Training Meta-Surrogate Model for Transferable Adversarial Attack

Yunxiao Qin[1, 2], **Yuanhao Xiong**[3], **Jinfeng Yi**[4], **Cho-Jui Hsieh**[3]

[1]State Key Laboratory of Media Convergence and Communication, Communication University of China, Beijing, China
[2]Neuroscience and Intelligent Media Institute, Communication University of China, Beijing, China
[3]University of California, Los Angeles, USA
[4]JD AI Research, Beijing, China
qinyunxiao@cuc.edu.cn, jinfengyi.ustc@gmail.com, {yhxiong, chohsieh}@cs.ucla.edu.

Abstract

The problem of adversarial attacks to a black-box model when no queries are allowed has posed a great challenge to the community and has been extensively investigated. In this setting, one simple yet effective method is to transfer the obtained adversarial examples from attacking surrogate models to fool the target model. Previous works have studied what kind of attacks to the surrogate model can generate more transferable adversarial examples, but their performances are still limited due to the mismatches between surrogate models and the target model. In this paper, we tackle this problem from a novel angle—instead of using the original surrogate models, can we obtain a **Meta-Surrogate Model** (MSM) such that attacks to this model can be easily transferred to other models? We show that this goal can be mathematically formulated as a bi-level optimization problem and design a differentiable attacker to make training feasible. Given one or a set of surrogate models, our method can thus obtain an MSM such that adversarial examples generated on MSM enjoy eximious transferability. Comprehensive experiments on Cifar-10 and ImageNet demonstrate that by attacking the MSM, we can obtain stronger transferable adversarial examples to deceive black-box models including adversarially trained ones, with much higher success rates than existing methods.

Introduction

The developments of Convolutional Neural Network (CNN) have greatly promoted the advancements in Computer Vision. However, previous works (Ganeshan, BS, and Babu 2019) shown a critical robustness issue that CNN models are vulnerable to human-imperceptible perturbations of input images, also known as adversarial examples (AEs). The design of AEs is useful for revealing the security threats on machine learning systems (Croce and Hein 2020b) and for understanding the representations learned by CNN (Ilyas et al. 2019).

In this paper, we consider the problem of black-box attack, where the target victim model is entirely hidden from the attacker. In this setting, standard white-box attacks (Moosavi-Dezfooli, Fawzi, and Frossard 2016; Carlini and Wagner 2017) or even query-based black-box attacks (Cheng et al. 2020, 2019) cannot be used, and the prevailing way to attack the victim is through transfer attack (Papernot et al. 2017; Wu et al. 2018). In transfer attack (Demontis et al. 2019; Dong et al. 2018; Naseer et al. 2019; Wu et al. 2020b), the attackers commonly generate AEs by attacking one or an ensemble of **surrogate models** and expect the obtained AEs can also successfully fool the victim black-box model.

Although great efforts have been made to improve the transferability of adversarial attacks (Wu et al. 2020a; Wang et al. 2021a; Huang and Kong 2022), the transfer attack-based methods can only achieve poor success rates. This is caused by a fundamental limitation of current approaches—they all leverage the surrogate models trained by standard learning tasks (e.g., classification, object detection), while it is not always the case that attacks fooling such models can be easily transferred, even though the optimization of adversarial examples have been greatly improved. We thus pose the following important question on transfer attack that has not been well studied in the literature: Instead of using standard (naturally trained) models as surrogate, can we find a **Meta-Surrogate Model** (MSM) such that attacks to this model can be easier transferred to other models?

We answer this question in the affirmative by developing a novel black-box attack pipeline called **Meta-Transfer Attack** (MTA). Assume a set of source models (standard surrogate models) are given, instead of directly attacking these source models, our algorithm aims to obtain a "meta-surrogate model (MSM)", which is trained for the goal that attacks to this model can be easier transferred to fool other models, and conduct attacks on the MSM to obtain transferable AEs. We show that this goal can be mathematically formulated as a well-posed (bi-level-like) training objective by unrolling the attacks on the MSM and defining an adversarial loss to supervise the transferability of the resulting AEs. To avoid discrete operations in the white-box attack, we propose a Customized PGD attacker that enables back-propagation through the whole MTA framework.

The proposed MTA differs greatly from existing transfer attacks, especially ensemble-based methods (Dong et al. 2019; Lin et al. 2020). The key difference is: Existing methods commonly generate AEs by directly attacking source models (optimizing AEs on source models), and to improve the transferability of AEs, they commonly propose some ideas (e.g., gradient momentum (Dong et al. 2018), gradient on skip-connections (Wu et al. 2020a)) to **implicitly improve** the optimization of the AEs. Rather than implicitly improves the transferability of AEs, MTA **explicitly optimizes the transferability** by bi-level training (Finn, Abbeel, and Levine 2017; Qin et al. 2020) an MSM and generates AEs by attacking the MSM (optimizing AEs on MSM rather than source models). **The bi-level training of MSM is a closed-loop** of 1) generating AEs by attacking MSM; 2) evaluating transferability of AEs on source models; and 3) improving transferability by optimizing MSM. This closed-loop training is the foremost reason why attacking the trained MSM can produce stronger transferable AEs than existing methods, no matter the number of source models. Through extensive experiments on various models and datasets, we show that the proposed MTA leads to clearly improved transfer attacks, proving the effectiveness of MTA.

The main contributions of our work are the follows. **1)** We propose a novel bi-level training framework MTA to train an MSM to improve transfer attack. To the best of our knowledge, our work is the first attempt to explore a better surrogate model for producing stronger transferable AEs. **2)** We carefully design a Customized PGD to enable back-propagation in MTA, and we analyze the necessity of Customized PGD in Gradient Calculation (page 4) and Appendix. **3)** We compare MTA with state-of-the-art transfer attack methods (*e.g.*, MI (Dong et al. 2018), DI (Xie et al. 2019), TI (Dong et al. 2019), SGM (Wu et al. 2020a), AEG (Bose et al. 2020), IR (Wang et al. 2021a), SI-NI (Lin et al. 2020), FIA (Wang et al. 2021b), DA (Huang et al. 2022)) on Cifar-10 (Krizhevsky, Hinton et al. 2009) and Imagenet (Deng et al. 2009). The comparisons demonstrate the effectiveness of MTA—the AEs generated by attacking MSM significantly outperform previous methods, in attacking both naturally trained and adversarially trained black-box target models.

