60.4/0.0	59.4/0.0	63.3/0.1	66.4/0.6	70.9/1.7	77.9/4.8	78.8/5.4
		Average (ϵ_t=3)	45.5/0.0	45.7/0.0	45.7/0.0	47.4/0.0	46.8/0.0	49.7/0.1	53.5/0.7	58.5/2.3	68.4/6.5	69.6/6.9

Table 12: Full results of Table 5 of the main manuscript showing the attack success rate (%) with standard and robust surrogate models for I-FGSM on ImageNet.

Source Model	Target Model	ϵ_s=0	ϵ_s=3
DenseNet161	Accuracy	77.37	66.98
	ResNet50	48.4/0.7	**63.0/5.3**
	DenseNet121	50.3/1.3	**65.5/9.2**
	VGG16	48.4/0.2	**61.0/1.4**
	MobileNetV2	49.8/0.3	**74.0/5.7**
	Average	49.2/0.6	**65.9/5.4**
MNASNet	Accuracy	60.97	41.83
	ResNet50	19.4/0.0	**41.6/0.4**
	DenseNet121	18.8/0.0	**40.5/0.2**
	VGG16	30.4/0.0	**48.2/0.2**
	MobileNetV2	37.0/0.0	**60.5/1.0**
	Average	26.4/0.0	**47.7/0.5**
ResNeXt	Accuracy	77.38	66.25
	ResNet50	44.7/0.7	**68.9/11.3**
	DenseNet121	36.3/0.2	**66.7/10.0**
	VGG16	41.7/0.2	**60.9/2.1**
	MobileNetV2	45.0/0.1	**73.5/7.6**
	Average	41.9/0.3	**67.5/7.8**
ShuffleNet	Accuracy	64.25	43.32
	ResNet50	20.2/0.0	**41.6/0.4**
	DenseNet121	17.6/0.0	**36.1/0.4**
	VGG16	26.6/0.0	**46.7/0.2**
	MobileNetV2	30.4/0.0	**58.0/0.8**
	Average	23.7/0.0	**45.6/0.5**