IMPROVING ADVERSARIAL TRANSFERABILITY WITH WORST-CASE AWARE ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Generating adversarial examples with high transferability is key to practical black-box attack scenarios, where the attacker has limited or no information about target models. While previous works mainly deal with input transformation or optimization process to reduce overfitting on a surrogate model and enhance transferability, we find that well-designed model manipulation can provide complementary gain to existing methods. We propose Worst-case Aware Attack (WAA), a simple effective method that provides access to a virtual ensemble of models to mitigate overfitting on a specific model during the adversarial example generation process. Specifically, WAA formulates max-min optimization to seek adversarial examples that are robust against the worst-case models, which are created by adding per-example weight perturbation to the source model towards the direction of weakening the adversarial sample in question. Unlike other model manipulation methods, WAA does not require multiple surrogate models or architecture-specific knowledge. Experimental results on ImageNet demonstrate that WAA can be incorporated with a variety of existing methods to consistently improve transferability over different settings, including naturally trained models, adversarially trained models, and adversarial defenses.

1 INTRODUCTION

Adversarial attacks aim to generate a small perturbation on examples that causes unintended results on target model; due to the wide existence of such perturbations in modern Deep Neural Networks (DNNs), adversarial attacks have received growing attention and have many useful applications such as evaluating of the robustness of DNNs (Carlini et al., 2019; Croce et al., 2020; Croce & Hein, 2020), understanding the underlying vulnerability of the model (He et al., 2018; Ilyas et al., 2019; Ignatiev et al., 2019), and to design defense mechanisms (Madry et al., 2017; Athalye et al., 2018; Kurakin et al., 2018; Zhang et al., 2019; Wang et al., 2019). Broadly, there are two classes of adversarial attack scenario: white- and black-box attacks. In the white-box scenario, adversaries have full access to the target model including the model architecture and parameters, thus can directly exploit the vulnerability through backpropagation. On the other hand, black-box scenarios provide limited or no information about the target, which leads to more practical yet challenging settings.

In the black-box setting, a common approach to generate attacks is employing a surrogate (or source) model as a proxy of the target model; it is widely observed that the adversarial examples generated from one model (source model) can easily transfer to the others (target models) (Liu et al., 2017). However, transferability of the black-box adversary is also highly dependent on various factors, such as the choice of surrogate models (Wu et al., 2018) or optimization strategies (Dong et al., 2018; Wang & He, 2021), since the adversary can easily overfit to the source model thus no longer effective on the target. Hence, designing a method that prevents adversarial examples from overfitting to enhance transferability is key to effective black-box attacks.

Common practices to avoid overfitting in black-box scenarios include input transformation (Xie et al., 2019; Dong et al., 2019; Lin et al., 2020; Wang et al., 2021b), optimization (Dong et al., 2018; Wang & He, 2021; Wang et al., 2021c; Huang & Kong, 2022), and feature-aware approaches (Huang et al., 2019; Li et al., 2020a; Wang et al., 2021d). These methods often contribute to different aspects of the attack generation process–input, optimization, and model–thus can be incorporated together to further enhance transferability. Compared to these approaches, however, the problem has been rela-

tively less investigated from the perspective of model manipulation, which directly manipulates the model parameters to create augmentations of the surrogate model. Designing a model manipulation for adversarial attacks is challenging since the impact of manipulation typically varies depending on the model architecture. Hence, existing works (Wu et al., 2020a; Li et al., 2020b) rely on a specific class of architectures that most source models share in common, *e.g.*, skip-connection, but the dependence induces limited applicability or performance gain. We question whether an adversary can manipulate the source model in a model-agnostic way to alleviate overfitting.

In this paper, we show that introducing carefully designed *per-example* perturbations to the model parameters can serve as an effective augmentation to the surrogate model, hence improving the transferability of the black-box adversary by mitigating overfitting. To this end, we characterize the desirable class of perturbations that can be useful in terms of model augmentation, and derive the optimization objective for adversarial attack that account for the augmented models. Our method coined **Worst-case Aware Attack** (WAA) follows a max-min optimization of the loss to generate an attack by alternating between maximization with respect to the data perturbation and minimization with respect to the weight perturbation. Throughout the optimization, the adversarial example encounters multiple weight-perturbed models and avoids overfitting to the source model. Compared to the prior works on model manipulation (Guo et al., 2020; Naseer et al., 2021), our method does not rely on any prior assumptions on model architectures and is generally applicable to various models. Experimental results confirm the effectiveness of WAA in various scenarios, outperforming the previous model manipulation approach, and providing complementary gains to the existing methods in enhancing transferability of black-box adversary.

2 METHODOLOGY

2.1 PRELIMINARIES

Given an image sample x with class label y, an adversarial attack aims to find an indistinguishable adversarial sample $x^{adv} \in \{x^{adv} | \|x^{adv} - x\|_\infty \leq \epsilon\}$[1] that is misclassified by the target classification model h, i.e., $h(x^{\text{adv}}) \neq y$[2]. We address the black-box scenario where adversaries have no access to the target model h but instead have white-box access to a source (surrogate) model f parameterized by w. A common practice of black-box adversary is to generate a white-box adversarial example on the source model and transfer it to the target model (Szegedy et al., 2013). The white-box attack on the source model has the following objective:

$$\arg\max_{x^{adv}} J(x^{adv}, y; w) \quad \text{s.t.} \|x^{adv} - x\|_\infty \leq \epsilon, \tag{1}$$

where $J(\cdot; w)$ is a loss function, e.g., cross entropy, of the source model with parameters w. The optimization problem (Eq. 1) is generally solved by applying gradient methods (Goodfellow et al., 2014), such as the widely used I-FGSM (Kurakin et al., 2018). I-FGSM iteratively applies gradient update T times with step size $\alpha = \epsilon/T$ as follows:

$$x^{adv}_{t+1} = x^{adv}_t + \alpha \cdot \text{sign}(\nabla_{x^{adv}_t} J(x^{adv}_t, y; w)), \tag{2}$$

where the attack begins at $x^{adv}_0 = x$ and x^{adv}_T becomes the output adversarial example.

However, optimizing adversarial attacks solely based on a single surrogate model can easily suffer from severe overfitting, in which case its effect on the target model quickly diminishes as it is not transferable across models. Thus, several works have proposed methods to avoid overfitting problem and improve transferability, which we briefly introduce below.

Optimization algorithms Momentum Iterative method (MI) (Dong et al., 2018) adds a momentum term into the iterative optimization procedure to escape from poor local maxima and stabilize update directions, which turns out to improve transferability:

$$x^{adv}_{t+1} = x^{adv}_t + \alpha \cdot \text{sign}(g_{t+1}), \quad g_{t+1} = \mu \cdot g_t + \frac{\nabla_{x^{adv}_t} J(x^{adv}_t, y; w)}{\|\nabla_{x^{adv}_t} J(x^{adv}_t, y; w)\|_1}. \tag{3}$$

[1]Here, $\|\cdot\|_\infty$ indicates the L_∞ norm.