Background

Adversarial Attacks. Szegedy et al. (2014) reveals the interesting phenomenon that CNN models are vunerable to adversarial attacks. After that, many attacks have been developed (Kaidi et al. 2019; Gao et al. 2020; Wu, Wang, and Yu 2020; Li, Guo, and Chen 2020; Sriramanan et al. 2020; Naseer et al. 2019). Adversarial attacks can be mainly classified into white-box and black-box attacks (Maksym et al. 2020) according to how much information about the target model is exposed to the attacker. White-box attacks (Kurakin, Goodfellow, and Bengio 2018) are often more effective than black-box attacks (Brendel, Rauber, and Bethge 2017) as they can leverage full knowledge of the target model including the model weights and architecture. For example, Fast Gradient Sign Method (FGSM) (Goodfellow, Shlens, and Szegedy 2014) uses 1-step gradient ascent to produce adversarial examples that enlarge the model's loss. Projected gradient descent (PGD) attack can be viewed as a multi-step FGSM attack (Madry et al. 2018). Many other white-box attacks have also been developed by leveraging full information of the target model (Croce and Hein 2020a). In the black-box setting, query-based black-box attacks (Huang and Zhang 2020; Du et al. 2020) assume model information is hidden but attackers can query the model and observe the corresponding hard-label or soft-label predictions. Among them, (Chen et al. 2017; Ilyas et al. 2018) considered soft-label probability predictions and (Chen, Jordan, and Wainwright 2020; Cheng et al. 2018) considered hard-label decision-based predictions. Considering that using a large number of queries to attack an image is impractical, several works try to further reduce the query counts (Li et al. 2020a; Wang et al. 2020).

Transferable Adversarial Attacks. In this paper, we consider the black-box attack scenario when the attacker cannot make any query to the target model (Huang et al. 2019; Huang and Kong 2022). In this case, the common attack method is based on transfer attack—the attacker generates AEs by attacking one or few surrogate models and hopes the AEs can also fool the target model (Liu et al. 2017; Liu, Jiang, and Jiang 2022). Compared with query-based attacks, crafting AEs from the surrogate model consumes less computational resources and is more realistic in practice. Along this direction, subsequent works have attempted to improve the transferability of AEs (Guo, Li, and Chen 2020; Wang and He 2021; Zhou et al. 2018; Li et al. 2020b). For instance, MI boosted the transferability by integrating the momentum term into the iterative process. Other techniques like data augmentations (Xie et al. 2019), exploiting gradients of skip-connection (Wu et al. 2020a), and negative interaction between pixels (Wang et al. 2021a) also contribute to stronger transferable attacks. DA (Huang et al. 2022) utilizes aggregated gradient direction during the attack process to avoid the generated adversarial examples overfitting to white-box surrogate models. MGAA (Yuan et al. 2021) shows that narrowing the direction gap between white-box gradient and black-box gradient improves transferability as well. Although MGAA also uses meta-learning, the proposed MTA differs greatly from MGAA because at each gradient ascent step, MGAA samples multiple source models from a source-model zoo to construct a meta-task to improve the optimization of AE. In addition to using the original surrogate models, AEG (Bose et al. 2020) adversarially trains a robust classifier together with an encoder-decoder-based perturbation generator. After the training, AEG uses the generator to generate transferable AEs. Compared to all the existing works, our method is the first that meta-trains a new meta-surrogate model (MSM) such that attacks on MSM can be easier transferred to other models. This not only differs from all the previous methods that attack standard surrogate models but also differs from the encoder-decoder based method such as AEG.

Methodology

We consider the black-box attack setting where the target model is hidden to the attacker and queries are not allowed. This setting is also known as the transfer attack setting (Dong et al. 2018, 2019) and the attacker 1) cannot access the weight, the architecture, and the gradient of the target model; and 2) cannot query the target model. The attacker can access 1) the dataset used by the target model; and 2) a single or a set of **surrogate models** (also known as **source models**) that may share the dataset with the target model. For example, it is common to assume that the attacker can access one or multiple well-performed (pretrained) image classification models. Existing transferable adversarial attack methods conduct various attacks to these models and hope to get transferable AEs that can fool an unknown target model. Instead of proposing

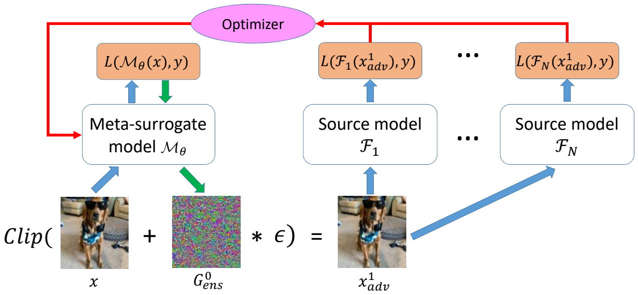
Optimizer
$L(\mathcal{M}_\theta(x), y)$
$L(\mathcal{F}_1(x_{adv}^1), y)$
...
$L(\mathcal{F}_N(x_{adv}^1), y)$
Meta-surrogate model $\mathcal{M}_\theta$
Source model $\mathcal{F}_1$
...
Source model $\mathcal{F}_N$
Clip(
+
$* \epsilon$) =
x
G_{ens}^0
x_{adv}^1

Figure 1: The framework of the proposed MTA when $T = 1$ and $\mathcal{A}(\mathcal{M}_\theta(x)) = x_{adv}^1$. The clean image x is first feed into the MSM $\mathcal{M}_\theta$ and obtain the loss $L(\mathcal{M}_\theta(x), y)$. Next we back-propagate the loss and use Eq 4 to obtain the noise g_{ens}^0. Then, via Eq 5, we obtain the adversarial example x_{adv}^1 which will be feed into the source models $\mathcal{F}_1$, $\mathcal{F}_2$, ..., and $\mathcal{F}_N$. Finally, by maximizing the source models' loss, we can optimize the MSM to learn a particular weight so that the adversarial example x_{adv}^1 attacking it can fool source models.

another attack method on surrogate models, we propose a novel framework MTA to train a **Meta-Surrogate Model (MSM)** with the goal that attacking the MSM can generate stronger transferable AEs than directly attacking the original surrogate models. When evaluating, the transferable AEs are generated by attacking the MSM with standard white-box attack methods (e.g., PGD attack). In the following, we will first review exiting attacks and then show how to form a bi-level optimization objective to train the MSM model.

Reviews of FGSM and PGD

We follow existing works (Xie et al. 2019; Wu et al. 2020a; Wang et al. 2021a) to focus on untargeted attack, where the attack is considered successful as long as the perturbed image is wrongly predicted.

FGSM conducts one-step gradient ascent to generate AEs to enlarge the prediction loss. The formulation is

$$x_{adv} = \text{Clip}\big(x + \epsilon \cdot \text{sign}(\nabla_x L(f(x), y))\big), \quad (1)$$

where x is a clean image and y is the corresponding label; ϵ is the attack step size that determines the maximum L_∞ perturbation of each pixel; f is the victim model that is transparent to the FGSM attacker; Clip is the function that clipping the values of x_{adv} to the legal range (*e.g.*, clipping the RGB AEs to the range of $[0, 255]$); L is usually the cross-entropy loss.

