MODEL MIMIC ATTACK: KNOWLEDGE DISTILLATION FOR PROVABLY TRANSFERABLE ADVERSARIAL EXAMPLES

Anonymous authors
Paper under double-blind review

ABSTRACT

The vulnerability of artificial neural networks to adversarial perturbations in the black-box setting is widely studied in the literature. The majority of attack methods to construct these perturbations suffer from an impractically large number of queries required to find an adversarial example. In this work, we focus on knowledge distillation as an approach to conduct transfer-based black-box adversarial attacks and propose an iterative training of the surrogate model on an expanding dataset. This work is the first, to our knowledge, to provide provable guarantees on the success of knowledge distillation-based attack on classification neural networks: we prove that if the student model has enough learning capabilities, the attack on the teacher model is guaranteed to be found within the finite number of distillation iterations.

1 INTRODUCTION

The robustness of deep neural networks to input perturbations is a crucial property to integrate them into various safety-demanding areas of machine learning, such as self-driving cars, medical diagnostics, and finances. Although neural networks are expected to produce similar outputs for similar inputs, they are long known to be vulnerable to adversarial perturbations [Szegedy et al. (2014)] – small, carefully crafted input transformations that do not change the semantics of the input object, but force a model to produce a predefined decision. The majority of methods to study the adversarial robustness of neural networks are aimed at crafting adversarial perturbations which indicate that, in general, the predictions of a neural network are unreliable. The most effective and stealthy attacks require access to the model's gradients and are therefore of little practical use on their own [Goodfellow et al. (2014); Madry et al. (2017); Carlini & Wagner (2016)]. However, in real-world scenarios, machine learning models are often deployed as services that are available via APIs. This setting, although poses certain limitations to exploring the robustness of machine learning as a service (MLaaS) models, does not make the computation of adversarial perturbations impossible [Chen et al. (2020); Andriushchenko et al. (2020); Qin et al. (2023); Vo et al. (2024)]. It is possible to compute an adversarial perturbation for the black-box model by either estimating its gradient in the vicinity of the target point Ilyas et al. (2018); Bai et al. (2020) or using random search Andriushchenko et al. (2020) or applying knowledge transfer to obtain an auxiliary model to attack in the white-box setting Li et al. (2023); Gubri et al. (2022).

However, these methods may require a lot of queries to the target model and, in general, are not guaranteed to find an adversarial example. In this paper, we focus on the following research question: is it possible to provably compute an adversarial example for a given black-box classification neural network for a finite number of queries? To answer this question, we propose *Model Mimic Attack*, the framework for conducting a black-box model transfer attack through multiple knowledge distillations.

Knowledge distillation attack methods have been studied extensively in recent years. It is used, for example, to protect intellectual property: the surrogate model obtained by extracting the knowledge of the source one and then is used to create watermarks that help to link the generated content and determine its origin [Yuan et al. (2022); Lukas et al. (2019); Kim et al. (2023); Pautov et al. (2024)]. This approach is also used in attacks on black-box models [Li et al. (2023); Gubri et al. (2022)]. We propose iterative training of a series of surrogate models on an expanding dataset. This approach allows each subsequent surrogate model to better mimic the behavior of the black-box model.

Our contributions are summarized as follows:

1. We propose *Model Mimic Attack*, a score-based black-box model transfer attack via knowledge distillation. The algorithm exploits the behavior of the target teacher network in the vicinity of the target point and yields the set of surrogate student models, which copy the predictions of the target model in the finite set of points. Then, the set of student models is used to compute an adversarial perturbation in the white-box setting, which transfers to the teacher model over a finite number of distillation iterations.

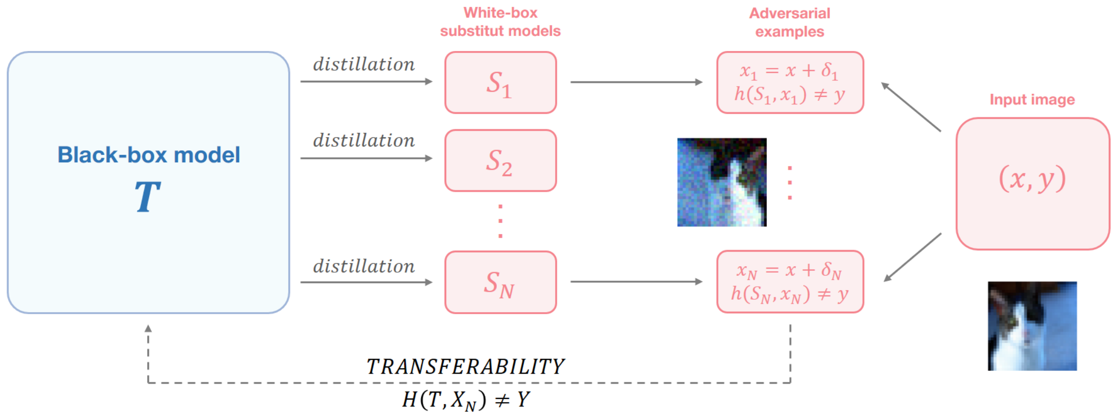

Figure 1: The illustration of the proposed method. Given the black-box teacher model T, the set of student models $S_1, \ldots, S_N$ is obtained via the soft-label knowledge distillation. Each student model is attacked in a white-box manner, and the set of adversarial examples $x_1, \ldots, x_N$ is computed. Note that, according to theoretical analysis, there is an adversarial example x_N for the student model S_N which is transferable to the teacher model T for some $N \in \mathbb{N}$.