[2]For targeted adversarial attacks, they aim $h(x^{adv}) = y_t$ for some given target label $y_t \neq y$.

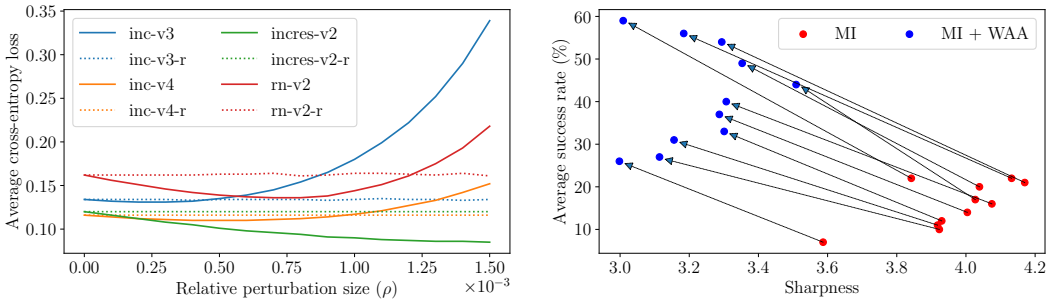

Figure 1: **(Left) Average loss changes by the relative perturbation size ρ.** We experiment how the (average) loss changes by two types of weight perturbations: worst-case by inner-loop of Eq 6 and random perturbation (denote with -r) inside $S_\rho(w)$. We report the average cross-entropy loss over 100 randomly selected images in the ImageNet validation set. It verifies that perturbations in $S_\rho(w)$ satisfy the loss-preserving property with proper choice of ρ ($\rho \leq 5 \times 10^{-4}$). **(Right) Change of the average success rate and sharpness by WAA.** We confirm that integration of WAA to MI-FGSM with different momentum parameters (each point) consistently reduces the sharpness while improving the success rate. We explain more experimental details in the appendix.

The gradients are accumulated over multiple steps with a momentum factor μ. Nesterov Iterative (NI) method (Lin et al., 2020) later extends the optimization of Eq. 3 by replacing the momentum update with Nesterov's accelerated gradient (NAG) (Nesterov, 1983) to achieve further improved transferability.

Input Transformations A line of work focuses on input transformations based on the idea that the composition of input transformation and the source model can serve as a model augmentation to reduce overfitting. Diverse Inputs (DI) (Xie et al., 2019) suggests random-resizing and padding; Translation-Invariant (TI) (Dong et al., 2019) method proposes a set of translations implemented by applying a convolution kernel on the gradient of the image given; Scale-Invariant (SI) (Lin et al., 2020) method suggests scaled copies of the image; Admix (Wang et al., 2021b) replaces SI with mixup-style augmentations. Most of these methods mentioned above can be combined together or with other adversarial attacks to produce stronger baselines of highly transferable adversarial examples.

2.2 Worst-case Aware Attack

This section introduces our method to improve the transferability of the black-box adversary through model augmentation. Our key idea is simple: we propose to directly augment the surrogate model by applying additive perturbations to the model parameters. Augmenting the model through weight perturbation is simple yet effective, as it can create a virtual ensemble from a single surrogate model and is generally applicable agnostic to architectures. However, choosing such perturbations requires careful consideration since arbitrary perturbations can easily deteriorate the model hence introducing noisy signals in optimization of the adversarial attack. To restrict the class of augmented models to be a valid proxy of the surrogate model, we revise the concept of loss-preserving transformation in (Lin et al., 2020) and define a *loss-preserving weight perturbation* as follows:

Definition 1 (Loss-preserving weight perturbation) *An input $x \in \mathcal{X}$ with its ground-truth label y, and a classifier $f(x; w)$ parameterized by w are given. If an additive weight perturbation v to w satisfies $J(x, y; w + v) \approx J(x, y; w)$ for any $x \in \mathcal{X}$ and the cross-entropy loss $J(x, y; w)$, we say v is a loss-preserving weight perturbation.*

The definition characterizes the valid augmentations as the ones that preserve the similar classification performance to the original model. While the theoretical characterization of full set of such perturbations is challenging, we can easily characterize their subsets with simple heuristic. Specifically, we characterize the (subset of) loss-preserving weight perturbations as a set:

$$S_\rho(w) = \{v | \forall l,\ \|v_l\|_2 \leq \rho \|w_l\|_2\}, \tag{4}$$

where l is the layer index and ρ is a constant characterizing the magnitude of the perturbation v. The perturbation bound of each layer is determined adaptively based on the norm of the weight w_l to consider layer-wise scale variance. Eq. 4 is based on a simple heuristic that constraining the norm of perturbation can prevent the augmented model from much diverging from the original one. As shown in Figure 1, such perturbations within a small bound of $\rho = 5 \times 10^{-4}$ do not have much impact on loss over a wide range of source models, which verifies that Eq. 4 can generally characterize a valid loss-preserving perturbations in Definition 1.

By applying the empirical loss-preserving weight perturbations $S_\rho(w)$ discussed above, the objective of optimizing adversarial examples on the set of weight-augmented models becomes:

$$\arg\max_{x^{adv}} \mathbb{E}_{v \in S_\rho(w)}[J(x^{adv}, y; w + v)] \quad \text{s.t. } \|x^{adv} - x\|_\infty \leq \epsilon. \tag{5}$$

Eq. 5 extends the objective of black-box adversary in Eq. 1 with the augmented weights $w + v$ constructed by weight perturbations $v \in S_\rho(w)$. Optimizing the expectation of the above equation is expected to be more robust against the overfitting than optimizing over the fixed surrogate model (Eq. 1) in principle, yet we observe that the improvement is marginal. It is because the loss-preserving weight perturbation is a necessary but not a sufficient condition to characterizes useful perturbations in terms of augmentation; indeed, Eq 4 loosely characterizes such useful perturbations that improve transferability, and contains many trivial solutions of Definition 1 *i.e.*, near-duplicates of the non-perturbed model. Hence, we propose applying the worst-case optimization using the lower-bound of the expectation in Eq. 4 to filter out the effect of degenerate solutions in $S_\rho(w)$ by:

$$\arg\max_{x^{adv}} \min_{v \in S_\rho(w)} J(x^{adv}, y; w + v) \quad \text{s.t. } \|x^{adv} - x\|_\infty \leq \epsilon. \tag{6}$$

Since it is a lower-bound, optimization of Eq. 6 eventually corresponds to the optimization of Eq. 5 over all $v \in S_\rho(w)$, while avoiding the degenerated solutions in $S_\rho(w)$ in generating adversarial attack x^{adv} at the outer maximization. Also, while Eq. 6 is the minimization problem with respect to the weight perturbation v, it still functions as ensemble in practice since the adversarial example x^{adv} keeps changing during the alternating optimization process (Section 2.3). Our experiments show that our objective function (Eq. 6) consistently improves the transferability of adversarial attack over the baseline (Eq. 1) and random augmentation (Eq. 5) (Section 4).