PGD (Kurakin, Goodfellow, and Bengio 2018), also known as I-FGSM, is a multi-step extension of FGSM. The formulation of PGD is

$$x_{adv}^k = \text{Clip}\big(x_{adv}^{k-1} + \frac{\epsilon}{T} \cdot \text{sign}(\nabla_{x_{adv}^{k-1}} L(f(x_{adv}^{k-1}), y))\big). \quad (2)$$

x_{adv}^k is the AEs generated in the k-th gradient ascent step. Note that x_{adv}^0 is the clean image equals to x. Eq 2 will be run for T iterations to obtain x_{adv}^T with perturbation size ϵ.

Meta-Transfer Attack

How to train the MSM where attacks to this model can be easier transferred to other models? We show this can be formulated as a bi-level training objective. Let $\mathcal{A}$ denote an attack algorithm (e.g., FGSM or PGD) and $\mathcal{M}_\theta$ denote the **MSM** parameterized by θ. For a given image x, the AE generated by attacking $\mathcal{M}_\theta$ can be denoted as $\mathcal{A}(\mathcal{M}_\theta, x, y)$. For example, if $\mathcal{A}$ is FGSM, then $\mathcal{A}(\mathcal{M}_\theta, x, y) = x_{adv} = \text{Clip}\big(x + \epsilon \cdot \text{sign}(\nabla_x L(\mathcal{M}_\theta(x), y))\big)$. Since in the attack time we only have access to a set of source models $\mathcal{F}_1, \ldots, \mathcal{F}_N$, we can evaluate the transferability of the adversarial example $\mathcal{A}(\mathcal{M}_\theta, x, y)$ on the source models and optimize the MSM via maximizing the adversarial losses of those N source models, leading to the following training objective:

Algorithm 1: Training of Meta-Transfer Attack

```
input: $N$ source models $\mathcal{F}_1, \ldots, \mathcal{F}_N$, Training set $\mathbb{D}$, batch size $b$, initialized MSM $\mathcal{M}_\theta$.
output: Optimized weight $\theta$.
while not done do
    sample data ($X$=[$x_1, \ldots, x_b$], $Y$=[$y_1, \ldots, y_b$]) $\in \mathbb{D}$
    $X_{adv}^0 = X$
    for k in [1, 2, ..., T]:
        $G^k = \nabla_{X_{adv}^{k-1}} L(\mathcal{M}_\theta(X_{adv}^{k-1}), Y)$
        obtain $G_{ens}^k$ via Eq 4
        obtain $X_{adv}^k$ via Eq 5
    end for
    for each source model $\mathcal{F}_i \in [\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_N]$, do
        evaluate $X_{adv}^T$ on $\mathcal{F}_i$ and obtain $L(\mathcal{F}_i(X_{adv}^T), Y)$
    end for
    $\theta = \theta + \alpha \cdot \nabla_\theta \sum_i^N L(\mathcal{F}_i(X_{adv}^T), Y)$
return $\theta$
```

$$\arg\max_\theta \mathbb{E}_{(x,y)\sim D}\big[\textstyle\sum_{i=1}^N L(\underbrace{\mathcal{F}_i(\overbrace{\mathcal{A}(\mathcal{M}_\theta, x, y)}^{\text{AE}})}_{\mathcal{F}_i's \text{ prediction for AE}}, y)\big], \quad (3)$$

where D is the distribution of training data. The structure of this objective and the training procedure can be illustrated in Figure 1, where we can view it as a meta-learning or bi-level optimization method. At the lower level, the AE is generated by a white-box attack (usually gradient ascent) on MSM, while at the higher level, we feed the AE to the source models to compute the robust loss. Solving Eq 3 will find an MSM where attacking it leads to stronger transferable AEs. The optimization steps of Eq 3 are detailed below.

First, $\mathcal{A}$ should be some strong white-box attacks, such as FGSM or PGD. However, directly using those attacks will make the gradient of meta training objective Eq 3 ill-defined since the sign function in both FGSM and PGD introduce a discrete operation. This results in that the gradient back-propagating through sign be zero and further prohibits the training of the MSM.

To overcome this challenge, we design $\mathcal{A}$ as an approximation of PGD and denote it as Customized PGD. Gradient Calculation subsection will show more analysis about how the sign function in PGD prohibits back-propagation and how Customized PGD enables the back-propagation. The crucial difference between PGD and the Customized PGD is the operation to the gradient $\nabla_{x_{adv}^{k-1}} L(\mathcal{M}_\theta(x_{adv}^{k-1}), y)$, where L is cross entropy. We simplify the vanilla gradient

$\nabla_{x_{adv}^k} L(\mathcal{M}_\theta(x_{adv}^k), y)$ at the k-th step as g^k, and generate another map g_{ens}^k via Eq 4:

$$\begin{cases} g_1^k = \frac{g^k}{\text{sum}(\text{abs}(g^k))} \\ g_t^k = \frac{2}{\pi} \cdot \text{arctan}(\frac{g^k}{\text{mean}(\text{abs}(g^k))}) \\ g_s^k = \text{sign}(g^k) \\ g_{ens}^k = g_1^k + \gamma_1 \cdot g_t^k + \gamma_2 \cdot g_s^k \end{cases} \tag{4}$$

Note that we set $\gamma_1 = \gamma_2 = 0.01$ as default for all experiments. Both g_1^k and g_t^k ensure the objective in Eq 3 be differentiable with respect to the MSM's weight θ; arctan$(\cdot)$ is a smooth approximation of sign and $\frac{1}{\text{mean}(\text{abs}(g^k))}$ prevents arctan from falling into the saturation or linear region. The item $\gamma_2 \cdot g_s^k$ provides the lower-bound for each pixel's perturbation in g_{ens}^k. Experiments in Ablation Study demonstrate the importances of g_t^k and g_s^k for Customized PGD. With Eq 4, the Customized PGD conducts the following update to generate AE:

$$x_{adv}^k = \text{Clip}(x_{adv}^{k-1} + \frac{\epsilon_c}{T} \cdot g_{ens}^{k-1}). \tag{5}$$

Note that ϵ_c differs from the perturbation ϵ in FGSM and PGD because g_{ens}^{k-1} in our update is not a sign vector and its size will depend on the magnitude of the original gradient. Finally, we get x_{adv}^T after T iterations of Eq 5.

Second, we feed x_{adv}^T into N source models and calculate the corresponding adversarial losses $L(\mathcal{F}_i(x_{adv}^T), y)$ for all $i = 1, \ldots, N$. Larger losses of the N source models indicate a higher likelihood that x_{adv}^T fooling the MSM can transfer to other models.