2. We are the first, to our knowledge, to theoretically show that the distillation-based model transfer attack *is guaranteed* to find an adversarial perturbation for the black-box teacher model.
3. We experimentally demonstrate the efficiency of the proposed approach over other black-box attack methods in the image classification domain.

2 Related Work

In this section, we provide a brief overview of existing black-box adversarial attacks and applications of knowledge distillation.

2.1 Transferable Adversarial Perturbations

In this work, we focus on the transferability of an adversarial attack from a white-box model to a black-box one, emulating a black-box attack. Black-box adversarial attacks can be divided into two categories: query-based and transfer-based. In a query-based attack, an adversary uses an output of the target model to compute an adversarial example. One way to do this is to estimate the gradient of the model to the input object [Bhagoji et al. (2018); Chen et al. (2017); Ilyas et al. (2019); Guo et al. (2019)]. However, these methods usually require a lot of queries to the target model, which makes them infeasible in practice. In a transfer-based attack, an adversary generates adversarial examples by attacking one or several surrogate models [Liu et al. (2022); Qin et al. (2023)]. The transferability of adversarial examples generated for surrogate models to the target model can be improved by utilizing data augmentations [Xie et al. (2019)], exploiting gradients [Wu et al. (2020)], gradient aggregation [Liu et al. (2023)] or direction tuning [Yang et al. (2023)].

There are plenty of black-box attack methods known, for example, ZOO [Chen et al. (2017)] and NES [Ilyas et al. (2018)]. ZOO attack sequentially adds a small positive or negative perturbation to each pixel of the target image. It then queries the black-box model to estimate the gradient in the vicinity of the target image. NES attack works similarly. However, instead of changing pixel by pixel, a set of random images is generated, which are used to approximately estimate the gradients.

Current SOTA methods are Square Attack [Andriushchenko et al. (2020)], NP-Attack [Bai et al. (2020)], MCG [Yin et al. (2023)] and Bayesian attack [Li et al. (2023)]. Square Attack works differently. The attack selects an area of the image that is subject to attack and then gradually changes this area as the algorithm runs. And within the selected area, random pixels are selected that are changed. NP-Attack leverages a neural predictor model to guide the search for adversarial perturbations by predicting the model's output with fewer queries. MCG is a meta-learning-based black-box attack that leverages a meta-classifier to generalize adversarial attacks across different black-box models. The idea is to train a meta-classifier to guide the adversarial example generation. Bayesian attack enhances the transferability of adversarial examples by using a substitute model with Bayesian properties. The key idea is to make the substitute model more Bayesian through techniques like Monte Carlo dropout or stochastic weights, which results in better uncertainty estimation. This improved uncertainty estimation enhances the transferability of adversarial examples crafted on the substitute model to the target black-box model.

Note that Bayesian attack [Li et al. (2023)] belongs to the transfer-based category and implies access to part of the training data of the black-box model. In our work, we assume that an adversary has no access to the training data and, thus, we do not compare our approach against methods from the transfer-based category.

2.2 Knowledge Distillation and Adversarial Robustness

Knowledge distillation (KD) is a method to transfer the performance of a large teacher neural network to a smaller, lightweight student neural network [Hinton (2015)]. Given a teacher model T, the framework is used to train a student network S by solving an optimization problem:

$$S = \arg\min_{S'} \mathbb{E}_{(x,y)\sim\mathcal{D}} \left[\alpha\mathcal{L}(S'(x), y) + (1-\alpha)\tau^2 KL(S'(x), T(x))\right], \tag{1}$$

where $\mathcal{D}$ is the distillation dataset, $\mathcal{L}$ is the classification loss function used to assess the performance of the student model, KL is the Kullback-Leibler divergence and α, τ are the scalar parameters. Knowledge distillation has been used in a large scope of problems, such as model compression [Sun et al. (2019); Wang et al. (2019); Li et al. (2020)], data privacy [Lyu & Chen (2020); Chourasia et al. (2022); Galichin et al. (2024); Pautov et al. (2024)], adapted for large language models [McDonald et al. (2024); Gu et al. (2024); Kang et al. (2024)] and diffusion models [Huang et al. (2024); Yao et al. (2024); Yin et al. (2024)].

It has recently been shown that knowledge distillation can be used to enhance the adversarial robustness of additive perturbations [Papernot et al. (2016); Kuang et al. (2024); Huang et al. (2023)]. In contrast to a large teacher model which can attain a satisfactory level of adversarial robustness, it is challenging to make a small student model both robust and similar to the teacher one in performance [Huang et al. (2023)]. To deal with this issue, adversarially robust distillation was proposed [Goldblum et al. (2020)]. This approach takes into account clean predictions [Goldblum et al. (2020)] or probability vectors [Zi et al. (2021)] of robust teacher model during the distillation procedure.

3 Problem Statement

In this section, we formally discuss a problem statement, introduce the notations used throughout the paper, and formulate the research question.

