THE ULTIMATE COMBO: BOOSTING ADVERSARIAL EXAMPLE TRANSFERABILITY BY COMPOSING DATA AUGMENTATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Transferring adversarial examples from surrogate (ML) models to evade target models is a common method for evaluating adversarial robustness in black-box settings. Researchers have invested substantial efforts to enhance transferability. Chiefly, attacks leveraging data augmentation have been found to help adversarial examples generalize better from surrogate to target models. Still, prior work has explored a limited set of augmentation techniques and their composition. To fill the gap, we conducted a systematic, comprehensive study of how data augmentation affects transferability. Particularly, we explored ten augmentation techniques of six categories originally proposed to help ML models generalize to unseen benign samples, and assessed how they influence transferability, both when applied individually and when composed. Our extensive experiments with the ImageNet dataset showed that simple color-space augmentations (e.g., color to greyscale) outperform the state of the art when combined with standard augmentations, such as translation and scaling. Additionally, except for two methods that may harm transferability, we found that composing augmentation methods impacts transferability monotonically (i.e., more methods composed $\rightarrow \geq$transferability)—the best composition we found significantly outperformed the state of the art (e.g., 95.6% vs. 90.9% average transferability from normally trained surrogates to other normally trained models). We provide intuitive, empirically supported explanations for why certain augmentations fail to improve transferability.

1 INTRODUCTION

Adversarial examples—variants of benign inputs minimally perturbed to induce misclassification at test time—have emerged as a profound challenge to machine learning (ML) (Biggio et al., 2013; Szegedy et al., 2014), calling its use in security- and safety-critical systems into question (e.g., Eykholt et al. (2018)). Many attacks have been proposed to generate adversarial examples in white-box settings, where adversaries are familiar with all the particularities of the attacked model (Papernot et al., 2016). By contrast, black-box attacks enable evaluating the vulnerability of ML in realistic settings, without access to the model (Papernot et al., 2016).

Attacks exploiting the transferability-property of adversarial examples (Szegedy et al., 2014) have received special attention. Namely, as adversarial examples produced against one model are often misclassified by others, transferability-based attacks produce adversarial examples against surrogate (a.k.a. substitute) white-box models to mislead black-box ones. To measure the risk of adversarial examples in black-box settings more accurately, researchers have proposed varied methods to enhance transferability beyond that of white-box attacks (e.g., Lin et al. (2020); Liu et al. (2017)).

Notably, attacks using data augmentation, such as translations (Dong et al., 2019) and scaling of pixel values (Lin et al., 2020), as a means to improve the generalizability of adversarial examples across models have accomplished state-of-the-art transferability rates. Still, to the best of our knowledge, previous transferability-based attacks have studied only four augmentation methods (see Section 3.1), out of many proposed in the data-augmentation literature (Shorten & Khoshgoftaar, 2019), primarily for reducing model overfitting. Hence, the extent to which different data-augmentation types boost transferability, either individually or when combined, remains largely unknown.

To fill the gap, we conducted a systematic, comprehensive study of how data-augmentation methods influence transferability. Specifically, alongside techniques considered in previous work, we studied how ten augmentation techniques pertaining to six categories impact transferability when applied individually or composed (Section 3). Integrating augmentation methods into attacks via a flexible framework we propose (Algorithm 1), we conducted extensive experiments using an ImageNet-compatible dataset and ten models, and measured transferability in diverse settings, including with and without defenses (Sections 4 and 5). Our results offer several interesting insights:

- Simple color-space augmentations outperform state-of-the-art transferability-based attacks when composed with standard augmentations (Section 5.1).
- Transferability has a mostly monotonic relationship with data-augmentation techniques. Except for two augmentation methods that may harm transferability, composing additional augmentation methods either improves of preserves transferability (Section 5.2).
- Out of 2^7 compositions explored, the best composition we found, ULTIMATECOMBO, outperforms state-of-the-art attacks by a large margin (Section 5.3).
- We show empirical support to conjectures we raise concerning when data-augmentation techniques may be counterproductive to transferability (Section 5.4).

2 BACKGROUND AND RELATED WORK

We now review prior evasion attacks on ML (incl. via transferability), and defenses against them.

2.1 EVASION ATTACKS

Researchers have proposed numerous evasion attacks against ML models in general, and deep neural networks (DNNs) in particular. Many attacks assume adversaries have white-box access to models—i.e., adversaries know models' architectures and weights (e.g., Goodfellow et al. (2015); Szegedy et al. (2014); Carlini & Wagner (2017)). These attacks typically leverage first- or second-order optimizations to generate adversarial examples models would misclassify. For example, given an input x of class y, model weights θ, and a loss function J, the Fast Gradient Sign method (FGSM) of Goodfellow et al. (2015), crafts an adversarial example $\hat{x}$ using the loss gradients $\nabla_x J(x, y, \theta)$:

$$\hat{x} = x + \epsilon * \text{sign}(\nabla_x J(x, y, \theta))$$

where $\text{sign}(\cdot)$ maps real numbers to -1, 0, or 1, depending on their sign.

Following FGSM, researchers proposed various advanced attacks. Notably, iterative FGSM (I-FGSM) of Kurakin et al. (2017b) performs multiple gradient-ascent steps, updating $\hat{x}$ iteratively to produce adversarial examples:

$$\hat{x}_{t+1} = \text{Proj}_x^{\epsilon}\left(\hat{x}_t + \alpha \cdot \text{sign}\left(\nabla_x J\left(\hat{x}_t, y, \theta\right)\right)\right)$$

where $\text{Proj}_x^{\epsilon}(\cdot)$ projects the perturbation into ℓ_∞-norm ϵ-ball centered at x, α is the step size, and $\hat{x}_0 = x$. The attacks we study in this work are based on I-FGSM.