Third, we optimize the MSM by maximizing the objective function defined in Eq 3, which can be written as

$$\theta' = \theta + \alpha \cdot \textstyle\sum_{i=1}^N \nabla_\theta L(\mathcal{F}_i(x_{adv}^T), y), \tag{6}$$

where x_{adv}^T can be written as a function of θ by unrolling the attack update rule Eq 5 T times. We will show how to explicitly compute the gradient in Gradient Calculation subsection. With this training procedure, the MSM is trained to learn a particular weight with which the white-box AEs fooling it can also fool other models. We summarize the training and testing of MTA in Algorithm 1 and Appendix, respectively. Each capitalized notation represents a batch of the variable denoted with lower case (*i.e.*, X denotes a batch of x). Note that Customized PGD is just a continuous approximation of PGD used to train the MSM. In the inference phase, we use standard attacks such as PGD to craft AEs on the MSM.

Gradient Calculation

In the calculation we set both N and T in Eq 6 to 1, so the gradient in Eq 6 is $\nabla_\theta L(\mathcal{F}_1(x_{adv}^1), y)$. According to Eq 5, we can replace x_{adv}^1 in Eq 6 with Clip$(x_{adv}^0 + \epsilon_c \cdot g_{ens}^0)$, where x_{adv}^0 equals to x. For simplicity, we ignore the clip function in the analysis and simplify the derivation as $\nabla_\theta L(\mathcal{F}_1(x + \epsilon_c \cdot g_{ens}^0), y)$. By chain rule and since x is independent to θ, we can further rewrite this as

$$\frac{\partial L(\mathcal{F}_1(x + \epsilon_c \cdot g_{ens}^0), y)}{\partial g_{ens}^0} \cdot \frac{\partial g_{ens}^0}{\partial \theta}. \tag{7}$$

By replacing g_{ens}^0 with Eq 4, the second term of Eq 7 can be expanded as

$$\nabla_\theta g_{ens}^0 = \nabla_\theta g_1^0 + \gamma_1 \cdot \nabla_\theta g_t^0 + \gamma_2 \cdot \nabla_\theta g_s^0. \tag{8}$$

Note that g_s^0 equals to sign(g^0) and the sign function introduces discrete operation so that the gradient of g_s^0 with respect to θ becomes 0 (unless $g^0 = 0$). Therefore, $\nabla_\theta g_{ens}^0$ can be further written as

$$\begin{aligned} \nabla_\theta g_{ens}^0 &= \nabla_\theta g_1^0 + \gamma_1 \cdot \nabla_\theta g_t^0 \\ &= \nabla_\theta(\frac{\nabla_x L(\mathcal{M}_\theta(x), y)}{\text{sum}(\text{abs}(\nabla_x L(\mathcal{M}_\theta(x), Y)))}) \\ &+ \gamma_1 \cdot \nabla_\theta(\text{arctan}(\frac{\nabla_x L(\mathcal{M}_\theta(x), y)}{\text{mean}(\text{abs}(\nabla_x L(\mathcal{M}_\theta(x), y)))})), \end{aligned} \tag{9}$$

where $\nabla_x L(\mathcal{M}_\theta(x), y)$ depends on θ and the second-order derivative of $\nabla_x L(\mathcal{M}_\theta(x), y)$ *w.r.t* θ can be obtained with lots of deep learning libraries. In summary, by integrating Eqs.6-9, the MSM can be optimized by an SGD-based optimizer. *Eqs.6-9 can also clearly explain why Customized PGD enables the training of MSM and why vanilla PGD blocks that*. When using vanilla PGD to attack the MSM and generate AE, Eq.7 will turn to $\frac{\partial L(\mathcal{F}_1(x+\epsilon_c \cdot g_s^0), y)}{\partial g_s^0} \cdot \frac{\partial g_s^0}{\partial \theta}$, where $\frac{\partial g_s^0}{\partial \theta}$ is zero because g_s^0 is the signed discrete gradient sign(g^0).

Experiment

We conduct experiments to show that the proposed method, under the same set of source models, can generate stronger transferable AEs than existing transfer attack methods.

Our general experimental settings: ***1)*** We conduct experiments on both Cifar-10 and ImageNet. ***2)*** We compare the proposed MTA with ten state-of-the-art transferable adversarial attack methods, including MI, DI, TI, SGM, SI-NI-TIDIM, AEG, IR, MGAA, FIA and DA-TIM. AEG is compared only on Cifar-10 because the official AEG is evaluated only on small scale datasets (Mnist and Cifar-10), and it is computational costly to train the perturbation generator on large-scale datasets. ***3)*** Since the number of attack iterations T is different between training and testing, we denote it as T_t in training and T_v in testing respectively to avoid confusion. ***4)*** When training the MSM, we use the Customized PGD with γ_1=γ_2=0.01 to attack the MSM, which is shown in Algorithm 1. When evaluating, we use PGD with T_v=10 and ϵ=15 to attack the MSM, which is shown in Algorithm 2 in Appendix. ***5)*** When using the baseline methods to generate AEs on multiple source models, we follow MI to ensemble the logits of the source models before loss calculation. ***6)*** We use source models to train the MSM and use target models to evaluate the transferability of the AEs generated on MSM. ***7)*** For fair comparisons between MTA and baselines, we implement baselines with the number of iterations T=10 and ϵ=15, and other hyper-parameters are tuned for their best possible performances (implementations are detailed in Appendix). ***8)*** Visualizations, computational cost analyses, simplified tensorflow code, more implementation details of MTA and baselines, and more experiments (*e.g.*, targeted transfer attack, attacks with ϵ=8, comparison between MTA and TAIG(Huang and Kong 2022), attacking ViT (Dosovitskiy et al. 2021), no overlapping training images between source and target models) will be presented in Appendix.

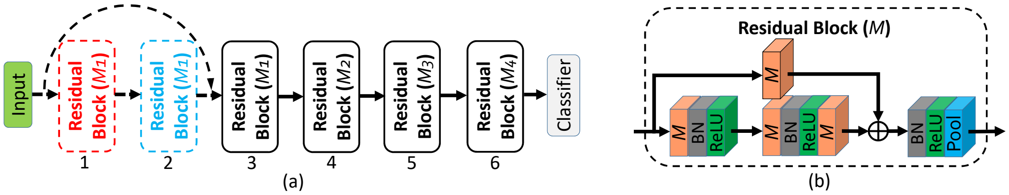

Figure 2: (a) Structures of ResNet-13 and -19. ResNet-13 contains the top four solid-line blocks and the classifier. ResNet-19 contains all the six blocks and the classifier. The parameter $M*$ of each block denotes the number of filters of its convolution layers. (b) Detailed structure of residual block. Orange cube is convolution layer and the number on it denotes its number of filters. Pool in the sixth block is global-average pooling while all the other pool is max-pooling with both stride and kernel size of 2×2. The convolution layer in the shortcut path uses 1×1 kernel size while all the other convolution layers use 3×3.