Connection with sharpness-aware optimization (Foret et al., 2020) Rewriting our objective function, we can draw useful insights from the model generalization perspective. Specifically, Eq. 6 can be rephrased as:

$$\max_{x^{adv}} \min_{v \in S_\rho(w)} J(x^{adv}, y; w + v) = \max_{x^{adv}} J(x^{adv}, y; w) - [J(x^{adv}, y; w) - \min_{v \in S_\rho(w)} J(x^{adv}, y; w + v)].$$

Note that the first term $J(x^{adv}, y; w)$ simply corresponds to the adversarial attack objective in (Eq. 1). The second term, which similarly appears in Foret et al. (2020), captures the sharpness of the example loss $J(x^{adv}, y; w)$ on the parameter space by measuring how quickly the loss can be decreased by moving from w to a nearby perturbed parameter value $w+v$. In Foret et al. (2020), it is shown that the sharpness of the loss is highly correlated with the generalization performance of the model *i.e.*, the parameters at flatter loss landscape tend to generalize better to test data. In contrast to Foret et al. (2020) that optimizes the sharpness over the training data by maximization, we optimize it over the adversarial examples by minimization to improve the transferability of *adversarial examples*, not the model parameters. Intuitively, without the sharpness term, naive optimization of Eq 1 will produce an adversarial attack positioned on a sharp local maximum with poor adversarial transferability since even a small weight perturbation will neutralize the attack. Our objective can be viewed as minimizing the sharpness of the loss landscape to reach an adversarial attack at a flat loss landscape that improves the transferability across the target models with different parameters. Indeed, the experimental result in Figure 1 shows the negative correlation between the sharpness of the loss and transferability, *i.e.*, more transferable adversarial examples are placed in the flatter landscape, which verifies our intuition.

2.3 Optimization strategy

We explain the optimization strategy of WAA. We use bi-level optimization that alternates between inner- and outer-loop optimization similar to weight perturbation methods (Wu et al., 2020b). For

Algorithm 1 WAA-MI-FGSM

Input: A target image x, label y, loss function J parameterized by w, maximum image perturbation ϵ, step size α, momentum factor μ, number of outer iterations T_{out}, relative maximum weight perturbation size ρ, number of inner iterations T_{in}, inner step size β.
Output: adversarial image x^{adv}
$x^{adv} \leftarrow x$, $v \leftarrow 0$, $g_0 \leftarrow 0$,
for $t = 1, \ldots, T_{\text{out}}$ **do**
  for $i = 1, \ldots, T_{\text{in}}$ **do**
    $v \leftarrow \Pi_\rho\big(v - \beta \frac{\nabla_v J(x^{adv}, y; w+v)}{\|\nabla_v J(x^{adv}, y; w+v)\|_2} \|w\|_2\big)$
  end for
  $\hat{g}_t \leftarrow \nabla_{x^{adv}} J(x^{adv}, y; w+v)$
  $g_t \leftarrow \mu \cdot g_{t-1} + \frac{\hat{g}_t}{\|\hat{g}_t\|_1}$
  $x^{adv} \leftarrow \text{Clip}(x^{adv} + \alpha \cdot \text{sign}(g_t), x - \epsilon, x + \epsilon)$
end for

the inner minimization, we use normalized gradient descent (Cortés, 2006) that suits constrained optimization on our layer-wise bounds:

$$v \leftarrow \Pi_\rho \left(v - \beta \frac{\nabla_v J(x^{adv}, y; w+v)}{\|\nabla_v J(x^{adv}, y; w+v)\|_2} \|w\|_2 \right), \tag{7}$$

where Π_ρ is a projection operation to satisfy the constraint of Eq. 4, defined as:

$$\Pi_\rho(v) = \begin{cases} \rho \frac{\|w\|_2}{\|v\|_2} v & \text{if } \|v\|_2 > \|w\|_2 \\ v & \text{otherwise} \end{cases}. \tag{8}$$

For the outer maximization, the gradient includes a second-order term:

$$\begin{aligned} \nabla_{x^{adv}} \min_{v \in S(w)} J(x^{adv}, y; w+v) &\approx \nabla_{x^{adv}} J(x^{adv}, y; w+v^*) \\ &= \nabla_{x^{adv}} J(x^{adv}, y; w)\big|_{w+v^*} + \frac{dv^*}{dx^{adv}} \nabla_w J(x^{adv}, y; w)\big|_{w+v^*}, \end{aligned} \tag{9}$$

where v^* is the weight perturbation given by inner minimization. To reduce computation, we use a gradient approximation from Foret et al. (2020) as follows:

$$\nabla_{x^{adv}} \min_{v \in S(w)} J(x^{adv}, y; w+v) \approx \nabla_{x^{adv}} J(x^{adv}, y; w)\big|_{w+v^*}. \tag{10}$$

We observe that dropping the second-order gradients does not degrade the performance. Our method can integrate into existing black-box methods by simply adding an inner loop minimization. We describe our method integrated with MI-FGSM in Alg 1.

3 Related Works

Model manipulation approaches Several works have explored source model manipulation methods. Li et al. (2020b) proposes to adjust the magnitude of skip-connection or dropout, enabling the ensemble effect to generate more transferable attacks. Naseer et al. (2021) suggest self-ensembling that exploits the output class-tokens of different intermediate vision transformer Dosovitskiy et al. (2021); Touvron et al. (2021) blocks. Wu et al. (2020a) further improves Li et al. (2020b) by changing only the backward computation of skip-connection while maintaining the forward. Guo et al. (2020) show that skipping nonlinear components during backpropagation can also improve black-box attacks. While these methods provide extra transferability in their settings, they rely on the architectural characteristics of model networks, i.e., class-token, limiting their applications. In contrast, our method does not lean on architectural properties and functions in a model-agnostic way.

Min-max optimization approaches A few adversarial attack methods employ minimax formulation. Bose et al. (2020) introduces a framework for crafting adversarial examples to hypothesis

classes by a min-max game between a generator of attacks and a classifier when training both from scratch. Wang et al. (2021a) suggest exploiting an affine combination of multiple source models to generate attacks in a min-max formulation. However, their experimental results are limited to small-scaled datasets, e.g., CIFAR-10. In our method, we introduce a worst-case weight perturbation as an inner-loop minimization and verify the effectiveness in more challenging ImageNet evaluation.