3.1 Adversarial Example for a Classification Neural Network

Suppose that $f : \mathbb{R}^d \to \Delta^K$ is the classification neural network that maps input object $x \in \mathbb{R}^d$ to the vector $f(x) \in \Delta^K$ of probabilities of K classes and

$$h(f, x) = \arg\max_{i\in[1,\ldots,K]} f(x)_i \tag{2}$$

is the associated classification rule. We begin by formally defining an adversarial example for the given classification neural network and the transferability of an adversarial example between the two networks.

Definition 3.1 (Adversarial Example)**.** Suppose that $x \in \mathbb{R}^d$ is the input object correctly assigned to class $y \in [1, \ldots, K]$ by the network f, namely, $h(f, x) = y$. Let $\delta > 0$ be a fixed constant. Then, the object $x' \in \mathbb{R}^d$: $\|x - x'\|_2 \leq \delta$ is the *untargeted* adversarial example for f at point x, if

$$h(f, x') \neq h(f, x). \tag{3}$$

If $h(f, x') = t$ for some predefined class index t, then x' is called *targeted* adversarial example.

Definition 3.2 (Transferable Adversarial Example)**.** Let x' be the adversarial example computed for the network f at point x and let $g : \mathbb{R}^d \to \Delta^K$ be the separate network. Then, x' is transferable from f to g, if

$$\begin{cases} h(f, x) = h(g, x), \\ h(f, x') = h(g, x'). \end{cases} \tag{4}$$

3.2 Knowledge Distillation of a Black-Box Model

In this paper, we focus on using knowledge distillation [Hinton (2015)] to construct adversarial perturbations for the given classification model deployed in the black-box setting. Namely, let $T : \mathbb{R}^d \to \Delta^K$ be the black-box teacher model trained on an unknown dataset $\mathcal{D}(T)$ and $S : \mathbb{R}^d \to \Delta^K$ be the white-box student model, possibly of a different architecture, and let $\mathcal{D}(S)$ be its training dataset. To approximate the teacher model, we apply soft-label knowledge distillation, which is done in two steps. Firstly, the teacher model is used to collect the training dataset for the student model. In our setting, we use a hold out dataset $\mathcal{D}_h = \{(x_i, y_i)\}_{i=1}^m$ to construct $\mathcal{D}(S)$:

$$\mathcal{D}(S) = \{(x_i, T(x_i))\}_{i=1}^m, \tag{5}$$

where $x_i \in \mathcal{D}_h$ and $T(x_i) \in \Delta^K$. Then, the student network S is trained on the dataset $\mathcal{D}(S)$ by minimizing an empirical risk

$$L(S, \mathcal{D}(S)) = \frac{1}{m} \sum_{(x_i, y_i) \in \mathcal{D}(S)} l(S, x_i, y_i), \tag{6}$$

where $l(S, x, y) = -\log(S(x)_y)$ is the cross-entropy loss function.

When the student model is trained, we ask the following research question. Given $x \in \mathbb{R}^d : h(S, x) = h(T, x)$ and $\delta > 0$ from the definition 3.1, is it possible to compute an adversarial example for the model S at point x which is *provably* transferable to T? In the next section, we answer this question and propose a knowledge distillation-based adversarial attack with transferability guarantees.

4 Methodology

In this section, we describe the proposed approach to generate adversarial examples for the black-box teacher model via knowledge distillation. In the last subsection, we prove that, under several assumptions, our approach generates an adversarial example that is transferable to the teacher model within the finite number of iterations.

4.1 Model Mimic Attack: Student Follows its Teacher

To perform an adversarial attack on the black-box teacher model T, we first apply soft-label knowledge distillation and obtain the white-box student model S. The training dataset for the student model is constructed by querying the teacher model and collecting its predictions for the points from the hold-out dataset $\mathcal{D}_h$, possibly disjoint from the teacher's training dataset ($\mathcal{D}(T) : \mathcal{D}_h \cap \mathcal{D}(T) = \emptyset$). In our setup, we use the test subset of the teacher's dataset as the hold-out dataset $\mathcal{D}_h$.

Recall that $\mathcal{D}(S) = \{(x_i, T(x_i))\}_{i=1}^m$, according to equation 5. Assuming that the student model has enough learning capability, we train it until it perfectly matches the teacher model on $\mathcal{D}(S)$, namely,

$$\begin{cases} h(S, x_i) = h(T, x_i) = y_i \\ \|S(x_i) - T(x_i)\|_\infty < \frac{\varepsilon}{4}, \end{cases} \tag{7}$$

for all $(x_i, y_i) \in \mathcal{D}(S)$, where $\varepsilon > 0$ is the predefined constant. In equation 7, the second condition reflects the ability of the student model to confidently mimic the teacher model on $\mathcal{D}(S)$.

4.2 Model Mimic Attack: Student Under Attack

In this subsection, we describe a procedure to generate a single adversarial example for the student model.