In real-world settings, adversaries often do not have white-box access to victim models. Hence, researchers also studied black-box attacks in which adversaries only have query access to models. Certain attack types, such as score- and boundary-based attacks perform multiple queries, often in the order of several thousands, to produce adversarial examples (e.g., Brendel et al. (2018); Ilyas et al. (2019)) By contrast, attacks leveraging *transferability* (e.g., Goodfellow et al. (2015); Szegedy et al. (2014)) avoid querying victim models when generating adversarial examples. Instead, transferability-based attacks create adversarial examples against surrogate white-box models, that are also likely misclassified by other black-box models.

Attempts to explain the transferability phenomenon attribute it to gradient norm of the target model (i.e., its susceptibility to attacks), the smoothness of classification boundaries, and, primarily, the alignment of gradient directions between the surrogate and target models (Demontis et al., 2019; Yang et al., 2021). Said differently, for adversarial examples to transfer, the gradient directions of surrogates need to be similar to those of target models (i.e., attain high cosine similarity).

Algorithm 1 MI-FGSM with data augmentation

Input: Benign sample x; ground-truth label y; loss function $J(\cdot)$; model parameters θ; iterations # T; momentum parameter μ; perturbation size ϵ; data-augmentation method $D(\cdot)$.
Output: Adversarial example $\hat{x}$

$\alpha = \epsilon / T$
$\hat{x}_0 = x$ # Initialize adversarial example
$g_0 = 0$ # Initialize momentum
for $t = 0$ to $T - 1$ **do**
  $\bar{g}_{t+1} = \frac{1}{m} \sum_{i=0}^{m-1} \nabla_x \left(J \left(D(\hat{x}_t)_i, y, \theta \right) \right)$ # Expected loss gradient on augmented samples
  $g_{t+1} = \mu \cdot g_t + \frac{\bar{g}_{t+1}}{\|\bar{g}_{t+1}\|_1}$ # Gradient with momentum
  $\hat{x}_{t+1} = \mathrm{Proj}_x^{\epsilon} \left(\hat{x}_t + \alpha \cdot \mathrm{sign} \left(g_{t+1} \right) \right)$ # Update adversarial example
return $\hat{x} = \hat{x}_T$

Enhancing adversarial example transferability has become an active research area. One category of methods integrates momentum into attacks such as I-FGSM to avoid surrogate-specific optima and saddle points which may hinder transferability (e.g., Dong et al. (2018); Lin et al. (2020)). Another attack category employs specialized losses, such as reducing the variance of intermediate activations (Huang et al., 2019) or the mean loss of model ensembles (Liu et al., 2017), to enhance transferability. Lastly, a prominent family of attacks leverages data augmentation to enhance the generalizability of adversarial examples between models. For instance, Dong et al. (2019) boosted transferability by integrating random translations into I-FGSM, producing translation-invariant attacks, while Xie et al. (2019) improved transferability using random crops, creating size-invariant attacks. Evasion attacks incorporating data augmentation attain state-of-the-art transferability rates (Lin et al., 2020; Wang et al., 2021a). Nonetheless, prior work has only considered a restricted set of data-augmentation methods for enhancing transferability. By contrast, our work aims to investigate the role of data augmentation at enhancing transferability more systematically, by exploring how a more comprehensive set of augmentation types and their compositions affect transferability.

2.2 Defenses

Various defenses have been proposed to mitigate evasion attacks. Adversarial training—a procedure integrating correctly labeled adversarial examples in training—is one of the most practical and effective methods for enhancing adversarial robustness (e.g., Goodfellow et al. (2015); Tramèr et al. (2018)). Other defense methods sanitize inputs prior to classification (e.g., Guo et al. (2018)); attempt to detect attacks (see Tramer (2022)); or seek to certify robustness in ϵ-balls around inputs (e.g., Cohen et al. (2019); Salman et al. (2019)). Following standard practices in the literature (Wang et al., 2021a), we evaluate transferability-based attacks against a representative set of these defense.

3 Data Augmentation for Enhancing Transferability

Data augmentation is traditionally used during training to reduce overfitting and improve generalizability (Shorten & Khoshgoftaar, 2019). Inspired by the original use, transferability-based attacks adopted data augmentation to limit overfitting to surrogate models and produce adversarial examples likely to generalize and be misclassified by victim models. Algorithm 1 depicts a general framework for integrating data augmentation into I-FGSM with momentum (MI-FGSM). In the framework, a method $D(\cdot)$ augments the attack with m variants of the estimated adversarial example at each iteration. Consequently, the adversarial perturbation found by the attack increases the expected loss over transformed counterparts of the benign sample x (i.e., the distribution set by $D(\cdot)$ given x). Note that $D(\cdot)'s$ output may include x.

The framework in Algorithm 1 is flexible, and can admit any data-augmentation method. We use it to describe previous attacks employing data augmentation and to systematically explore new ones. Next, we detail previous attacks, describe data augmentation methods we adopt for the first time to enhance transferability, and explain how these can be combined for best performance.

3.1 PREVIOUS ATTACKS LEVERAGING DATA AUGMENTATION

Previous work explored the following augmentation methods to set $D(\cdot)$.

Translations Using random translations of inputs, Dong et al. (2019) proposed a translation-invariant attack to promote transferability. They also offered an optimization to reduce the attack's time and space complexity by simply convolving the model's gradients (w.r.t. non-translated inputs) with a Gaussian kernel. While we use this optimization in the implementation for the interest of efficiency, we highlight that the attack can be well-captured by our framework.

Diverse Inputs Xie et al. (2019) proposed a size-invariant attack. Their augmentation procedure samples random crops from $\hat{x}_t$ that are later resized per the model's input dimensionality.

Scaling Pixels Lin et al. (2020) showed that adversarial perturbations invariant to scaling pixel values transfer with higher success between DNNs. In their case, $D(\cdot)$ produces m samples such that $D(x)_i = \frac{x}{2^i}$ for $i \in \{0, 1, ..., m-1\}$, where m=5 by default.