Adversarial training One of the best defenses against adversarial attacks to date is adversarial training (Madry et al., 2017; Zhang et al., 2019; Carmon et al., 2019; Wang et al., 2019), in which the training set is augmented by adversarially attacked images of the original training data. However, some works allow the adversary to directly perturb the weight parameters to obtain more robust model. Sun et al. (2021) demonstrates that a carefully designed weight corruption is sufficient to cause misclassification, and training the model parameters to be robust against such corruption can further enhance adversarial robustness. Another line of similar work is Adversarial Weight Perturbations (AWP) (Wu et al., 2020b). Similar to Sharpness-Aware Minimization (SAM) (Foret et al., 2020) relating flat loss landscape with improved generalization to unseen natural images, AWP draws connection between flat loss landscape and adversarial robustness generalization by performing adversarial training under weight perturbations that aim to maximize the loss function. Our method can be interpreted as applying AWP in the adversarial attack scheme, where we optimize adversarial examples under the presence of loss-minimizing weight perturbations.

4 Experiments

4.1 Experimental Setup

Dataset Following the experiment protocols provided by Lin et al. (2020)[3], we conduct experiments on 1000 images randomly chosen from each category of the ILSVRC 2012 validation set (Russakovsky et al., 2015). The RGB values of all images are scaled to range $[0, 1]$. All models that we use in our experiments correctly classifies the images with at least 99.8% accuracy. When simulating the targeted attack, we follow the protocols in the ImageNet-compatible dataset[4] and assign a fixed random target label on each image for evaluation.

Models Following Lin et al. (2020), we consider four classifiers as both the source and the target model in different architectures, namely Inception-v3 (Inc-v3) (Szegedy et al., 2016), Inception-v4 (Inc-v4) (Szegedy et al., 2017), Inception-ResNet-v2 (IncRes-v2) (Szegedy et al., 2017), and ResNet-v2-101 (RN-v2) (He et al., 2016). We also consider four adversarially trained models as extra target models, which are Adv-Inc-v3 (Kurakin et al., 2017), Ens3-Inc-v3, Ens4-Inc-v4, and Ens3-IncRes-v2 (Tramèr et al., 2018). Additionally, we also include advanced defense models for evaluation: FD (Liu et al., 2019), JPEG (Guo et al., 2017), Bit-Red (Xu et al., 2017), NRP (Naseer et al., 2020), and R&P (Xie et al., 2017).

Baselines We compare our method with various black-box attack methods, many of which also address improving the transferability of the attack. As optimization-based methods, we consider three popular baselines: I-FGSM (I) (Kurakin et al., 2018), MI-FGSM (MI) (Dong et al., 2018), and NI-FGSM (NI) (Lin et al., 2020). As approaches based on data augmentation, we consider DI (Xie et al., 2019), TI (Dong et al., 2019), SI (Lin et al., 2020), and CT as their combination (DI, TI, and SI). Since these two classes of baselines address different aspects of black-box transferability, we also consider their combinations as strong baselines in our experiment.

Hyperparameters Following the settings of (Dong et al., 2018; Lin et al., 2020), we set the maximum perturbation bound of the adversarial attack as $\epsilon = 16/255$, number of iterations $T_{out} = 10$, and step size $\alpha = 1.6/255$. In MI-FGSM and NI-FGSM, the decay factor is set to $\mu = 1.0$. In DI, the transformation probability is $p = 0.5$. In TI, we use Gaussian kernel of size 7×7. In SI, we use $m = 5$ scale copies of the input image. For our method, we set the relative perturba-

[3] https://github.com/JHL-HUST/SI-NI-FGSM

[4] https://github.com/cleverhans-lab/cleverhans/tree/master/cleverhans_v3.1.0/examples/nips17_adversarial_competition

Table 1: **Performance of untargeted transfer attacks.** We evaluate using three naturally trained models that were not used to generate the attack and report the success rate (%).

Method	source: inc-v3			source: inc-v4		
	inc-v4	incres-v2	rn-v2	inc-v3	incres-v2	rn-v2
I / +WAA	20.4 / **25.3**	17.7 / **22.1**	14.9 / **18.4**	30.7 / **37.5**	20.7 / **25.2**	18.6 / **22.5**
MI / +WAA	45.4 / **48.5**	42.0 / **46.3**	34.5 / **39.2**	56.2 / **62.1**	46.6 / **50.9**	42.5 / **46.2**
NI / +WAA	52.8 / **54.8**	49.2 / **51.7**	41.6 / **42.4**	64.2 / **66.4**	51.5 / **55.6**	45.4 / **46.9**
MI-DI / +WAA	64.7 / **68.4**	60.9 / **65.9**	54.6 / **58.6**	73.2 / **74.0**	64.3 / **67.1**	55.8 / **58.5**
MI-TI / +WAA	48.2 / **53.9**	43.7 / **48.4**	39.4 / **43.0**	58.2 / **63.5**	46.5 / **52.7**	43.4 / **46.0**
MI-SI / +WAA	70.0 / **76.0**	66.8 / **74.1**	62.0 / **66.8**	80.8 / **85.6**	74.3 / **80.9**	68.7 / **73.5**
MI-CT / +WAA	85.8 / **88.5**	81.8 / **85.7**	77.9 / **81.6**	87.4 / **88.5**	84.3 / **86.5**	78.1 / **81.2**
NI-CT / +WAA	85.0 / **88.1**	82.1 / **83.0**	76.9 / **78.4**	88.2 / **89.8**	83.8 / **86.1**	77.5 / **79.5**
Method	source: incres-v2			source: rn-v2		
	inc-v3	inc-v4	rn-v2	inc-v3	inc-v4	incres-v2
I / +WAA	32.8 / **38.1**	24.3 / **30.0**	20.8 / **22.4**	32.2 / **35.9**	26.1 / **29.6**	22.0 / **26.4**
MI / +WAA	59.3 / **66.7**	50.0 / **57.9**	44.9 / **48.7**	56.9 / **62.6**	52.2 / **56.9**	48.8 / **54.7**
NI / +WAA	62.1 / **64.2**	55.0 / **55.8**	45.5 / **45.7**	64.4 / **67.1**	58.3 / **61.0**	57.1 / **59.1**
MI-DI / +WAA	69.9 / **70.6**	64.2 / **65.4**	**58.7** / **58.7**	32.2 / **35.9**	26.1 / **29.6**	22.0 / **26.4**
MI-TI / +WAA	63.3 / **67.0**	54.3 / **62.1**	50.8 / **53.4**	74.9 / **79.4**	69.8 / **74.0**	70.4 / **73.9**
MI-SI / +WAA	84.0 / **88.2**	80.2 / **84.3**	76.0 / **79.7**	59.4 / **63.8**	53.6 / **57.9**	53.1 / **58.1**
MI-CT / +WAA	88.3 / **90.1**	86.3 / **88.3**	83.2 / **85.5**	73.1 / **79.3**	69.9 / **74.3**	68.8 / **73.2**
NI-CT / +WAA	90.3 / **91.6**	87.3 / **89.3**	82.8 / **84.3**	87.3 / **89.4**	83.5 / **85.1**	84.9 / **86.2**

Table 2: **Performance of untargeted transfer attacks on adversially trained models.** We report the success rate (%) for attacks generated using CT as input transformation.