When the student model is trained, we perform the white-box adversarial attack on it. To do so, we use Projected Gradient Descent [PGD, Madry et al. (2018)]. Given input object $x \in \mathbb{R}^d$ of class $y \in [1, \dots, K]$ correctly predicted by both teacher and student models, PGD performs iterative gradient ascent to find an adversarial example x' within $U_\delta(x)$, the $\delta-$neighborhood of x. Namely, for all $t \in [1, \dots, M]$,

$$\begin{cases} x^{t+1} = \text{Proj}_{U_\delta(x)} \left[x^t + \alpha \operatorname{sign} \nabla_{x^t} L(S, x^t, y) \right], \\ x^1 = x, x' = x^M, \end{cases} \tag{8}$$

where $\alpha > 0$ is the value of a single optimization step, M is the maximum number of PGD iterations, $\text{Proj}_{U_\delta(x)}$ is the projection onto $U_\delta(x)$, defined as

$$U_\delta(x) = \{x' : \|x - x'\|_2 \le \delta\}, \tag{9}$$

and $L(S, x^t, y)$ is the loss function reflecting the error of the model S on the sample (x^t, y). In our setting, $L(S, x^t, y)$ is the cross-entropy loss.

When the adversarial example x' for the student model S is found, we verify if it transfers to the teacher model, namely, if $h(S, x') = h(T, x')$. Not that x' does not have to be a transferable adversarial example. If $h(S, x') \neq h(T, x')$, then we add x' to the training dataset $\mathcal{D}(S)$ of the student model and repeat both the training of S and adversarial attack on it.

Remark. To increase the computational efficiency of the attack, we generate not a single adversarial example x' for the student model, but a batch $\{x'_1, \ldots, x'_l\}$ of l adversarial examples. The pseudo-code of the proposed method is presented in the Algorithm 1. Note that we use a Projected Gradient Descent attack because of its simplicity; our approach is not limited to a specific type of white-box attack.

Algorithm 1 Model Mimic Attack

Require: Black-box teacher model T, input object x of class y, distance threshold δ, gradient step α, maximum number of PGD iterations M, maximum number of distillation iterations N, hold-out dataset $\mathcal{D}_h$, the number l of adversarial examples to generate for the student model S_i

Ensure: Set of student models $S_1, \ldots, S_N$, the set $AE(T)$ of adversarial examples for the teacher model T

$z \leftarrow (x, T(x))$ ▷ compute the logits of T at the target point
$\mathcal{D}(S) \leftarrow \{(x_i, T(x_i))\}_{i=1}^m$ ▷ compute the training set $\mathcal{D}(S)$ according to the equation 5
$\mathcal{D}(S_1) \leftarrow \mathcal{D}(S) \cup z$ ▷ initialize the training set for the first student model S_1
$AE(T) \leftarrow \emptyset$ ▷ initialize the set of adversarial examples for the teacher model T
for $i = 1$ to N **do**
  $S_i \leftarrow \texttt{train}(\mathcal{D}(S_i))$ ▷ train the student model S_i using $\mathcal{D}(S_i)$
  for $j = 1$ to l **do**
    $(x'_j, y'_j) \leftarrow PGD(\alpha, \delta, S_i, (x, y))$ ▷ compute an adversarial example for the student model S_i according to equation 8
    if $h(S_i, x'_j) = h(T, x'_j)$ **then** ▷ check if the adversarial example transfers from S_i to T
      $AE(T) \leftarrow AE(T) \cup \{(x'_j, y'_j)\}$ ▷ update the set of adversarial examples for the model T
    end if
    $\mathcal{D}(S_{i+1}) \leftarrow \mathcal{D}(S_i) \cup \{(x'_j, T(x'_j))\}$ ▷ update the training set for the model S_{i+1}
  end for
end for

4.3 Model Mimic Attack: Provably Transferable Adversarial Examples

It should be mentioned that, under several assumptions, the Algorithm 1 is *guaranteed* to find an adversarial example that is transferable from the student model to the teacher model within the finite number of iterations. Namely, let T be the teacher model and S_i be the student model on i'th iteration with the corresponding training dataset $\mathcal{D}(S_i)$. Let $x \in \mathbb{R}^d$ be the input object correctly assigned by the teacher model to class $y \in [1, \ldots, K]$, and $\delta > 0$ be the distance threshold. Suppose that for every $i \in \mathbb{Z}_+$, the learning capability conditions from the equation 7 hold. Then, the following theorem holds.

Theorem 4.1. If $f_i = S_i - T$ be the functions with the bounded gradient in $U_\delta(x)$ for every $i \in \mathbb{Z}_+$ and let

$$\beta = \sup_{f_i} \sup_{x' \in U_\delta(x)} \|\nabla f_i(x')\|_F. \tag{10}$$

Suppose that for every $i \in \mathbb{Z}_+$, Algorithm 1 yields an adversarial example for the model S_i within the $\delta-$neighborhood of x. Then, exists $N \in \mathbb{Z}_+$ such that Algorithm 1 on N'th iteration yields an adversarial example transferable from S_N to T.

Proof. Let $\{x'_i\}_{i=1}^\infty$ be the sequence of adversarial examples generated by Algorithm 1 such that $\|x'_i - x\|_2 \leq \delta$ and x'_i is the adversarial example for the model S_i. Then, the sequence $\{x'_i\}_{i=1}^\infty$ is bounded in $U_\delta(x)$ and, hence, there exists the subsequence $\{x'_{i_j}\}_{j=1}^\infty$ such that exists

$$\lim_{j \to \infty} x'_{i_j} = z \in U_\delta(x). \tag{11}$$

Without the loss of generality, assume that $z \neq x$ and let $\{x'_{i_j}\}_{j=1}^\infty = \{z_i\}_{i=1}^\infty$.