Admix Wang et al. (2021a) assumed that the adversary has a gallery of images from different classes and adopted augmentations similar to *MixUp* (Zhang et al., 2018a). For each sample x' from the gallery, Admix augments attacks with m (typically set to 5) samples, such that $D(x, x')_i = \frac{1}{2^i} \cdot (\hat{x}_t + \eta \cdot x')$, where $i \in \{0, 1, ..., m-1\}$, and $\eta \in [0, 1]$ is set to 0.2 by default. Notably, Admix degenerates to pixel scaling when $\eta = 0$.

The leading transferability-based attacks compose *(1)* diverse inputs, scaling, and translations (Lin et al.'s (2020) DST-MI-FGSM attack); or *(2)* Admix, diverse inputs, and translation (Wang et al.'s (2021a) Admix-DT-MI-FGSM attack). We describe how these compositions operate in Section 3.3.

3.2 NEW AUGMENTATIONS FOR ENHANCING TRANSFERABILITY

While prior work studied the effect of spatial transformations (i.e., translations and diverse inputs), pixel scaling, and mixing on transferability, a substantially wider range of data-augmentation methods exist (Shorten & Khoshgoftaar, 2019). Yet, the impact of these on transferability remains unknown. To fill the gap, we examined Shorten & Khoshgoftaar's (2019) recent survey on data augmentation for reducing overfitting in deep learning and identified ten representative methods of six categories that may boost transferability. We present them in what follows, one category at a time.

3.2.1 COLOR-SPACE TRANSFORMATIONS

Potentially the simplest of all augmentation types are those applied in color-space. Given images represented as three-channel tensors, methods in this category manipulate pixel values only based on information encoded in the tensors. We evaluate four color-space transformations.

Color Jitter (CJ) This transformation applies random color manipulation (Wu et al., 2015). In this work, we consider random adjustments of pixel values within a pre-defined range in terms of hue, contrast, saturation, and brightness around original values.

Fancy Principle Component Analysis (fPCA) Used in AlexNet (Krizhevsky et al., 2017), fPCA adds noise to the image proportionally to the variance in each channel. Given an RGB image, fPCA adds the following quantity to each image pixel:

$$[\mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3] \left[\alpha_1\lambda_1, \alpha_2\lambda_2, \alpha_3\lambda_3\right]^T,$$

where p_i and λ_i are the ith eigenvector and eigenvalue of the of $3\times$ 3 covariance matrix of RGB pixels, respectively, and α_i is sampled once per image from Gaussian distribution $\mathcal{N}(0, 0.1)$.

Channel shuffle (CS) Included in ShuffleNet training (Zhang et al., 2018b), CS simply swaps the orders of the image's RGB channels at random.

Greyscale (GS) This simple augmentation converts images into greyscale (replicating it three times to obtain an RGB representation). Mathematically, the conversion is calculated by $\omega_R \cdot x^R + \omega_G \cdot x^G + \omega_B \cdot x^B$, where x^R, x^G, and x^B, correspond to the RGB channels, respectively, and ω_R, ω_G, and ω_B, all $\in [0, 1]$, denote the channel weights, and sum up to 1.

3.2.2 RANDOM ERASING

Inspired by dropout regularization, ***random erasing*** **(RE)** helps ML models focus on descriptive features of images and promote robustness to occlusions (Zhong et al., 2020). To do so, randomly selected rectangular regions in images are replaced by masks composed of random pixel values. Similarly to RE, ***CutOut*** masks out regions of inputs to improve DNNs' accuracy (DeVries & Taylor, 2017). The main difference from e is that CutOut uses fixed masking values, and may perform less aggressive masking when selected regions lie outside the image.

3.2.3 KERNEL FILTERS

Convolving images with kernels of different types can produce certain effects, such as blurring (via Gaussian kernels), sharpening (via edge filters), or edge enhancement. We study the effect of ***sharpening (Sharp)*** on transferability with edge-enhancement filters.

3.2.4 MIXING IMAGES

As a form of vicinal risk minimization, some augmentation methods mix images together, creating virtual examples for training. The MixUp method, the cornerstone behind Admix, computes weighted sums of images (Zhang et al., 2018a). By contrast, we consider ***CutMix***, which replaces a region within one image with a region from another image picked from a gallery (Yun et al., 2019).

3.2.5 NEURAL TRANSFER

Augmentations building upon ***neural transfer (NeuTrans)*** preserve images' semantic content while changing their style. We consider Gatys et al.'s (2015) generative model for artistic style transfer. Particularly, we use their approach to transfer image styles to that of Picasso's 1907 self-portrait.

3.2.6 META-LEARNING-INSPIRED AUGMENTATIONS

Meta-learning is a subfield of ML studying how ML algorithms can optimize other learning algorithms (Hospedales et al., 2021). In the context of data augmentation, algorithms such as ***AutoAugment*** have been proposed to train controllers to select an appropriate augmentation method to avoid overfitting (Cubuk et al., 2019). We use the pre-trained AutoAugment controller, encoded as a recurrent neural network, to select augmentation methods and their magnitude from a set of 13 augmentation methods.

3.3 COMPOSING AUGMENTATIONS

There are two primary ways to compose data-augmentation methods in attack algorithms, namely: *parallel* and *serial* composition. In parallel composition, augmentation methods are applied independently on the input, and their outputs are aggregated (i.e., taking their union) to augment attacks. By contrast, serial composition applies augmentation methods sequentially, one after the other, where the first method operates on the original sample, and each of the subsequent augmentation functions is applied on its predecessor's output. State-of-the-art attacks (i.e., DST-MI-FGSM and Admix-DT-MI-FGSM) use serial composition. By contrast, in this work, we consider a substantially larger number of augmentation methods, which may lead to prohibitive memory requirements in the case of serial composition. Additionally, because the order of applying certain augmentations matters (e.g., GS then CutMix leads to different outcome that CutMix followed by GS), exploring a meaningful number of serial compositions (out of an order of 10! possibilities) becomes virtually impossible. Accordingly, we mainly consider parallel composition between data-augmentation methods. The *only* settings we use serial composition are when combining augmentations with translation, scaling, and diverse inputs, for consistency with prior work (e.g., Wang et al. (2021a)). A few serial compositions we tested were significantly outperformed by their parallel counterparts. While inconclusive, the results hint that serially composing augmentations may not be a promising direction for enhancing transferability.