Method	MN-V3	SN-V1	SN-V2	SN-A	SN-B	Res-34$_{adv}$	SE-50$_{adv}$	FAST
DI	57.8%	72.5%	56.4%	65.7%	64.6%	18.1%	18.7%	8.1%
MI	70.2%	85.6%	72.6%	83.7%	83.0%	28.6%	38.0%	15.9%
AEG	90.8%	92.5%	85.8%	91.3%	91.0%	**62.5%**	53.3%	17.3%
IR	59.3%	77.9%	62.5%	71.6%	69.1%	21.7%	23.2%	11.5%
MGAA	71.0%	74.3%	69.7%	84.3%	82.6%	55.2%	52.9%	**17.9%**
MTA	**91.8%**	**98.4%**	**90.9%**	**94.9%**	**93.8%**	56.1%	**58.8%**	17.0%
MTA$_{\gamma_1=0}$	70.0%	80.9%	68.5%	58.5%	59.4%	30.4%	35.1%	13.2%
MTA$_{\gamma_2=0}$	90.0%	98.2%	90.5%	93.9%	93.1%	55.8%	57.2%	16.5%
MTA$_{dense}$	86.9%	96.2%	87.1%	89.0%	87.6%	53.0%	55.9%	16.1%

Table 1: Results on eight Cifar-10 target models: MobileNet-V3 (MN-V3), ShuffleNet-V1 (SN-V1), -V2 (SN-V2), SqueezeNet-A (SN-A), -B (SN-B), adversarially trained ResNet-34 (Res-34$_{adv}$) and SeResNet-50 (SeRes-50$_{adv}$), and the robust model FAST.

Experiments on Cifar-10

Source and Target Models . We use 8 source models including ResNet-10, -18, -34 (He et al. 2016), SeResNet-14, -26, -50 (Hu, Shen, and Sun 2018), MobileNet-V1 (Howard et al. 2017), and -V2 (Sandler et al. 2018) to train the MSM. To ensure mismatches between the source and target models and to avoid saturated transfer attack performances (*i.e.*, attack success rates close to 100%), we select the 8 target models including MobileNet-V3 (Howard et al. 2019), ShuffleNet-V1, -V2 (Zhang et al. 2018), SqueezeNet-A, -B (Iandola et al. 2016), adversarially trained ResNet-34 and SeResNet-50, and robust model FAST(Wong, Rice, and Kolter 2020). FAST is a public robust model available at RobustBench[1]. The network architectures of all the other 15 source and target models are defined on GitHub repositories[2,3,4]. We train these 15 models and describe the training details of these models in Appendix. The trained models and the code will be released to the community for reproducibility.

Training the MSM . The default network architecture of the MSM is ResNet-13 shown in Figure 2, with $M1$, $M2$, $M3$, and $M4$ set to 64, 128, 256, and 512, respectively. We use the 8 source models to train the MSM for 60 epochs with the number of attack steps T_t of 7. ϵ_c of the Customized PGD is initialized to 1,600 and is exponentially decayed by 0.9× for every 4,000 iterations. The learning rate α and the batch size are set to 0.001 and 64, respectively.

Evaluating the MSM . On each target model, we only attack the correctly classified test images because attacking wrongly classified clean images is less meaningful.

Experimental Results . As Table 1 shows, MTA performs the best on almost all target models. On Res-34$_{adv}$ and FAST, MTA performs comparably to AEG and MGAA, and outperforms the other methods. The possible reason why AEG outperforms MTA on Res-34$_{adv}$ is that it trains a perturbation generator to fool robust classifiers and simultaneously adversarially trains the robust classifiers to be robust to the generated perturbations. So it naturally transfers better to some adversarially trained target models as it already "sees" adversarially trained models in its training phase. However it performs worse in all other models. MTA$_{\gamma_1=0}$, MTA$_{\gamma_2=0}$, and MTA$_{dense}$ will be discussed in Ablation Study.

Experiments on Imagenet

Source and Target Models . We directly use the public trained ImageNet models[5,6,7] including ResNet-50, -101, -152, DenseNet-121, -161 (Huang et al. 2017), Inception-V3 (Szegedy et al. 2016), -V4 (Szegedy et al. 2017), Inception-ResNet-V2, Inception-V3$_{ens3}$, Inception-V3$_{ens4}$, and Inception-ResNet-V2$_{ens}$. The former eight models are normally trained models while the latter three are secure models trained by ensemble adversarial training (Tramèr et al. 2017). We shorten these models as Res-50, Res-101, Res-152,

[1] https://github.com/RobustBench/robustbench

[2] https://github.com/yxlijun/cifar-tensorflow

[3] https://github.com/TropComplique/ShuffleNet-tensorflow

[4] https://github.com/TropComplique/shufflenet-v2-tensorflow

[5] https://github.com/pudae/tensorflow-densenet

[6] https://github.com/tensorflow/models/tree/r1.12.0/research/slim

[7] https://github.com/tensorflow/models/tree/r1.12.0/research/adv_imagenet_models