Then,

$$\begin{aligned} |\|f_{i+1}(x)\|_\infty - \|f_{i+1}(z_{i+1})\|_\infty| &\le \|f_{i+1}(x) - f_{i+1}(z_{i+1})\|_\infty \le & (12)\\ &\le \|f_{i+1}(x) - f_{i+1}(z_i)\|_\infty + \|f_{i+1}(z_i) - f_{i+1}(z_{i+1})\|_\infty \le \\ &\le \|f_{i+1}(x)\|_\infty + \|f_{i+1}(z_i)\|_\infty + \|f_{i+1}(z_i) - f_{i+1}(z_{i+1})\|_\infty \le & (13)\\ &\le \frac{\varepsilon}{4} + \frac{\varepsilon}{4} + \|f_{i+1}(z_i) - f_{i+1}(z_{i+1})\|_\infty, \end{aligned}$$

where the last inequality is due to conditions from equation 7.

According to the mean value theorem,

$$f_{i+1}(z_i) - f_{i+1}(z_{i+1}) = \nabla f_{i+1}(\tau_{i+1})^\top (z_i - z_{i+1}), \tag{14}$$

for some $\tau_{i+1} \in [z_i, z_{i+1}] \subset U_\delta(x)$.

Since $\lim_{i\to\infty} z_i = z$, then $\lim_{i\to\infty} \|z_i - z_{i+1}\|_F = 0$ and $\exists N \in \mathbb{Z}_+ : \|z_{N-1} - z_N\|_F < \frac{\varepsilon}{4\beta}$.

Then,

$$\begin{aligned} \|f_N(z_{N-1}) - f_N(z_N)\|_\infty \le \|f_N(z_{N-1}) - f_N(z_N)\|_F \le \|\nabla f_N(\tau_N)\|_F \|z_{N-1} - z_N\|_F < \\ < \frac{\varepsilon}{4}. \end{aligned} \tag{15}$$

Substituting equation 15 into equation 12, we get

$$|\|f_N(x)\|_\infty - \|f_N(z_N)\|_\infty| < \frac{3\varepsilon}{4}, \text{ yielding } \|f_N(z_N)\|_\infty < \|f_N(x)\|_\infty + \frac{3\varepsilon}{4} = \varepsilon. \tag{16}$$

By setting ε to be small enough, for example,

$$\varepsilon < \frac{p_1 - p_2}{2}, \text{ where } p_1, p_2 \text{ are the two largest components of } S_N(z_N), \tag{17}$$

we get $h(S_N, z_N) = h(T, z_N)$, what finalizes the proof.

□

5 Experiments

This section will describe the experiments and everything needed to reproduce them. In particular, a description of the datasets, a method for evaluating the experiments, a description of the methods we compare with, and the methodology for conducting the experiments.

5.1 Setup of Experiments

Datasets and Training. In our experiments, we use CIFAR-10 and CIFAR-100 [Krizhevsky et al. (2009)] as the training datasets for the teacher model. We use ResNet50 [He et al. (2016)] as the teacher model T, which was trained for 250 epochs to achieve high classification accuracy (namely, 82% for CIFAR-10 and 47% for CIFAR-100. To train the teacher model, we use the SGD optimizer with the learning rate of 0.1, the weight decay of 10^{-4}, and the momentum of 0.9.

MMAttack Setup. We use ResNet18 and SmallCNN as the white-box student models. The architecture of SmallCNN is presented in the Appendix. We conduct the PGD attack on the student models with the following parameters: the number of PGD steps is set to be $M = 10$, the gradient step is set to be $\alpha = 0.005$, the distance threshold is set to be $\delta = 0.05$. The detailed architecture of the Small CNN model is presented in the appendix A.

Methods for Comparison. In this section, we briefly list the set of methods we compare our approach against. We evaluate MMAttack against ZOO [Chen et al. (2017)], NES [Ilyas et al. (2018)] as the main competitors. Among the black-box attack methods based on a random search, we choose Square attack [Andriushchenko et al. (2020)] as the state-of-the-art in terms of an average number of queries to conduct an attack. In the group of methods using gradient estimation, NP-Attack [Bai et al. (2020)] is among the most efficient attacks. In the category of combined methods, we choose MCG [Yin et al. (2023)]. The hyperparameters that were used in the experiments with Methods for Comparison are described in detail in the appendix B.

Note that the MCG algorithm originally assumes the training on the data from a distribution that is close to the teaches model's one, which in general may not be known. Here, we highlight that our method does not have such a limitation.

Table 1: Comparison of black-box attack methods. We report the average number of queries (AQN) required to generate the first adversarial example for the black-box model. Here, δ denotes the value of the maximum possible distance from the target point in terms of l_∞ norm. (**Lit**) denotes the metric values taken from the literature.