4 EXPERIMENTAL SETUP

Now we turn to the setup of our experiments, including the data, models, and attack configurations.

Data Following prior work (Dong et al., 2019; Gao et al., 2020), we used an ImageNet-compatible dataset (Russakovsky et al., 2015) originally collected for the NeurIPS 2017 evasion-attack competition.[1] The dataset contains 1,000 images, covering 430 ImageNet classes, with a maximum of five images per class. These images are mostly classified correctly—the normally trained classifiers we considered (see below) obtain 96.7% benign accuracy.

Models We used ten DNNs to transfer attacks from (as surrogates) and to (as targets). Six models were normally trained, while others were adversarially trained. Specifically, for normally trained models, we selected: Inception-v3 (Inc-v3) (Szegedy et al., 2016)); Inception-v4 (Inc-v4); Inception-ResNet-v2 (IncRes-v2) (Szegedy et al., 2017)); ResNet-v2-50 (Res-50); ResNet-v2-101 (Res-101); and ResNet-v2-152 (Res-152) (He et al., 2016). For adversarially trained models, we selected: Inception-v3-adv (Inc-v3$_{adv}$) (Kurakin et al., 2017a); ens3-Inception-v3 (Inc-v3$_{ens3}$); ens4-Inception-v3 (Inc-v3$_{ens4}$); and ens-adv-Inception-ResNet-v2 (IncRes-v2$_{ens}$) (Tramèr et al., 2018). These models constitute a superset of ones considered in prior work (Wang et al., 2021a). We obtained the models' PyTorch implementations and weights from a public GitHub repository.[2]

Attack Parameters We tested standard attack configurations, in line with prior work (Dong et al., 2018; Wang et al., 2021a). More precisely, we evaluated MI-FGSM-based attacks, bounded in ℓ_∞-norm, with a maximum perturbation norm of $\epsilon = \frac{16}{255}$. The attacks were not targeted toward specific classes. Accordingly, we evaluated them via *transferability rates*—the percentages of attempts at which adversarial examples created against surrogates were misclassified by victims. We set the MI-FGSM decay factor μ=1.0 (similarly to Wang et al. (2021b)), and the number of iterations T=10. We used DST-MI-FGSM and Admix-DT-MI-FGSM, two state-of-the-art transferability-based attacks, as baselines. The exact parameters used for all augmentation methods are reported in Appendix A. Using our parameters with the baselines, we managed to reproduce and improve transferability rates reported in prior work (Wang et al., 2021a).

We measured transferability rates from each normally trained model, as surrogate, to every other model, as target. Furthermore, for adversarially trained target models, we employed Liu et al.'s (2017) method and evaluated transferability from an ensemble of normally trained models as surrogates. Specifically, we considered an ensemble containing Inc-v4, Res-50, Res-101, and Res-152.

5 EXPERIMENTAL RESULTS

This section summarizes our findings. We start by evaluating individual augmentation methods and standard combinations with scaling, diverse inputs, and translations (Section 5.1). We then turn to analyzing *all possible* compositions between different augmentation types to assess whether transferability typically improves when considering additional augmentations (Section 5.2). Our analysis helped us identify the best performing composition for boosting transferability, denoted by ULTIMATECOMBO, outperforming state-of-the-art attacks. Section 5.3 reports rigorous comparisons between ULTIMATECOMBO and the baselines, including against defended models. Finally, we help develop intuition for when augmentations may or may not help improve transferability (Section 5.4).

5.1 COLOR-SPACE AUGMENTATIONS OUTPERFORM THE STATE OF THE ART

Initially, we evaluated transferability integrating a single augmentation at a time in attacks, or when composing individual augmentations with diverse inputs, scaling, and translation (DST), as is standard (Lin et al., 2020; Wang et al., 2021a). We found that considering each of the ten augmentations individually does *not* lead to competitive performance with the baselines. However, composing individual augmentations with DST enhanced transferability markedly. Surprisingly, augmentations in color-space fared particularly well, outperforming the baselines and advanced augmentation methods in most cases. Specifically, composing GS with DST (GS-DST-MI-FGSM attack) performed

[1] https://bit.ly/3fq4pN6

[2] https://github.com/ylhz/tf_to_pytorch_model

best in this setting. Table 1 reports the transferability rates from four normally trained models to other models. It can be immediately seen that GS-DST-MI-FGSM attains higher transferability than the baselines (93.6% vs. ≤90.9%, on avg.). The same trend holds when transferring adversarial examples from normally trained models and the ensemble of models to adversarially trained victims (Tables 2 and 3, respectively). According to a paired t-test, the differences between GS-DST-MI-FGSM and the baselines across different surrogate and target models were statistically significant (p <0.01). Transferability results using other surrogate models are included in Appendix B.

Model	**Attack**	**Inc-v3**	**Inc-v4**	**Res-50**	**Res-101**	**Res-152**	**IncRes-v2**
Inc-v3	Admix-DT-MI-FGSM	*99.5*	92.3	88.5	87.0	85.3	90.9
	DST-MI-FGSM	**100.0**	92.9	89.5	87.2	86.4	91.2
	GS-DST-MI-FGSM	**100.0**	*95.6*	*93.7*	*91.8*	*90.9*	*94.9*
	UltimateCombo	**100.0**	**98.0**	**95.1**	**94.3**	**92.7**	**97.1**
Inc-v4	Admix-DT-MI-FGSM	93.7	99.3	86.7	84.9	84.7	89.6
	DST-MI-FGSM	94.4	**100.0**	90.2	88.1	88.2	92.8
	GS-DST-MI-FGSM	*96.5*	**100.0**	*94.1*	*92.5*	*93.0*	*95.4*
	UltimateCombo	**98.1**	*99.9*	**94.8**	**95.0**	**94.6**	**96.8**
Res-101	Admix-DT-MI-FGSM	82.6	78.1	93.5	97.4	93.9	79.3
	DST-MI-FGSM	86.7	83.2	97.3	**99.9**	96.6	84.9
	GS-DST-MI-FGSM	*89.0*	*84.8*	*97.6*	*99.8*	*97.7*	*87.6*
	UltimateCombo	**93.0**	**90.4**	**98.1**	99.7	**97.8**	**91.8**
IncRes-v2	Admix-DT-MI-FGSM	93.8	91.9	91.1	90.6	89.5	98.9
	DST-MI-FGSM	95.8	94.2	93.5	92.0	92.4	*99.8*
	GS-DST-MI-FGSM	*96.5*	*95.6*	*95.5*	*94.2*	*94.7*	**100.0**
	UltimateCombo	**98.2**	**97.1**	**96.3**	**96.5**	**95.7**	**100.0**