Source	Method	ens3-inc-v3	ens4-inc-v3	ens3-incres-v2	adv-inc-v3
inc-v3	MI-CT / +WAA	67.1 / **71.6**	64.1 / **69.1**	47.6 / **52.1**	65.4 / **70.5**
	NI-CT / +WAA	61.3 / **63.7**	56.5 / **60.1**	41.7 / **43.8**	61.9 / **62.7**
inc-v4	MI-CT / +WAA	71.9 / **74.0**	68.4 / **71.9**	57.9 / **61.5**	66.6 / **70.6**
	NI-CT / +WAA	66.6 / **68.9**	62.8 / **66.2**	50.6 / **53.3**	63.0 / **66.0**
incres-v2	MI-CT / +WAA	77.9 / **80.5**	74.9 / **77.6**	72.1 / **75.0**	75.3 / **78.8**
	NI-CT / +WAA	74.2 / **76.4**	67.9 / **70.7**	64.5 / **66.5**	72.2 / **73.9**
rn-v2	MI-CT / +WAA	77.2 / **79.3**	72.5 / **75.1**	63.0 / **66.6**	73.3 / **76.1**
	NI-CT / +WAA	72.8 / **74.2**	68.0 / **69.5**	57.7 / **59.5**	71.4 / **72.4**

tion size to $\rho = 5 \times 10^{-4}$ for all models, number of inner iterations $T_{in} = 1$, and inner step size $\beta = \rho/(T_{out} \cdot T_{in})$ unless indicated otherwise.

4.2 Main results

Untargeted attacks Following (Lin et al., 2020), we first evaluate the performance of WAA on four source models, by fusing with seven different combinations of optimization algorithms and input transformation methods: I, MI, NI, MI-DI, MI-TI, MI-SI, MI-CT, and NI-CT. Table 1 shows the attack performance when the generated adversarial examples are transferred to different target classifiers.

When combined with various optimization methods, I, MI, and NI, we observe that our method improves the success rates of the transfer attack consistently over all methods, each of which by 4.4%, 5.1%, and 2.0% on average, respectively. Furthermore, when combining MI with various input transformation methods to create stronger baselines, MI-DI, MI-TI, and MI-SI, our method consistently improves the baselines by 2.8%, 4.7%, and 5.1%, respectively. Finally, even when we create the strongest baselines, MI-CT and NI-CT, by combining all input transformation methods with an optimization algorithm, the average improvements gained by WAA are a considerable 2.3% and 1.8%. It shows that our method provides complementary gains to improve the transferability of the black-box attacks over the existing algorithms and their combinations.

Table 3: **Performance of untargeted transfer attacks on advanced defense methods.** We report the attack success rate (%) of adversarial examples after they pass through defense mechanisms.

Method	Source	FD	JPEG	Bit-Red	NRP	R&P
MI-CT / +WAA	inc-v3	72.0 / **74.8**	76.4 / **80.5**	46.7 / **49.8**	41.5 / **45.0**	68.7 / **73.0**
	inc-v4	71.5 / **74.8**	77.5 / **79.9**	50.4 / **54.3**	47.7 / **50.3**	72.3 / **74.9**
	incres-v2	78.7 / **81.1**	82.6 / **85.3**	60.6 / **63.4**	57.2 / **60.1**	78.8 / **80.7**
	rn-v2	77.8 / **80.1**	81.9 / **84.7**	57.5 / **59.0**	54.8 / **57.3**	78.2 / **80.5**

Table 4: **Performance of targeted transfer attacks.** We evaluate against three naturally trained models that were not used to generate the attack and report the success rate (%).

Methods	Iterations	source: inc-v3			source: inc-v4		
		inc-v4	incres-v2	rn-v2	inc-v3	incres-v2	rn-v2
MI-CT / +WAA	100	16.6 / **21.7**	14.8 / **19.7**	8.4 / **10.1**	9.8 / **15.6**	11.0 / **16.7**	5.3 / **8.2**
	300	19.3 / **28.0**	17.6 / **25.8**	10.4 / **13.2**	10.8 / **19.3**	12.5 / **23.5**	6.3 / **10.3**
Attack	iterations	source: incres-v2			source: rn-v2		
		inc-v3	inc-v4	rn-v2	inc-v3	inc-v4	incres-v2
MI-CT / +WAA	100	10.4 / **20.1**	13.1 / **23.4**	10.0 / **17.0**	26.4 / **31.5**	23.8 / **29.7**	30.2 / **36.8**
	300	12.2 / **22.1**	15.2 / **25.7**	10.4 / **18.4**	29.6 / **36.1**	28.4 / **35.5**	35.7 / **42.8**

Since MI-CT and NI-CT are the strongest baselines with the highest transfer success rate, we additionally evaluate those attacks (with WAA) on four adversarially trained models. As shown in Table 2, WAA brings 3.4% and 2.2% extra success rate over the baselines MI-CT and NI-CT respectively. Altogether, the results on untargeted attcks demonstrate that WAA can generate more transferable adversarial examples from the same source model compared to baselines of varying strength.

Untargeted attacks on advanced defense methods We also evaluate a stong baseline MI-CT and our method against several advanced adversarial defense methods. To this end, we feed the adversarial examples through each defense method and evaluate the output images on the Ens3-Inc-v3 classifier. Table 3 summarizes the result. We can observe that MI-CT-WAA achieves higher attack success rate under all scenarios, and outperforms the baseline MI-CT with a clear margin of 2.7% averaged across all models and defenses.

Targeted attacks We also evaluate our method on a more challenging task of targeted attack, whose objective is to produce the pre-defined label from the targeted attack. Following Zhao et al. (2021), we employ the logit loss to generate attacks, and increase the number of steps to $T = 100$ or $T = 300$ while fixing other hyperparameters. Note that the inner minimization objective in line 6 of Alg 1 is also replaced with the logit loss to demonstrate that WAA works with loss functions other than standard cross entropy as well.

Table 4 summarizes the results. After 100 iterations, WAA improves the attack success rate against target models by 5.9%, and shows even larger average improvement of 7.7% after 300 iterations. Hence we conclude WAA shows clear improvements in targeted attack success rates across all source models.

4.3 Analysis

Comparison with model manipulation baselines We compare our method with other alternatives of model manipulation. (1) Random weight augmentation derived from Eq. 5: Instead of adding the worst-case perturbation in line 6 of Alg 1, we replace this line by adding a random Gaussian noise of the same norm. (2) Dropout erosion proposed by Li et al. (2020b): They apply random dropout to intermediate feature layers, which can also be viewed as a virtual ensemble of models created in a model-agnostic manner. The dropout probability is set to 0.006 for Inception-v3 network following the official code repository[5].