$\mathcal{D}$	Attack	δ	AQN (↓)
CIFAR-10	ZOO [Chen et al. (2017)]	0.05	$\geq 3 \times 10^5$
	NES [Ilyas et al. (2018)]	0.1	3578
	Square [Andriushchenko et al. (2020)]	0.1	368
	NP-Attack [Bai et al. (2020)] (**Lit**)	0.05	500
	MCG [Yin et al. (2023)] (**Lit**)	0.1	130
	MMAttack resnet18 (**ours**)	0.05	530
	MMAttack SmallCNN (**ours**)	0.05	**32.8**
CIFAR-100	ZOO [Chen et al. (2017)]	0.05	$\geq 3 \times 10^5$
	NES [Ilyas et al. (2018)]	0.1	4884
	Square [Andriushchenko et al. (2020)]	0.1	193
	NP-Attack [Bai et al. (2020)]	0.05	325
	MCG [Yin et al. (2023)] (**Lit**)	0.1	48
	MMAttack resnet18 (**ours**)	0.05	407
	MMAttack SmallCNN (**ours**)	0.05	**24**

Evaluation Protocol. To illustrate the efficiency of the proposed approach, we report the Average Query Number (AQN) and demonstrate the trade-off between AQN and the Average Success Rate (ASR). AQN denotes the number of queries required to generate all the adversarial examples for the black-box model, averaged over all the examples. ASR measures the fraction of adversarial examples assigned to a different class in an untargeted attack setting or to the predefined other class in the targeted attack setting. For AQN, a lower value indicates better attack performance, while for ASR, a higher value indicates a better attack performance. Note that both metrics are calculated over successful adversarial attacks only. In this paper, the emphasis is made on minimizing the AQN.

5.2 Results of Experiments

In the experiments, ZOO, NES, and Square attack methods were executed 100 times with different random seeds, NP-Attack, MCG, MMA methods were executed 30 times.

Table 1 shows a comparison of existing SOTA methods and the MMAttack method proposed in this work with two different substitute model architectures on the CIFAR-10 and CIFAR-100 datasets. The best results are highlighted in bold. It can be seen that the MMAttack method with the substitute model SmallCNN outperforms the competitors in terms of the AQN metric. (Table data for the MCG method on the CIFAR-100 dataset was taken from [Yin et al. (2023)]. Table data for the MCG and NP-Attack methods for the CIFAR-10 dataset were taken from [Zheng et al. (2023)]).

Note that if the results of a method presented in the literature do not match the results obtained in our implementation, then the result with the smallest number of average queries is reported. In the tables, the results taken from the literature are marked as (**Lit**).

5.3 Ablation Study

Note that the success of our black-box attack crucially depends on the architecture of the white-box student model. On the one hand, the student model does not have to have many training parameters since it implies several retraining iterations. On the other hand, it has to have enough learning capacity to mimic the behavior of the black-box model in the vicinity of the target point. In Table 2, we report the AQN values for the different pairs of teacher and student models on the CIFAR-10 dataset. Together with the average number of queries, we report the size of the initial training dataset $\mathcal{D}(S_1)$ of the student model and the number of adversarial examples to generate for the student model, l. We found that the simpler the architecture of the student model, the fewer queries to the teacher model are required to conduct a successful attack.

Table 2: Impact of hyperparameters on the performance of the MMAttack.

Teacher model, T	Student model, S	Initial dataset size, $\lvert\mathcal{D}(S_1)\rvert$	l	AQN (↓)
ResNet101	ResNet34	800	400	4520
ResNet50	ResNet34	600	400	4160
ResNet101	ResNet18	600	300	1560
ResNet50	ResNet18	600	200	530
ResNet34	ResNet18	300	30	455
ResNet101	SmallCNN	10	10	37.7
ResNet50	SmallCNN	10	10	32.8
ResNet34	SmallCNN	10	10	34
ResNet50	SmallCNN	5	5	–

The initial set size, $|\mathcal{D}(S_1)|$, represents the number of random data points to be included in the initial training dataset of the white-box student model. It can be seen from Table 2, that the more complex the student model is, the larger this parameter should be. The same is true for the number of adversarial examples for the student model, l.

Note that there is no AQN value corresponding to $|\mathcal{D}(S_1)| = 5$ and $l = 5$. This is because the Algorithm 1 does not succeed in finding a single adversarial example for the black-box teacher model until it reaches the maximum iterations threshold.

It is also worth mentioning that Model Mimic Attack implies a certain trade-off between ASR and AQN metrics. At the start, when the size of the training dataset of the student model is relatively small and very few iterations of knowledge distillation are passed, the algorithm is less likely to find an adversarial example for the teacher model. In contrast, after more distillation iterations, the algorithm tends to find more transferable adversarial examples on each iteration. In tables 3 and 4, we show the trade-off between the ASR and AQN metrics from one distillation iteration to another: when the number of passed distillation iterations increases, so does the number of queries to the teacher model used to collect additional training samples for the student model by that iteration, QN_1. In contrast, the number of queries remaining to find an attack on the black-box model, QN_2, decreases (here, we fix the total number of queries to be $QN_1 + QN_2 = 200$).

However, if the goal is not to obtain the minimum value of the AQN metric, but to improve the trade-off between the ASR and AQN metrics, one could run several cycles of the algorithm to better study the behavior of the teacher model in the vicinity of the target point.

The choice of the white-box attack method plays an important role in finding the transferable adversarial example: on one hand, the more powerful the white-box attack is, the more frequently an adversarial example will be found for the student model; on the other hand, the faster the attack is, the more distillation iterations can be performed within a limited time. In this work, a projected gradient descent (PGD) attack with the l_∞ norm constraint is used, but the method is not limited to any specific type of white-box attack. It is possible to use variants of the white-box attack with l_2 or l_1 constraints, to conduct an attack in a targeted setting or use more complicated attack methods. In any case, MMAttack is expected to have similar properties. The optimal choice depends on the specific domain and the effectiveness of each white-box attack method on a given dataset.