Table 1: Transferability rates (%) from normally trained surrogates (rows) to normally trained target models (columns) using four attacks. All attacks are black-box, except for when the surrogate and target models are the same (i.e., white-box attacks).

Model	**Attack**	**Inc-v3**$_{adv}$	**Inc-v3**$_{ens3}$	**Inc-v3**$_{ens4}$	**IncRes-v2**$_{ens}$
Inc-v3	Admix-DT-MI-FGSM	84.6	84.3	83.5	70.8
	DST-MI-FGSM	81.3	81.2	77.7	61.4
	GS-DST-MI-FGSM	*87.3*	*88.5*	*85.5*	*72.2*
	UltimateCombo	**88.2**	**88.7**	**86.7**	**72.6**
Inc-v4	Admix-DT-MI-FGSM	82.7	83.3	81.3	73.7
	DST-MI-FGSM	80.6	81.8	80.8	70.5
	GS-DST-MI-FGSM	*87.6*	**89.5**	*87.2*	*78.0*
	UltimateCombo	**88.6**	*89.4*	**88.4**	**78.2**
Res-101	Admix-DT-MI-FGSM	79.5	80.4	78.6	71.2
	DST-MI-FGSM	78.9	78.7	76.7	68.7
	GS-DST-MI-FGSM	*82.3*	*83.7*	*81.1*	*73.2*
	UltimateCombo	**83.5**	**86.7**	**82.8**	**76.8**
IncRes-v2	Admix-DT-MI-FGSM	89.0	89.0	88.7	87.1
	DST-MI-FGSM	87.2	89.2	86.4	82.9
	GS-DST-MI-FGSM	*91.1*	*92.2*	*90.0*	*88.2*
	UltimateCombo	**92.2**	**92.6**	**92.0**	**88.5**

Table 2: Transferability rates (%) from normally trained surrogates (rows) to adversarially trained target models (columns) using four attacks.

5.2 The Monotonicity of Transferability When Adding Augmentations

We wanted to evaluate whether transferability is monotonic in the number of augmentation types considered—i.e., whether composing more techniques increases, or at least does not harm, transferability. To this end, we selected the best performing augmentation method of each of the six categories presented in Section 3.2 as well as DST-MI-FGSM, and evaluated all 2^7 (=128) compositions possible (per Section 3.3). More precisely, we tested every possible combination of GS, CutOut, Sharp, NeuTrans, AutoAugment, Admix, and DST-MI-FGSM. Given a composition, we

Attack	Inc-v3$_{adv}$	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens}$
DST-MI-FGSM	89.0	90.0	87.6	82.4
Admix-DT-MI-FGSM	90.1	90.5	89.4	84.7
GS-DST-MI-FGSM	*92.4*	*93.5*	*92.5*	*88.7*
UltimateCombo	**93.6**	**95.2**	**93.7**	**91.2**

Table 3: Transferability rates (%) from an ensemble of normally trained surrogates (containing Inc-v4, Res-50, Res-101 and Res-152) to adversarially trained target models.

produced adversarial examples against the Inc-v3 DNN as surrogate, and computed the expected transferability rate against all other nine DNNs, both normally and adversarially trained. Then, for every pair of attacks differing only in whether a single augmentation method was incorporated in the composition, we tested whether adding the augmentation method improved transferability.

The results reflected a mostly monotonic relationship between transferability and augmentations. Except for NeuTrans and Sharp, which sometimes harmed transferability when considered within a composition, adding augmentation method increased or preserved transferability. Notably, comparing all compositions enabled us to find that a composition of all seven augmentation methods except for NeuTrans attained the best transferability. We call this composition the UltimateCombo.

5.3 The Most Effective Combination

We evaluated UltimateCombo extensively, testing transferability to normally and adversarially trained DNNs. As shown in Table 1, UltimateCombo obtained higher transferability to normally trained models than the baselines (95.6% vs. $\leq$90.9% avg. transferability) and GS-DST-MI-FGSM, when normally trained models were used as surrogates. Furthermore, UltimateCombo achieved the best performance also when transferring attacks from normally trained to adversarially trained DNNs (Table 2; 86.0% vs. $\leq$81.7% avg. transferability). Transferring adversarial examples crafted by UltimateCombo using an ensemble of models increased transferability further (Table 3; 93.4% avg. transferability). Per a paired t-test, the differences between UltimateCombo and the baselines over all pairs of surrogates and targets considered are statistically significant (p <0.01).

Attack	Bit-Red	NRP	RS	ARS
DST-MI-FGSM	85.3	40.7	84.2	39.8
Admix-DT-MI-FGSM	86.4	39.4	86.6	43.0
UltimateCombo	**87.5**	**47.7**	**88.4**	**43.5**

Table 4: Transferability rates (%) from an ensemble of normally trained surrogates (Inc-v4, Res-50, Res-101 and Res-152) to models defended by provable methods or input transformations.

Besides adversarially trained models, we evaluated UltimateCombo's transferability against four defenses. Two defenses, bit reduction (Bit-Red) (Xu et al., 2018) and neural representation purification (NRP) (Naseer et al., 2020), transform inputs to sanitize adversarial perturbations. The two others, randomized smoothing (RS) (Cohen et al., 2019) and randomized smoothing with adversarial training (ARS) (Salman et al., 2019) offer provable robustness guarantees. We used the defenses with default parameters (see Appendix C), and transferred adversarial examples crafted against the ensemble of normally trained models. Results are shown in Table 4. Similar to other settings, here too, UltimateCombo outperformed the baselines (66.8% vs. $\leq$63.9% avg. transferability).