Source	Method	Inc-V3	Inc-V4	IncRes-V2	Res-152	Inc-V3$_{ens3}$	Inc-V3$_{ens4}$	IncRes-V2$_{ens}$
Inc-V3	DI	99.3%	35.2%	28.2%	22.3%	5.1%	4.3%	2.5%
	MI	**99.9%**	38.1%	35.8%	29.6%	9.1%	8.8%	4.5%
	MI-DI	99.1%	61.7%	57.3%	48.0%	13.6%	12.0%	6.5%
	IR	95.6%	33.6%	28.1%	15.9%	5.1%	5.5%	3.0%
	FIA	95.2%	69.0%	66.8%	52.5%	29.3%	27.7%	14.9%
	DA-TIM	99.2%	62.9%	58.3%	50.6%	46.9%	43.4%	31.7%
	SI-NI-TIDIM	99.0%	79.6%	73.5%	69.3%	58.1%	53.8%	37.0%
	MTA	99.9%	90.9%	87.3%	74.1%	67.7%	39.3%	26.1%
	MTA-IR	95.6%	**95.5%**	**93.2%**	**85.0%**	**83.5%**	**56.9%**	**40.7%**
Inc-V4	DI	44.9%	97.6%	30.5%	26.7%	5.9%	5.5%	3.3%
	MI	52.7%	**99.6%**	41.8%	37.3%	12.4%	11.0%	5.8%
	MI-DI	69.1%	97.1%	58.7%	49.3%	16.6%	14.1%	8.2%
	IR	46.5%	94.9%	33.2%	18.9%	8.1%	8.8%	4.9%
	FIA	63.6%	87.5%	55.2%	45.9%	28.5%	26.1%	16.8%
	DA-TIM	69.5%	99.3%	63.7%	55.7%	49.1%	44.3%	37.2%
	SI-NI-TIDIM	82.5%	98.3%	76.1%	69.1%	61.3%	56.8%	43.6%
	MTA	87.3%	99.5%	84.7%	73.1%	61.7%	38.2%	29.0%
	MTA-IR	**93.3%**	94.9%	**90.5%**	**82.0%**	**77.2%**	**57.7%**	**44.9%**
IncRes-V2	DI	46.9%	42.0%	90.7%	29.5%	8.6%	6.5%	5.5%
	MI	53.2%	45.2%	97.3%	38.8%	16.2%	13.3%	9.7%
	MI-DI	64.7%	61.7%	90.3%	50.6%	23.7%	18.6%	13.6%
	IR	49.7%	44.9%	90.2%	25.2%	13.6%	11.2%	10.9%
	FIA	63.2%	57.8%	79.6%	51.3%	35.1%	30.3%	25.0%
	DA-TIM	70.3%	66.7%	96.8%	58.1%	52.8%	45.6%	44.3%
	SI-NI-TIDIM	**79.6%**	**78.5%**	97.8%	71.0%	**63.1%**	**60.8%**	**53.6%**
	MTA	44.7%	41.7%	**98.0%**	57.9%	23.5%	19.4%	17.5%
	MTA$_{Inc}$	64.3%	51.7%	**98.0%**	76.0%	46.2%	39.3%	27.5%
	MTA-IR$_{Inc}$	66.2%	52.3%	90.2%	**78.3%**	49.0%	42.2%	31.7%
Res-152	DI	51.8%	48.1%	40.6%	99.5%	9.7%	8.3%	6.2%
	MI	50.2%	44.9%	39.4%	**99.6%**	13.9%	12.0%	7.8%
	MI-DI	**76.2%**	73.3%	**69.5%**	**99.6%**	24.6%	21.1%	12.7%
	IR	42.3%	33.8%	34.1%	95.3%	22.0%	20.6%	16.2%
	FIA	73.8%	67.2%	67.9%	99.3%	48.0%	43.7%	30.4%
	DA-TIM	69.0%	65.1%	66.3%	99.0%	60.7%	56.9%	52.5%
	SI-NI-TIDIM	75.1%	71.8%	67.3%	98.8%	**66.0%**	61.5%	54.9%
	MTA	70.7%	77.5%	62.8%	99.1%	53.0%	59.2%	56.3%
	MTA-IR	72.8%	**78.0%**	64.3%	95.3%	54.9%	**63.0%**	**59.3%**

Table 2: Transfer attack results on seven black-box networks when using one source model.

DN-121, DN-161, Inc-V3, Inc-V4, IncRes-V2, Inc-V3$_{ens3}$, Inc-V3$_{ens4}$, and IncRes-V3$_{ens}$.

Training the MSM . The default network architecture of the MSM is ResNet-19 shown in Figure 2, with $M1$, $M2$, $M3$, and $M4$ set to 32, 80, 200, and 500, respectively. We follow previous works MI and SGM to evaluate the transferability of AEs in two settings: using a single source model and using multiple source models. We set the input shape of the MSM to 224×224. When the resolution of the source model differs from that of the MSM, we resize the AE x^T_{adv} to the resolution of the source model before feeding it into the source model. Appendix will show more training details.

Evaluating the MSM . Following the official testing data settings in the papers of DI and SGM, we also randomly choose 5,000 validation images from ImageNet that are correctly classified by all models for evaluation. Note that, when the resolutions of the MSM and the target model are different, we resize the AE x^T_{adv} to the resolution of the target model. For instance, when attacking Inc-V3 whose resolution is 299×299, we first resize x^T_{adv} from 224×224 to 299×299 and then use the resized x^T_{adv} to attack Inc-V3.

Using One Source Model. Table 2 reports the experimental results. MI-DI is a combination of MI and DI. SI-NI-TIDIM is a combination of SI-NI, TI, DI, and MI. DA-TIM is a combination of DA, TI, and MI. MGAA is not compared here because it needs a model zoo containing several source models to construct meta-tasks, which is costly. Obviously, MTA outperforms the baselines on most of the testing scenes. Compared with FIA, MTA improves the transfer attack success rates by about 31.7%, 30.7%, 41.1%, 131.1%, 41.9%, and 75.2% when using the Inc-V3 source model and attacking the target models (Inc-V4, IncRes-152, Res-152, Inc-V3$_{ens3}$, Inc-V3$_{ens4}$, IncRes-V2$_{ens}$). MTA-IR combines MTA with IR. In evaluation, MTA generates AEs by attacking the MSM using vanilla PGD while MTA-IR generates AEs by attacking the MSM using IR. Compared with MTA, MTA-IR improves the attack success rates by about 5.1%, 6.8%, 14.7%, 23.3%, 44.8%, and 55.9% on the target models when using the Inc-V3 source model, indicating that existing

Source	Method	Inc-V3	Inc-V4	IncRes-V2	Res-101	Inc-V3$_{ens3}$	Inc-V3$_{ens4}$	IncRes-V2$_{ens}$
Res-50 + Res-152 + DN-161	TI-DI	60.6%	59.2%	50.2%	86.8%	54.9%	56.2%	46.9%
	SGM	81.8%	74.7%	73.9%	**98.7%**	54.9%	50.1%	38.7%
	SGM-DI	86.2%	83.9%	81.6%	98.3%	69.8%	64.9%	54.4%
	SGM-MI	86.5%	84.3%	82.7%	98.2%	71.1%	67.4%	60.8%
	IR	75.2%	70.3%	67.9%	90.6%	51.7%	49.1%	37.5%
	MTA	90.4%	94.3%	87.6%	97.5%	75.5%	79.7%	79.0%
	MTA-IR	**93.1%**	**95.8%**	**90.5%**	98.3%	**83.6%**	**87.2%**	**85.0%**
Res-50 + Inc-V1 + DN-121	TI-DI	61.9%	58.5%	49.0%	79.7%	53.1%	54.1%	41.9%
	SGM	62.7%	53.5%	50.9%	89.1%	33.8%	30.4%	19.3%
	SGM-DI	87.2%	83.6%	**79.5%**	95.1%	59.6%	54.9%	37.9%
	SGM-MI	82.8%	76.0%	74.3%	**95.9%**	62.2%	59.7%	45.3%
	IR	76.5%	70.9%	64.0%	92.1%	51.3%	44.9%	31.5%
	MTA	91.7%	86.4%	76.0%	93.6%	81.7%	**79.6%**	**61.6%**
	MTA-IR	**92.8%**	**87.9%**	77.2%	93.8%	**82.6%**	79.3%	61.5%
Res-50 + Inc-V1	TI-DI	51.6%	46.9%	38.4%	73.4%	43.4%	44.2%	32.8%
	SGM	46.1%	35.6%	33.3%	82.0%	22.1%	19.5%	12.3%
	SGM-DI	79.2%	70.6%	68.7%	91.9%	47.9%	42.0%	28.1%
	SGM-MI	71.9%	62.0%	61.3%	94.3%	49.6%	47.2%	33.8%
	IR	60.2%	49.0%	46.2%	93.0%	36.5%	30.6%	21.0%
	MTA	84.1%	88.8%	78.4%	93.9%	60.6%	61.1%	55.1%
	MTA-IR	**87.6%**	**91.8%**	**83.9%**	**95.2%**	**71.5%**	**72.6%**	**63.7%**

Table 3: Transfer attack results on seven black-box models when using multiple source models.

transferable attack methods can further improve MTA.