6 LIMITATIONS

Note that the transferability guarantee from Theorem 4.1 is given for the soft-label distillation. It is worth mentioning that the Theorem can not be adapted to the hard-label distillation without significant changes. Instead, to provide the transferability guarantee in hard-label distillation, when the teacher model outputs the predicted class label only, one can estimate the *probability of transferability* of an adversarial example within the finite number of iterations, conditioned on the white-box attack. If the lower bound of this probability is separated from zero, one can estimate the expected number of distillation iterations required to yield the transferable adversarial example.

7 CONCLUSION AND FUTURE WORK

In this paper, we propose the Model Mimic Attack, the first framework to compute adversarial perturbations for a black-box neural network that is guaranteed to find an adversarial example for the latter. To conduct an attack, we

Table 3: Trade-off between ASR and AQN metrics for MMAttack, CIFAR-10 dataset. QN_1 represents the number of data points added to the training dataset of the student model by corresponding iteration; QN_2 represents the attack budget, or upper bound of the number of queries to find an attack on the black-box model.

Iteration number	QN_1	QN_2	Number of generated attacks	ASR (↑)	AQN (↓)
1	10	190	121.67	0.66	2.36
2	20	180	112.05	0.68	2.50
3	30	170	105.57	0.67	2.67
4	40	160	99.38	0.67	2.86
5	50	150	93.10	0.67	3.03
6	60	140	87.48	0.67	3.24
7	70	130	81.00	0.67	3.49
8	80	120	75.14	0.68	3.74
9	90	110	69.19	0.68	4.05
10	100	100	62.81	0.68	4.43
11	110	90	56.43	0.70	4.83
12	120	80	50.43	0.69	5.50
13	130	70	43.62	0.69	6.35
14	140	60	37.43	0.68	7.42
15	150	50	30.95	0.67	9.13
16	160	40	25.00	0.68	11.16
17	170	30	19.05	0.68	14.62
18	180	20	12.62	0.74	20.39
19	190	10	6.86	0.72	38.38

Table 4: Trade-off between ASR and AQN metrics for MMAttack, CIFAR-100 dataset. QN_1 represents the number of data points added to the training dataset of the student model by corresponding iteration; QN_2 represents the attack budget, or upper bound of the number of queries to find an attack on the black-box model.

Iteration number	QN_1	QN_2	Number of generated attacks	ASR (↑)	AQN (↓)
1	10	190	163.17	0.84	1.38
2	20	180	153.17	0.85	1.46
3	30	170	144.90	0.85	1.54
4	40	160	135.86	0.85	1.64
5	50	150	127.34	0.85	1.75
6	60	140	119.14	0.85	1.87
7	70	130	110.48	0.85	2.02
8	80	120	102.03	0.85	2.19
9	90	110	93.59	0.85	2.39
10	100	100	85.52	0.85	2.63
11	110	90	76.97	0.84	2.92
12	120	80	68.48	0.85	3.25
13	130	70	59.97	0.86	3.69
14	140	60	51.62	0.86	4.30
15	150	50	42.97	0.86	5.14
16	160	40	34.72	0.86	6.37
17	170	30	26.41	0.85	8.42
18	180	20	17.66	0.88	12.28
19	190	10	8.72	0.90	24.08

apply knowledge distillation to obtain the student model, which is essentially the functional copy of the black-box teacher network. Then, we perform the white-box adversarial attack on the student model and theoretically show that, under several assumptions, the attack transfers to the teacher model. We demonstrate experimentally that a successful adversarial attack can be found within a small number of queries to the target model, making the approach feasible for practical applications. Possible directions for future work include an extension of the transferability guarantees to the hard-label distillation and adaptation of the proposed method for other domains, in particular, for attacking large language models.

REFERENCES

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pp. 484–501. Springer, 2020.

Yang Bai, Yuyuan Zeng, Yong Jiang, Yisen Wang, Shu-Tao Xia, and Weiwei Guo. Improving query efficiency of black-box adversarial attack. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pp. 101–116. Springer, 2020.

Arjun Nitin Bhagoji, Warren He, Bo Li, and Dawn Song. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 154–169, 2018.

Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy (SP)*, pp. 39–57, 2016.

Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 IEEE Symposium on Security and Privacy (SP)*, pp. 1277–1294. IEEE, 2020.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. 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*, pp. 15–26, 2017.

Rishav Chourasia, Batnyam Enkhtaivan, Kunihiro Ito, Junki Mori, Isamu Teranishi, and Hikaru Tsuchida. Knowledge cross-distillation for membership privacy. *Proceedings on Privacy Enhancing Technologies*, 2022.

Andrey V Galichin, Mikhail Pautov, Alexey Zhavoronkin, Oleg Y Rogov, and Ivan Oseledets. Glira: Black-box membership inference attack via knowledge distillation. *arXiv preprint arXiv:2405.07562*, 2024.

Micah Goldblum, Liam Fowl, Soheil Feizi, and Tom Goldstein. Adversarially robust distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 3996–4003, 2020.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *CoRR*, abs/1412.6572, 2014.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Minillm: Knowledge distillation of large language models. In *The Twelfth International Conference on Learning Representations*, 2024.