5.4 When Do Augmentations Fail to Improve Transferability?

While augmentation methods mostly increased transferability, in some cases they were counterproductive. Particularly, NeuTrans and Sharp decreased transferability when composed with certain methods. We conducted simple experiments as a preliminary assessment of two conjectures we had concerning when augmentations may harm transferability.

First, we expected augmentation methods that harm model accuracy on benign samples to be less conducive for transferability. As DNNs fail to generalize to samples produced by such augmentation

Augmentation	Inc-v3	Inc-v4	Res-50	Res-101	Res-152	IncRes-v2	Avg.
None	96.2	97.4	94.5	96.3	95.8	99.8	96.7
CS	94.0	95.7	95.3	94.6	95.4	99.5	95.8
fPCA	91.6	96.8	89.9	92.8	93.6	99.4	94.0
CJ	90.0	92.3	90.3	90.3	91.4	96.8	91.9
Admix	86.7	91.6	86.8	88.9	89.7	94.6	89.7
CutOut	86.5	89.2	85.7	87.2	88.6	92.3	88.2
GS	86.6	90.3	84.7	87.6	86.5	92.7	88.1
AutoAugment	82.9	86.2	82.1	84.3	84.4	89.8	85.0
Sharp	69.5	87.3	71.5	76.7	75.5	90.6	78.5
NeuTrans	24.4	25.4	24.2	27.0	24.0	32.7	26.3

Table 5: Benign accuracy (%) after applying data augmentation methods. Rows are sorted in a descending order of average transferability.

methods, we anticipated that adversarial perturbations informed by augmented samples would not be informative. To support this conjecture, we tested the normally trained DNNs' accuracy on benign samples transformed by each augmentation method. As can be seen from Table 5, the methods least conducive for transferability (NeuTrans and Sharp) harmed the DNN accuracy the most (6.5%–58.7% lower accuracy than other methods), supporting our conjecture.

Augmentation	Inc-v4	Res-50	Res-101	Res-152	IncRes-v2	Avg.
CutOut	0.568	0.583	0.581	0.574	0.591	0.579
CS	0.565	0.578	0.576	0.570	0.590	0.576
None	0.564	0.575	0.573	0.568	0.586	0.573
Admix	0.563	0.575	0.573	0.567	0.586	0.573
CJ	0.560	0.575	0.573	0.568	0.584	0.572
GS	0.559	0.572	0.569	0.563	0.582	0.569
AutoAugment	0.558	0.569	0.567	0.562	0.579	0.567
fPCA	0.560	0.568	0.566	0.561	0.578	0.567
NeuTrans	0.546	0.556	0.554	0.549	0.565	0.554
Sharp	0.548	0.548	0.545	0.540	0.558	0.548

Table 6: Cosine similarities between gradients of benign images computed on Inc-v3 after applying augmentation methods composed with DST-MI-FGSM, and gradients of other normally trained models on benign images. Rows are sorted in a descending order of average cosine similarity.

Prior work demonstrated that gradient alignment between surrogates and targets is needed for transferability (Demontis et al., 2019). Thus, we expected augmentation methods that estimate target model gradients more accurately to increase transferability further. To assess this conjecture, we evaluated the cosine similarity between the gradients of the Inc-v3 model while using augmentations composed with DST applied to benign samples, and the gradients of other normally trained models on (untransformed) benign samples. The results (Table 6) show some support to the conjecture—NeuTrans and Sharp led to lower cosine similarities with target models' gradients. Yet, the differences in cosine similarities between augmentation methods were small ($\leq$0.031, on avg.).

6 Conclusion

Our study uncovered a mostly monotonic relationship between data-augmentation methods and transferability, and helped us identify a composition of data-augmentation methods, UltimateCombo, that outperforms previously proposed methods when integrated into attacks. The resulting attack should be considered as a standard baseline in follow-up work on transferability. In addition for assessing the vulnerability of ML models in black-box settings, it would be interesting to evaluate whether the UltimateCombo-based attack advances methods leveraging adversarial examples for defensive purposes, by deceiving adversaries (e.g., to attain privacy (Cherepanova et al., 2021; Shetty et al., 2018)). Our work also puts forward conjectures for when augmentation techniques are expected to improve transferability, and offers some empirical support. In the future, it would be informative to develop formal explanations.

REPRODUCIBILITY STATEMENT

In the interest of reproducibility, we make our code publicly available at the following repository: `https://tinyurl.com/UltimateComboICLR`.

REFERENCES

Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Šrndić, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion attacks against machine learning at test time. In *Proc. ECML/PKDD*, 2013.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *Proc. ICLR*, 2018.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Proc. IEEE S&P*, 2017.

Valeriia Cherepanova, Micah Goldblum, Harrison Foley, Shiyuan Duan, John Dickerson, Gavin Taylor, and Tom Goldstein. Lowkey: Leveraging adversarial attacks to protect social media users from facial recognition. In *Proc. ICLR*, 2021.

Jeremy Cohen, Elan Rosenfeld, and Zico Kolter. Certified adversarial robustness via randomized smoothing. In *Proc. ICML*, 2019.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. AutoAugment: Learning augmentation policies from data. In *Proc. CVPR*, 2019.

Ambra Demontis, Marco Melis, Maura Pintor, Matthew Jagielski, Battista Biggio, Alina Oprea, Cristina Nita-Rotaru, and Fabio Roli. Why do adversarial attacks transfer? Explaining transferability of evasion and poisoning attacks. In *Proc. USENIX Security*, 2019.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint 1708.04552*, 2017.

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

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

Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proc. CVPR*, 2018.

Lianli Gao, Qilong Zhang, Jingkuan Song, Xianglong Liu, and Heng Tao Shen. Patch-wise attack for fooling deep neural network. In *Proc. ECCV*, 2020.

Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint 1508.06576*, 2015.

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

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. In *Proc. ICLR*, 2018.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Proc. ECCV*, 2016.

Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *IEEE PAMI*, 44(9):5149–5169, 2021.

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

Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. In *Proc. ICLR*, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. *CACM*, 60(6):84–90, 2017.

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. 2017a.

Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Proc. ICLRW*. 2017b.

Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *Proc. ICLR*, 2020.

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

Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *Proc. CVPR*, 2020.

Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *Proc. Euro S&P*, 2016.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015.

Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In *Proc. NeurIPS*, 2019.

Rakshith Shetty, Bernt Schiele, and Mario Fritz. A4NT: Author attribute anonymity by adversarial training of neural machine translation. In *Proc. USENIX Security*, 2018.

Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of big data*, 6(1):1–48, 2019.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *Proc. ICLR*, 2014.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proc. CVPR*, 2016.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, Inception-ResNet and the impact of residual connections on learning. In *Proc. AAAI*, 2017.

Florian Tramer. Detecting adversarial examples is (nearly) as hard as classifying them. In *Proc. ICML*, 2022.

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

Xiaosen Wang, Xuanran He, Jingdong Wang, and Kun He. Admix: Enhancing the transferability of adversarial attacks. In *Proc. ICCV*, 2021a.

Xiaosen Wang, Jiadong Lin, Han Hu, Jingdong Wang, and Kun He. Boosting adversarial transferability through enhanced momentum. In *Proc. BMVC*, 2021b.

Ren Wu, Shengen Yan, Yi Shan, Qingqing Dang, and Gang Sun. Deep image: Scaling up image recognition. *arXiv preprint 1501.02876*, 2015.

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 *Proc. CVPR*, 2019.

Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. In *Proc. NDSS*, 2018.

Zhuolin Yang, Linyi Li, Xiaojun Xu, Shiliang Zuo, Qian Chen, Pan Zhou, Benjamin Rubinstein, Ce Zhang, and Bo Li. TRS: Transferability reduced ensemble via promoting gradient diversity and model smoothness. In *Proc. NeurIPS*, 2021.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. CutMix: Regularization strategy to train strong classifiers with localizable features. In *Proc. ICCV*, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proc. ICLR*, 2018a.

Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In *Proc. CVPR*, 2018b.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proc. AAAI*, 2020.

A AUGMENTATION METHODS' PARAMETERS

We mostly used default or commonly used parameters of augmentation methods. For CJ, we performed random adjustments of image hue $\in [-0.5, 0.5]$, contrast $\in [0.5, 1.5]$, saturation $\in [0.5, 1.5]$, and brightness$\in$ $[0.5, 1.5]$. For CutOut, we replaced values in selected regions with zeros, and the portion of masked areas compared to image dimensions lied in $[0.02, 0.4]$, with aspect ratios $\in [0.4, 2.5]$. In comparison, for RE, the dimension of masked areas relatively to the image dimensions lied in $[0.02, 0.2]$, with aspect ratios $\in$ $[0.3, 3.3]$. For Sharp, we used the following edge-enhancement mask:

$$\begin{bmatrix} -0.5 & -0.5 & -0.5 \\ -0.5 & 5.0 & -0.5 \\ -0.5 & -0.5 & -0.5 \end{bmatrix}.$$

For diverse inputs, images were transformed with probability 0.5. For the Admix operation, consistently with Wang et al. (2021a), we randomly sampled three images from other categories for mixing as part of the Admix-DT-MI-FGSM attack. However, for the interest of computational efficiency, we use only one image for mixing when composing Admix with other augmentation methods. We did not find that mixing with fewer images harmed performance. In fact, it even improved transferability in some cases. Finally, in CutMix, we picked the top left coordinate (r_x, r_y), the width, r_w, and height, r_h, of the region to be replaced, using the formulas:

$$r_x \sim \mathcal{U}(0, W), \quad r_w = W\sqrt{1-\lambda},$$

$$r_y \sim \mathcal{U}(0, H), \quad r_h = H\sqrt{1-\lambda},$$

where $\mathcal{U}$ is the uniform distribution, W is the image width, H is the image height, and λ is a parameter set to 0.5.

In an attempt to enhance transferability further, we optimized the parameters of a few augmentation methods we considered via grid search. Except for the Gaussian kernel's size used in translation-invariant attacks (Dong et al., 2019), we found that the selected parameters had little impact on transferability. Specifically, for translations, after considering Gaussian kernels of sizes $\in \{5 \times 5, 7 \times 7, 9 \times 9\}$, we set the default to 7×7, except for Admix-DT-MI-FGSM, for which the 9×9 kernel performed best. For GS, we found ω_R, ω_G, and ω_B had little impact on transferability, as long as the weight assigned to each channel was >0.1. Accordingly, we set ω_R, ω_G, and ω_B to 0.299, 0.587, and 0.114, respectively, per commonly used values (e.g., in the Python `PyTorch` package[3]). Finally, for CS, we only swapped the blue and green channels, as this led to a minor improvement compared to swapping all three channels.

[3] `https://bit.ly/3ynCyUD`

We clarify that each of our augmentation methods emit the original image once, alongside the transformed images. Moreover, when aggregating the gradients, the gradients of the original and transformed images are assigned equal weights. We tested whether weighting the gradients differently (e.g., assigning higher or lower weight to the original sample) can help improve transferability using the GS method. However, we found that equal weights attained the best results.

B COMPLETE RESULTS

Tables 7 and 8 complement Tables 1 and 2, reporting trasferability rates to normally and adversarially trained models, respectively, from different surrogates than those considered above. Here, we also consider transferring adversarial examples from adversarially trained surrogates.