[5] https://github.com/LiYingwei/ghost-network

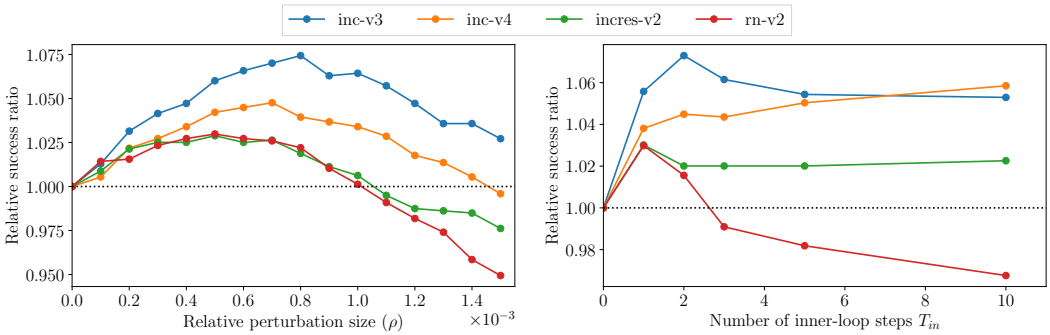

Figure 2: **Ablation study on hyperparameters** (Left) ρ and (Right) T_{in}.

Table 5: **Comparison with baseline model manipulation methods.** We report the success rate (%) of the untargeted transfer attacks created from Inception-v3 model using different model manipulation methods. We denote **+R**: random augmentation by Eq. 5, **+D**: Dropout erosion (Li et al., 2020b), and **+W**: WAA (ours).

Methods	inc-v4	incres-v2	rn-v2
MI / +R / +D / +W	45.4 / 44.0 / 47.8 / **48.5**	42.0 / 42.1 / 45.7 / **46.3**	34.5 / 36.2 / 38.6 / **39.2**
MI-CT / +R / +D / +W	85.8 / 85.6 / 60.5 / **88.5**	81.8 / 82.3 / 57.7 / **85.7**	77.9 / 77.2 / 52.0 / **81.6**

The results are shown in Table 5. We first observe that random weight augmentation occasionally provides improvement over the baselines MI and MI-CT, yet the improvements are marginal in general as discussed in Section 2.2. On the other hand, Dropout erosion provides consistent improvement over MI, while its performance drops significantly when combined with input transformation (MI-CT).

Ablation on relative maximum perturbation size ρ While we vary the perturbation size ρ, the inner step size β must change accordingly as $\beta = \rho/(T_{out} \cdot T_{in})$ to allow the optimization process reach the perturbation bound. With that in mind, the results of MI-CT-WAA with different values of ρ against 4 source models are illustrated in Figure 2 (Left). While even small values of ρ show better transferability, the range $4 \times 10^{-4} \leq \rho \leq 7 \times 10^{-4}$ shows consistent improvement across all models, then the performance starts declining as the bound becomes too wide and no longer qualifies as a set of loss-preserving weight perturbations. Thus we adopt $\rho = 5 \times 10^{-4}$ for our experiments.

Ablation on the number of inner steps T_{in} We vary the number of inner steps in the range $T_{in} \in \{1, 2, 3, 5, 10\}$ for each source model, and the results are in Figure 2 (Right). Interestingly, varying T_{in} delivers mixed effects depending on the source model. Increasing the number of inner steps imply that earlier steps of the outer optimization have to deal with larger magnitudes of weight perturbation. Since the nature of I-FGSM produces input perturbations of smaller norm in the earlier stages, such exposure to extreme worst-case weight perturbation might overwhelm the adversarial effect of input perturbation.

5 Conclusion

In this work, we propose Worst-case Aware Attack (WAA), an intriguing direction of model augmentation that improves the transferability of generated adversarial examples. WAA applies per-example worst-case weight perturbations on the source model to obtain weight-augmented models, which provide a virtual ensemble of models to mitigate overfitting on a specific model. While existing model manipulation approaches rely on parts that some model architectures have in common to avoid overfitting, the weight perturbation operation of WAA is model-agnostic and applies to a broader range of networks without any adaptation. Extensive experiments on a subset of ImageNet demonstrate that WAA can be combined with baseline attacks to improve the transferability of adversarial attacks further. Finally, we draw some similarities between weight loss landscape and adversarial transferability, which we leave as an interesting future work direction for a more thorough investigation.

REPRODUCIBILITY STATEMENT

Our method is simple and easy to implement. To support reproducibility, we include a pseudo-code description of our method in Sec. 2.3 while also including the core part of implementation with TensorFlow in the Appendix. For the experimental results in Sec. 4, we provide all the detailed experimental setups in Sec 4.1, including the references to our baselines. For the results in Fig 1, we add more experimental details in the Appendix for reproducibility. We will release our code in public as soon as the reviewing process ends.

ETHICS STATEMENT

Our work focuses on improving black-box adversarial transferability, which may assist groups with a malicious intention on disrupting neural network models deployed in real-world environments. However, a black-box adversary must have access to a compatible training dataset or at least a surrogate model to maximize the chances of adversarial transferability, which restricts the use cases of our method. We believe our work also highlights the need for more advanced defense mechanisms against such attacks, which our work will support such research in the long run.

REFERENCES

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pp. 274–283. PMLR, 2018.

Joey Bose, Gauthier Gidel, Hugo Berard, Andre Cianflone, Pascal Vincent, Simon Lacoste-Julien, and Will Hamilton. Adversarial example games. *Advances in neural information processing systems*, 33:8921–8934, 2020.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. *Advances in Neural Information Processing Systems*, 32, 2019.

Jorge Cortés. Finite-time convergent gradient flows with applications to network consensus. *Automatica*, 42(11):1993–2000, 2006.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International conference on machine learning*, pp. 2206–2216. PMLR, 2020.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4312–4321, 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 *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *arXiv preprint arXiv:1711.00117*, 2017.

Yiwen Guo, Qizhang Li, and Hao Chen. Backpropagating linearly improves transferability of adversarial examples. *Advances in Neural Information Processing Systems*, 33:85–95, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016.

Warren He, Bo Li, and Dawn Song. Decision boundary analysis of adversarial examples. In *International Conference on Learning Representations*, 2018.

Qian Huang, Isay Katsman, Horace He, Zeqi Gu, Serge Belongie, and Ser-Nam Lim. Enhancing adversarial example transferability with an intermediate level attack. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4733–4742, 2019.

Yi Huang and Adams Wai-Kin Kong. Transferable adversarial attack based on integrated gradients. *arXiv preprint arXiv:2205.13152*, 2022.

Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva. On relating explanations and adversarial examples. *Advances in neural information processing systems*, 32, 2019.

Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*, 2016.

Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=BJm4T4Kgx.

Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pp. 99–112. Chapman and Hall/CRC, 2018.

Qizhang Li, Yiwen Guo, and Hao Chen. Yet another intermediate-level attack. In *European Conference on Computer Vision*, pp. 241–257. Springer, 2020a.