When using IncRes-V2 source model, MTA sometimes performs not good, possibly because the MSM with ResNet-19 backbone is unsuitable to be trained to attack IncRes-V2. We then replace the backbone ResNet-19 with another simplified Inception network (the architecture will be shown in Appendix) and retrain the MSM, and denote the newly trained MSM as MTA$_{Inc}$. Compared with ResNet-19, the simplified Inception backbone is more similar to IncRes-V2 so that MTA$_{Inc}$ turns to be easier to generate adversarial attacks to fool IncRes-V2 than MTA, leading to easier convergence of MTA$_{Inc}$. The results show that MTA$_{Inc}$ outperforms not only MTA but also most of the compared methods, indicating 1) the effectiveness of the proposed MTA and 2) MTA can be further improved by using more suitable backbones.

Using Multiple Source Models . Table 3 reports the experimental results of using multiple source models. We use three source model groups (Res-50+Res-152+DN161, Res-50+Inc-V1+DN-121, Res-50+Inc-V1) to train the MSM, respectively, and use seven target models (Inc-V3, Inc-V4, InvRes-V2, Res-101, Inc-V3$_{ens3}$, Inc-V3$_{ens4}$, IncRes-V2$_{ens}$) to evaluate the transferability of the attacks to the MSM. SGM-X is the combination of SGM and X (X=DI or MI). TI-DI is the combination of TI and DI. The results show that MTA outperforms the baselines in almost all testing scenes, especially when attacking defensive models. For instance, compared with SGM-DI, MTA improves the transfer attack success rates by 6.2%, 25.8%, 14.1%, 2.2%, 26.5%, 45.5%, and 96.1% on the seven target models when using Res-50 and Inc-V1 source models. Besides, MTA-IR outperforms MTA.

Ablation Study

Network Structure . The comparison between MTA and MTA$_{Inc}$ in Table 2 has validated the effect of backbone on the MSM. Here we further verify the effect of backbone by replacing the backbone from ResNet-13 to DenseNet-22BC (Appendix shows the structure of DenseNet-22BC) and denote the newly trained MSM as MTA$_{dense}$(in Table 1). The comparisons among MTA, MTA$_{dense}$, and the baselines indicate that 1) backbone affects the performance of MTA; 2) MTA outperforms the baselines with various backbones.

The Effects of γ_1 and γ_2 . We verify how γ_1 and γ_2 in Eq. 5 affect the transfer attack performance on Cifar-10. Here we set γ_1 and γ_2 to zero respectively, and amplify ϵ_c appropriately to offset the decrease of the perturbation size caused by zeroing γ_1 or γ_2. We denote the two newly performed MTA as MTA$_{\gamma_1=0}$ and MTA$_{\gamma_2=0}$. The results shown in Table 1 indicate that the performances of MTA are greatly damaged by setting γ_1 to zero. Setting γ_2 to zero also decreases MTA's performances, but the effect is smaller than that of γ_1. Overall, the two experiments demonstrate the indispensability of Customized PGD for the proposed MTA framework.

Conclusion

Existing query free black-box adversarial attack methods directly use image classification models as surrogate models to generate transferable adversarial attacks to attack black-box models neglecting the study of surrogate models. In this paper, we propose a novel framework called meta-transfer attack (MTA) to improve the transferability of adversarial attacks via training an MSM using these surrogate models. To enable and improve the training of the MSM, a novel Customized PGD is also developed. Through extensive experiments, we validate that by attacking the trained MSM, we can get transferable adversarial attacks that are generalizable to attack black-box target models with much higher success rates than existing methods, demonstrating the effectiveness of the proposed MTA framework.

Acknowledgments

This work was supported in part by Beijing Academy of Artificial Intelligence(BAAI), and in part by the Chinese National Natural Science Foundation Projects (Grant No. 62206259).

References

Bose, A. J.; Gidel, G.; Berrard, H.; Cianflone, A.; Vincent, P.; Lacoste-Julien, S.; and Hamilton, W. L. 2020. Adversarial Example Games. *Advances in neural information processing systems*.

Brendel, W.; Rauber, J.; and Bethge, M. 2017. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*.

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

Chen, J.; Jordan, M. I.; and Wainwright, M. J. 2020. HopSkipJumpAttack: a query-efficient decision-based adversarial attack. In *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE.

Chen, P.-Y.; Zhang, H.; Sharma, Y.; Yi, J.; and Hsieh, C.-J. 2017. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, 15–26.

Cheng, M.; Le, T.; Chen, P.-Y.; Yi, J.; Zhang, H.; and Hsieh, C.-J. 2018. Query-efficient hard-label black-box attack: An optimization-based approach. *arXiv preprint arXiv:1807.04457*.

Cheng, M.; Singh, S.; Chen, P. H.; Chen, P.-Y.; Liu, S.; and Hsieh, C.-J. 2020. Sign-OPT: A Query-Efficient Hard-label Adversarial Attack. In *international conference on learning representations*.

Cheng, S.; Dong, Y.; Pang, T.; Su, H.; and Zhu, J. 2019. Improving Black-box Adversarial Attacks with a Transfer-based Prior. In *Advances in neural information processing systems 32 (NIPS 2019)*, 10932–10942.

Croce, F.; and Hein, M. 2020a. Minimally distorted adversarial examples with a fast adaptive boundary attack. In *International Conference on Machine Learning*, 2196–2205. PMLR.

Croce, F.; and Hein, M. 2020b. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *International Conference on Machine Learning*, 2206–2216. PMLR.

Demontis, A.; Melis, M.; Pintor, M.; Jagielski, M.; Biggio, B.; Oprea, A.; Nita-Rotaru, C.; and Roli, F. 2019. Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks. *USENIX Security Symposium*, 321–338.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

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

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

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; Uszkoreit, J.; and Houlsby, N. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *International Conference on Learning Representations*.

Du, J.; Zhang, H.; Zhou, T. J.; Yang, Y.; and Feng, J. 2020. Query-efficient Meta Attack to Deep Neural Networks. *International Conference on Learning Representations*.

Finn, C.; Abbeel, P.; and Levine, S. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, 1126–1135. PMLR.

Ganeshan, A.; BS, V.; and Babu, R. V. 2019. Fda: Feature disruptive attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8069–8079.

Gao, L.; Zhang, Q.; Song, J.; Liu, X.; and Shen, H. T. 2020. Patch-wise attack for fooling deep neural network. In *European Conference on Computer Vision*, 307–322. Springer.

Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2014. Explaining and harnessing adversarial examples. *international conference on learning representations*.

Guo, Y.; Li, Q.; and Chen, H. 2020. Backpropagating Linearly Improves Transferability of Adversarial Examples. In *Advances in neural information processing systems 33 (NIPS 2020)*.

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

Howard, A.; Sandler, M.; Chu, G.; Chen, L.-C.; Chen, B.; Tan, M.; Wang, W.; Zhu, Y.; Pang, R.; Vasudevan, V.; et al. 2019. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 1314–1324.

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

Hu, J.; Shen, L.; and Sun, G. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 7132–7141.

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

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

Huang, T.; Menkovski, V.; Pei, Y.; Wang, Y.; and Pechenizkiy, M. 2022. Direction-aggregated attack for transferable adversarial examples. *ACM Journal on Emerging Technologies in Computing Systems (JETC)*, 18(3): 1–22.

Huang, Y.; and Kong, A. W.-K. 2022. Transferable Adversarial Attack based on Integrated Gradients. In *International Conference on Learning Representations*.

Huang, Z.; and Zhang, T. 2020. Black-box adversarial attack with transferable model-based embedding. *International Conference on Learning Representations*.

Iandola, F. N.; Han, S.; Moskewicz, M. W.; Ashraf, K.; Dally, W. J.; and Keutzer, K. 2016. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. *arXiv preprint arXiv:1602.07360*.

Ilyas, A.; Engstrom, L.; Athalye, A.; and Lin, J. 2018. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, 2137–2146. PMLR.

Ilyas, A.; Santurkar, S.; Tsipras, D.; Engstrom, L.; Tran, B.; and Madry, A. 2019. Adversarial examples are not bugs, they are features. *arXiv preprint arXiv:1905.02175*.

Kaidi, X.; Sijia, L.; Pu, Z.; Pin-Yu, C.; Huan, Z.; Quanfu, F.; Deniz, E.; Yanzhi, W.; and Xue, L. 2019. Structured Adversarial Attack: Towards General Implementation and Better Interpretability. *International Conference on Learning Representations*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. Technical Report TR-2009, Univercity of Toronto, Toronto, Canada.

Kurakin, A.; Goodfellow, I. J.; and Bengio, S. 2018. Adversarial examples in the physical world. *Artificial intelligence safety and security*, 99–112.

Li, H.; Xu, X.; Zhang, X.; Yang, S.; and Li, B. 2020a. QEBA: Query-Efficient Boundary-Based Blackbox Attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1221–1230.

Li, Q.; Guo, Y.; and Chen, H. 2020. Practical No-box Adversarial Attacks against DNNs. *Advances In Neural Information Processing Systems 2020*.

Li, Y.; Bai, S.; Zhou, Y.; Xie, C.; Zhang, Z.; and Yuille, A. 2020b. Learning Transferable Adversarial Examples via Ghost Networks. *AAAI*, 11458–11465.

Lin, J.; Song, C.; He, K.; Wang, L.; and Hopcroft, J. E. 2020. Nesterov accelerated gradient and scale invariance for adversarial attacks. *International Conference on Learning Representations*.

Liu, Y.; Chen, X.; Liu, C.; and Song, D. 2017. Delving into Transferable Adversarial Examples and Black-box Attacks. *international conference on learning representations*.

Liu, Y.; Jiang, M.; and Jiang, T. 2022. Transferable Adversarial Examples Based on Global Smooth Perturbations. *Computers & Security*, 102816.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards deep learning models resistant to adversarial attacks. *international conference on learning representations*.

Maksym, A.; Francesco, C.; Nicolas, F.; and Matthias, H. 2020. Square Attack: a query-efficient black-box adversarial attack via random search. *european conference on computer vision*, 484–501.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2574–2582.

Naseer, M. M.; Khan, S. H.; Khan, M. H.; Shahbaz Khan, F.; and Porikli, F. 2019. Cross-domain transferability of adversarial perturbations. *Advances in Neural Information Processing Systems*, 32: 12905–12915.

Papernot, N.; McDaniel, P.; Goodfellow, I.; Jha, S.; Celik, Z. B.; and Swami, A. 2017. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, 506–519.

Qin, Y.; Zhang, W.; Wang, Z.; Zhao, C.; and Shi, J. 2020. Layer-Wise Adaptive Updating for Few-Shot Image Classification. *IEEE Signal Processing Letters*, 27: 2044–2048.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

Sriramanan, G.; Addepalli, S.; Baburaj, A.; and Babu, V. R. 2020. Guided Adversarial Attack for Evaluating and Enhancing Adversarial Defenses. *Advances In Neural Information Processing Systems*.

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31.

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

Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, J. I.; and Fergus, R. 2014. Intriguing properties of neural networks. *international conference on learning representations*.

Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; and McDaniel, P. 2017. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*.

Wang, L.; Zhang, H.; Yi, J.; Hsieh, C.-J.; and Jiang, Y. 2020. Spanning attack: reinforce black-box attacks with unlabeled data. *Machine Learning*, 109(12): 2349–2368.

Wang, X.; and He, K. 2021. Enhancing the Transferability of Adversarial Attacks Through Variance Tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1924–1933.

Wang, X.; Ren, J.; Lin, S.; Zhu, X.; Wang, Y.; and Zhang, Q. 2021a. A unified approach to interpreting and boosting adversarial transferability. *International Conference on Learning Representations*.

Wang, Z.; Guo, H.; Zhang, Z.; Liu, W.; Qin, Z.; and Ren, K. 2021b. Feature Importance-aware Transferable Adversarial Attacks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Wong, E.; Rice, L.; and Kolter, J. Z. 2020. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations*.

Wu, D.; Wang, Y.; Xia, S.-T.; Bailey, J.; and Ma, X. 2020a. Skip connections matter: On the transferability of adversarial examples generated with resnets. *international conference on learning representations*.

Wu, K.; Wang, A.; and Yu, Y. 2020. Stronger and Faster Wasserstein Adversarial Attacks. *International Conference on Machine Learning*, 10377–10387.

Wu, L.; Zhu, Z.; Tai, C.; et al. 2018. Understanding and enhancing the transferability of adversarial examples. *arXiv preprint arXiv:1802.09707*.

Wu, W.; Su, Y.; Chen, X.; Zhao, S.; King, I.; Lyu, R. M.; and Tai, Y.-W. 2020b. Boosting the Transferability of Adversarial Samples via Attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1158–1167.

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

Yuan, Z.; Zhang, J.; Jia, Y.; Tan, C.; Xue, T.; and Shan, S. 2021. Meta Gradient Adversarial Attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Zhang, X.; Zhou, X.; Lin, M.; and Sun, J. 2018. ShuffleNet: An Extremely Efficient Convolutional Neural Network for Mobile Devices. *computer vision and pattern recognition*.

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