Model	**Attack**	**Inc-v3**	**Inc-v4**	**Res-50**	**Res-101**	**Res-152**	**IncRes-v2**
Res-50	Admix-DT-MI-FGSM	83.9	80.0	97.8	93.4	93.2	78.9
	DST-MI-FGSM	89.1	84.1	**99.9**	97.3	96.8	84.3
	GS-DST-MI-FGSM	*90.3*	*85.7*	**99.9**	*98.3*	*97.9*	*87.5*
	ULTIMATECOMBO	**94.1**	**91.6**	**99.9**	**99.0**	**98.2**	**90.9**
Res-152	Admix-DT-MI-FGSM	82.9	79.0	92.5	92.3	96.5	77.9
	DST-MI-FGSM	86.7	83.6	95.5	97.1	**99.8**	83.1
	GS-DST-MI-FGSM	*87.7*	*84.3*	*96.7*	**97.6**	**99.8**	*85.3*
	ULTIMATECOMBO	**91.8**	**90.4**	**97.0**	*97.3*	99.5	**90.3**
Inc-v3$_{adv}$	Admix-DT-MI-FGSM	92.5	87.9	89.0	88.2	86.8	88.3
	DST-MI-FGSM	93.6	90.9	91.2	90.6	89.5	91.7
	GS-DST-MI-FGSM	*95.2*	*92.8*	*93.8*	*92.8*	*92.2*	*94.2*
	ULTIMATECOMBO	**97.1**	**95.5**	**95.3**	**94.6**	**94.0**	**95.8**
Inc-v3$_{ens3}$	Admix-DT-MI-FGSM	87.9	83.4	85.3	84.5	83.9	85.3
	DST-MI-FGSM	90.2	85.9	86.7	86.0	85.3	87.3
	GS-DST-MI-FGSM	*93.2*	*90.4*	*90.7*	*90.0*	*90.1*	*90.7*
	ULTIMATECOMBO	**94.5**	**92.1**	**93.2**	**92.3**	**92.1**	**92.9**
Inc-v3$_{ens4}$	Admix-DT-MI-FGSM	86.0	80.3	81.4	82.3	81.1	80.7
	DST-MI-FGSM	88.8	85.6	83.4	85.9	83.7	84.4
	GS-DST-MI-FGSM	*92.8*	*88.4*	*90.2*	*89.5*	*87.8*	*88.5*
	ULTIMATECOMBO	**94.5**	**90.2**	**92.0**	**91.0**	**91.1**	**91.5**
IncRes-v2$_{ens}$	Admix-DT-MI-FGSM	82.8	80.4	81.8	79.5	80.3	83.5
	DST-MI-FGSM	85.5	82.9	84.8	84.5	83.9	88.1
	GS-DST-MI-FGSM	*90.1*	*88.2*	**91.0**	*89.7*	*88.6*	*91.6*
	ULTIMATECOMBO	**91.7**	**90.9**	*90.9*	**90.9**	**89.7**	**93.3**

Table 7: Transferability rates (%) from various surrogates (rows) to normally trained target models (columns) using four attacks. All attacks are black-box, except for when the surrogate and target models are the same (i.e., white-box attacks). Note the different surrogate models compared to Table 1.

C DEFENSE PARAMETERS

We used standard parameters when attacking defenses. For RS, we used a normally trained ResNet-50 and set σ to 0.25, following Cohen et al. (2019). For ARS, the target model was ResNet-50 trained with isotropic Gaussian-noise augmentations (sampled from $\mathcal{N}(0, 0.25)$), and σ was set to 0.25 during prediction, per Salman et al. (2019). In both cases, we used 10,000 noisified samples during inference. For Bit-Red, we used a squeezer with bit-depth of one, in accordance with Xu et al. (2018). Finally, we used the default NRP parameters and pre-trained model from the official GitHub repository (Naseer et al., 2020).

Model	**Attack**	**Inc-v3$_{adv}$**	**Inc-v3$_{ens3}$**	**Inc-v3$_{ens4}$**	**IncRes-v2$_{ens}$**
Res-50	Admix-DT-MI-FGSM	79.1	78.0	77.1	68.0
	DST-MI-FGSM	76.1	77.7	75.6	63.6
	GS-DST-MI-FGSM	*81.4*	*83.6*	*78.9*	*70.4*
	UltimateCombo	**85.1**	**85.7**	**83.4**	**74.0**
Res-152	Admix-DT-MI-FGSM	77.7	78.2	75.9	*71.9*
	DST-MI-FGSM	75.0	78.4	75.7	70.1
	GS-DST-MI-FGSM	*79.6*	*82.1*	*79.0*	71.3
	UltimateCombo	**83.6**	**85.0**	**82.5**	**74.8**
Inc-v3$_{adv}$	Admix-DT-MI-FGSM	98.3	93.0	92.2	87.4
	DST-MI-FGSM	99.7	93.0	93.5	85.4
	GS-DST-MI-FGSM	**99.9**	*95.1*	*94.1*	*88.5*
	UltimateCombo	**99.9**	**96.1**	**96.2**	**91.0**
Inc-v3$_{ens3}$	Admix-DT-MI-FGSM	88.3	98.2	91.6	84.6
	DST-MI-FGSM	87.8	**99.9**	91.9	83.0
	GS-DST-MI-FGSM	*91.6*	**99.9**	*93.4*	*85.8*
	UltimateCombo	**94.0**	**99.9**	**94.7**	**89.6**
Inc-v4$_{ens4}$	Admix-DT-MI-FGSM	88.3	90.3	98.7	83.4
	DST-MI-FGSM	88.2	91.8	99.8	83.8
	GS-DST-MI-FGSM	*91.8*	*94.1*	**99.9**	*86.9*
	UltimateCombo	**93.4**	**95.4**	**99.9**	**90.2**
IncRes-v2$_{ens}$	Admix-DT-MI-FGSM	86.2	87.7	88.6	96.3
	DST-MI-FGSM	88.2	89.4	89.8	98.7
	GS-DST-MI-FGSM	*92.0*	*93.4*	*92.7*	*99.1*
	UltimateCombo	**92.5**	**93.5**	**93.1**	**99.3**

Table 8: Transferability rates (%) from various surrogates (rows) to adversarially trained target models (columns) using four attacks. Note the different surrogate models compared to Table 2