Yingwei Li, Song Bai, Yuyin Zhou, Cihang Xie, Zhishuai Zhang, and Alan Yuille. Learning transferable adversarial examples via ghost networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 11458–11465, 2020b.

Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJlHwkBYDH.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations*, 2017. URL https://openreview.net/forum?id=Sys6GJqxl.

Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 860–868. IEEE, 2019.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 262–271, 2020.

Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Fahad Khan, and Fatih Porikli. On improving adversarial transferability of vision transformers. In *International Conference on Learning Representations*, 2021.

Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o (1/kˆ 2). In *Doklady an ussr*, volume 269, pp. 543–547, 1983.

Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Alexey Kurakin, Cihang Xie, Yash Sharma, Tom Brown, Aurko Roy, Alexander Matyasko, Vahid Behzadan, Karen Hambardzumyan, Zhishuai Zhang, Yi-Lin Juang, Zhi Li, Ryan Sheatsley, Abhibhav Garg, Jonathan Uesato, Willi Gierke, Yinpeng Dong, David Berthelot, Paul Hendricks, Jonas Rauber, and Rujun Long. Technical report on the cleverhans v2.1.0 adversarial examples library. *arXiv preprint arXiv:1610.00768*, 2018.

Zeyu Qin, Yanbo Fan, Yi Liu, Yong Zhang, Jue Wang, and Baoyuan Wu. Boosting the transferability of adversarial attacks with reverse adversarial perturbation, 2022. URL https://openreview.net/forum?id=i7FNvHnPvPc.

Leslie Rice, Eric Wong, and Zico Kolter. Overfitting in adversarially robust deep learning. In *International Conference on Machine Learning*, pp. 8093–8104. PMLR, 2020.

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. *International journal of computer vision*, 115(3):211–252, 2015.

Ali Shafahi, Mahyar Najibi, Mohammad Amin Ghiasi, Zheng Xu, John Dickerson, Christoph Studer, Larry S Davis, Gavin Taylor, and Tom Goldstein. Adversarial training for free! *Advances in Neural Information Processing Systems*, 32, 2019.

Xu Sun, Zhiyuan Zhang, Xuancheng Ren, Ruixuan Luo, and Liangyou Li. Exploring the vulnerability of deep neural networks: A study of parameter corruption. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11648–11656, 2021.

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.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Herve Jegou. Training data-efficient image transformers & distillation through attention. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 10347–10357. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/touvron21a.html.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rkZvSe-RZ.

Jingkang Wang, Tianyun Zhang, Sijia Liu, Pin-Yu Chen, Jiacen Xu, Makan Fardad, and Bo Li. Adversarial attack generation empowered by min-max optimization. *Advances in Neural Information Processing Systems*, 34:16020–16033, 2021a.

Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1924–1933, 2021.

Xiaosen Wang, Xuanran He, Jingdong Wang, and Kun He. Admix: Enhancing the transferability of adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16158–16167, 2021b.

Xiaosen Wang, Jiadong Lin, Han Hu, Jingdong Wang, and Kun He. Boosting adversarial transferability through enhanced momentum. *arXiv preprint arXiv:2103.10609*, 2021c.

Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2019.

Zhibo Wang, Hengchang Guo, Zhifei Zhang, Wenxin Liu, Zhan Qin, and Kui Ren. Feature importance-aware transferable adversarial attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 7639–7648, 2021d.

Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. In *International Conference on Learning Representations*, 2020a. URL https://openreview.net/forum?id=BJlRs34Fvr.

Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Advances in Neural Information Processing Systems*, 33:2958–2969, 2020b.

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

Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *arXiv preprint arXiv:1711.01991*, 2017.

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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2730–2739, 2019.

Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *arXiv preprint arXiv:1704.01155*, 2017.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pp. 7472–7482. PMLR, 2019.

Zhengyu Zhao, Zhuoran Liu, and Martha Larson. On success and simplicity: A second look at transferable targeted attacks. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan (eds.), *Advances in Neural Information Processing Systems*, volume 34, pp. 6115–6128. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper/2021/file/30d454f09b771b9f65e3eaf6e00fa7bd-Paper.pdf.

A DETAILED DESCRIPTION OF TOY EXPERIMENTS

All experiments are performed on the same dataset with the main experiments.

Figure 1 (Left) We visualize the average cross-entropy loss using the following formula:

$$\mathcal{L}(x_j, \rho) = \frac{1}{\|\mathcal{D}\|} \sum_{x_i \in \mathcal{D}} J(x_i, y_i, w - \rho v_j), \tag{11}$$

$$v_j = \frac{\nabla_v J(x_j, y_j; w)}{\|\nabla_v J(x_j, y_j; w)\|_2} \|w\|_2. \tag{12}$$

Here, we sample a fixed set of 100 random images from the dataset to create a validation subset $\mathcal{D}$, and we average $\mathcal{L}(x_j, \rho)$ over a fixed set of 20 random images distinct from the validation subset. For random weight perturbations, v_j is replaced by a random Gaussian noise of the same L_2 norm.

Figure 1 (Right) We visualize the relationship between sharpness and transferability using the Inception-v3 model. Given a set of attacked images $\mathcal{D} = x_i^{adv}$, we define sharpness as:

$$S(\mathcal{D}) = \frac{1}{\|\mathcal{D}\|} \sum_{x_i^{adv} \in \mathcal{D}} \left[J(x_i^{adv}, y_i; w) - \min_{v \in S_\rho(w)} J(x_i^{adv}, y_i; w + v) \right] \tag{13}$$

Under this definition, a larger sharpness implies the set of attacks can be easily weakened by adding small weight perturbations. In Figure 1 (Right), we demonstrate that MI attack augmented with WAA shows lower sharpness (i.e. is more robust weight perturbations) while the average success rate also increases.

B IMPLEMENTATION

For reproducibility, we include the core implementation of our method WAA, below:

```
if FLAGS.beta > 0:
        for j in range(FLAGS.inner):
            weights_nat = tf.get_collection(tf.GraphKeys.
    TRAINABLE_VARIABLES, scope=f'Nat/{scope_name}')

            # Calculate gradient of current adversarial example
            with tf.control_dependencies(deps):
                if FLAGS.si and FLAGS.mode == 'full':
                    x_scaled_list = []
                    for scale in [1.0, 2.0, 4.0, 8.0, 16.0]:
                        x_scaled_list.append(x_nes / scale)
                    x_scaled_list = tf.concat(x_scaled_list, 0)
                    one_hot_batched = tf.concat([one_hot, one_hot,
    one_hot, one_hot, one_hot], 0)
                    inner_adv_logit, _ = model_forward(x_scaled_list, '
    Adv')
                    inner_cross_entropy = tf.losses.softmax_cross_entropy
    (one_hot_batched, inner_adv_logit)
                else:
                    if FLAGS.mode == 'simple':
                        inner_adv_logit, _ = model_forward(x_nes, 'Adv')
                    else:  # 'simplest'
                        inner_adv_logit, _ = model_forward_no_diversity(
    x_nes, 'Adv')
                    inner_cross_entropy = tf.losses.softmax_cross_entropy
    (one_hot, inner_adv_logit)
                    adv_loss_list = tf.tensor_scatter_update(
    adv_loss_list, [[i, j]], [inner_cross_entropy])

            # Collect reference to weights
            deps = [inner_cross_entropy, adv_loss_list]
```

```
            weights_adv = tf.get_collection(tf.GraphKeys.
    TRAINABLE_VARIABLES, scope=f'Adv/{scope_name}')

            # Add weight perturbation in the loss-decreasing direction
            with tf.control_dependencies(deps):
                grad_weights = tf.gradients(inner_cross_entropy,
    weights_adv)
                for grad_weight, weight_adv, weight_nat in zip(
    grad_weights, weights_adv, weights_nat):
                    if 'BatchNorm' in weight_adv.name or 'Aux' in
    weight_adv.name:
                        continue
                    grad_norm = tf.norm(grad_weight, FLAGS.lp)
                    weight_norm = tf.norm(weight_nat) if FLAGS.norm == '
    rel' else 1
                    scale = - FLAGS.beta * weight_norm / (grad_norm + 1e
    -12)
                    new_pert = (weight_adv - weight_nat) + grad_weight *
    scale
                    new_pert = tf.clip_by_norm(new_pert, FLAGS.beta *
    weight_norm * FLAGS.multiplier)
                    deps.append(weight_adv.assign(weight_nat + new_pert))
```

C ADDITIONAL EXPERIMENTAL RESULTS

Targeted attack against adversarially trained models Extending Table 4, we evaluate targeted attacks generated using MI-CT and MI-CT-WAA on adversarially trained models as well. Results provided in Table 6 demonstrate that WAA still delivers improvement on the extremely challenging setting of generating targeted attacks that are transferable to adversarially trained models.

Table 6: Performance of targeted transfer attacks on adversarially trained models.

Source	Attack	ens3-inc-v3	ens4-inc-v3	ens3-incres-v2	adv-inc-v3
inc-v3	MI-CT / +AWP (100step)	0.4 / 0.5	0.4 / 0.5	0.3 / 0.1	0.7 / 0.7
	MI-CT / +AWP (300step)	0.5 / 0.9	0.4 / 0.7	0.2 / 0.2	0.6 / 0.7
inc-v4	MI-CT / +AWP (100step)	1 / 1.3	0.5 / 0.7	0.3 / 0.4	0.6 / 0.8
	MI-CT / +AWP (300step)	0.6 / 1.1	0.4 / 0.5	0.1 / 0.3	0.7 / 0.9
incres-v2	MI-CT / +AWP (100step)	2 / 2.8	0.9 / 1	0.4 / 0.6	1.1 / 1.3
	MI-CT / +AWP (300step)	1.9 / 3	1 / 1	0.4 / 0.5	1 / 1.4
rn-v2	MI-CT / +AWP (100step)	4.6 / 5.1	2.3 / 2.8	0.6 / 1	3.2 / 4.1
	MI-CT / +AWP (300step)	3.5 / 6.5	1.5 / 3	0.8 / 1.4	2.7 / 4.4

Experimental results *w.r.t.* different values of perturbation size ρ In the main submission, Table 1 provides the evaluation under $\rho = 5 \times 10^{-4}$. Here, we further report the performance of WAA under different values of ρ in Tables 7 and 8, to show that our method outperforms baselines at wide range of hyperparameters.

Table 7: Additional results of $\rho = 4 \times 10^{-4}$

Attack	source: inc-v3			source: inc-v4		
	inc-v4	incres-v2	rn-v2	inc-v3	incres-v2	rn-v2
MI / +AWP	45.4 / 48.2	42 / 46.1	34.5 / 39.8	56.2 / 60.8	46.6 / 49.7	42.5 / 44.4
NI / +AWP	52.8 / 55.2	49.2 / 53.3	41.6 / 43.5	64.2 / 64.8	51.5 / 53.6	45.4 / 46.4
MI-CT / +AWP	85.8 / 88.3	81.8 / 85.3	77.9 / 80.9	87.4 / 88.9	84.3 / 86.2	78.1 / 81.3
NI-CT / +AWP	85 / 86.9	82.1 / 83.1	76.9 / 78.6	88.2 / 89.4	83.8 / 85.3	77.5 / 79.4
Attack	source: incres-v2			source: rn-v2		
	inc-v3	inc-v4	rn-v2	inc-v3	inc-v4	rn-v2
MI / +AWP	59.3 / 64.1	50 / 55.7	44.9 / 48.5	56.9 / 61.1	52.2 / 55.3	48.8 / 54.1
NI / +AWP	62.1 / 63.8	55 / 55.1	45.5 / 46	64.4 / 67.2	58.3 / 60.2	57.1 / 59.4
MI-CT / +AWP	88.3 / 89.7	86.3 / 88.1	83.2 / 85.1	86.5 / 88.1	82.7 / 84.5	84.7 / 86
NI-CT / +AWP	90.3 / 91.4	87.3 / 88.6	82.8 / 84.3	87.3 / 89.3	83.5 / 84.2	84.9 / 85.7

Table 8: Additional results of $\rho = 6 \times 10^{-4}$

Attack	source: inc-v3			source: inc-v4		
	inc-v4	incres-v2	rn-v2	inc-v3	incres-v2	rn-v2
MI / +AWP	45.4 / 49.6	42 / 47.2	34.5 / 41.1	56.2 / 62.7	46.6 / 52.4	42.5 / 45.8
NI / +AWP	52.8 / 55	49.2 / 51.8	41.6 / 42.9	64.2 / 65.7	51.5 / 54.9	45.4 / 45.7
MI-CT / +AWP	85.8 / 88.8	81.8 / 86.2	77.9 / 82.2	87.4 / 88.9	84.3 / 87	78.1 / 80.9
NI-CT / +AWP	85 / 87.2	82.1 / 83.2	76.9 / 79	88.2 / 89.4	83.8 / 85.9	77.5 / 79.9
Attack	source: incres-v2			source: rn-v2		
	inc-v3	inc-v4	rn-v2	inc-v3	inc-v4	rn-v2
MI / +AWP	59.3 / 65.8	50 / 57.8	44.9 / 49.3	56.9 / 63.5	52.2 / 57.9	48.8 / 54.8
NI / +AWP	62.1 / 65.8	55 / 56.4	45.5 / 47.7	64.4 / 67.8	58.3 / 61.4	57.1 / 59.7
MI-CT / +AWP	88.3 / 89.7	86.3 / 88.3	83.2 / 85.1	86.5 / 88.1	82.7 / 85.4	84.7 / 85.8
NI-CT / +AWP	90.3 / 91.9	87.3 / 88.9	82.8 / 84.9	87.3 / 89.1	83.5 / 85.3	84.9 / 86.6