Martin Gubri, Maxime Cordy, Mike Papadakis, Yves Le Traon, and Koushik Sen. Lgv: Boosting adversarial example transferability from large geometric vicinity. In *European Conference on Computer Vision*, pp. 603–618. Springer, 2022.

Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. Simple black-box adversarial attacks. In *International Conference on Machine Learning*, pp. 2484–2493. PMLR, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.

G Hinton. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Bo Huang, Mingyang Chen, Yi Wang, Junda Lu, Minhao Cheng, and Wei Wang. Boosting accuracy and robustness of student models via adaptive adversarial distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24668–24677, 2023.

Tao Huang, Yuan Zhang, Mingkai Zheng, Shan You, Fei Wang, Chen Qian, and Chang Xu. Knowledge diffusion for distillation. *Advances in Neural Information Processing Systems*, 36, 2024.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pp. 2137–2146. PMLR, 2018.

Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. In *International Conference on Learning Representations*, 2019.

Minki Kang, Seanie Lee, Jinheon Baek, Kenji Kawaguchi, and Sung Ju Hwang. Knowledge-augmented reasoning distillation for small language models in knowledge-intensive tasks. *Advances in Neural Information Processing Systems*, 36, 2024.

Byungjoo Kim, Suyoung Lee, Seanie Lee, Sooel Son, and Sung Ju Hwang. Margin-based neural network watermarking. In *International Conference on Machine Learning*, pp. 16696–16711. PMLR, 2023.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

Huafeng Kuang, Hong Liu, YongJian Wu, Shin'ichi Satoh, and Rongrong Ji. Improving adversarial robustness via information bottleneck distillation. *Advances in Neural Information Processing Systems*, 36, 2024.

Qizhang Li, Yiwen Guo, Wangmeng Zuo, and Hao Chen. Making substitute models more bayesian can enhance transferability of adversarial examples. *arXiv preprint arXiv:2302.05086*, 2023.

Tianhong Li, Jianguo Li, Zhuang Liu, and Changshui Zhang. Few sample knowledge distillation for efficient network compression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 14639–14647, 2020.

Xuannan Liu, Yaoyao Zhong, Yuhang Zhang, Lixiong Qin, and Weihong Deng. Enhancing generalization of universal adversarial perturbation through gradient aggregation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4435–4444, 2023.

Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations*, 2022.

Nils Lukas, Yuxuan Zhang, and Florian Kerschbaum. Deep neural network fingerprinting by conferrable adversarial examples. *arXiv preprint arXiv:1912.00888*, 2019.

Lingjuan Lyu and Chi-Hua Chen. Differentially private knowledge distillation for mobile analytics. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1809–1812, 2020.

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.

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

Daniel McDonald, Rachael Papadopoulos, and Leslie Benningfield. Reducing llm hallucination using knowledge distillation: A case study with mistral large and mmlu benchmark. *Authorea Preprints*, 2024.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 582–597. IEEE, 2016.

Mikhail Pautov, Nikita Bogdanov, Stanislav Pyatkin, Oleg Rogov, and Ivan Oseledets. Probabilistically robust watermarking of neural networks. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI-24*, pp. 4778–4787, 2024.

Yunxiao Qin, Yuanhao Xiong, Jinfeng Yi, and Cho-Jui Hsieh. Training meta-surrogate model for transferable adversarial attack. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 9516–9524, 2023.

Siqi Sun, Yu Cheng, Zhe Gan, and Jingjing Liu. Patient knowledge distillation for bert model compression. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4323–4332, 2019.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *2nd International Conference on Learning Representations*, 2014.

Quoc Viet Vo, Ehsan Abbasnejad, and Damith Ranasinghe. Brusleattack: A query-efficient score-based black-box sparse adversarial attack. In *The Twelfth International Conference on Learning Representations*, 2024.

Ji Wang, Weidong Bao, Lichao Sun, Xiaomin Zhu, Bokai Cao, and S Yu Philip. Private model compression via knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 1190–1197, 2019.

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*, 2020.

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.

Xiangyuan Yang, Jie Lin, Hanlin Zhang, Xinyu Yang, and Peng Zhao. Improving the transferability of adversarial examples via direction tuning. 2023.

Xufeng Yao, Fanbin Lu, Yuechen Zhang, Xinyun Zhang, Wenqian Zhao, and Bei Yu. Progressively knowledge distillation via re-parameterizing diffusion reverse process. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 16425–16432, 2024.

Fei Yin, Yong Zhang, Baoyuan Wu, Yan Feng, Jingyi Zhang, Yanbo Fan, and Yujiu Yang. Generalizable black-box adversarial attack with meta learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(3): 1804–1818, 2023.

Tianwei Yin, Michaël Gharbi, Richard Zhang, Eli Shechtman, Fredo Durand, William T Freeman, and Taesung Park. One-step diffusion with distribution matching distillation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6613–6623, 2024.

Xiaoyong Yuan, Leah Ding, Lan Zhang, Xiaolin Li, and Dapeng Oliver Wu. Es attack: Model stealing against deep neural networks without data hurdles. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(5): 1258–1270, 2022.

Meixi Zheng, Xuanchen Yan, Zihao Zhu, Hongrui Chen, and Baoyuan Wu. Blackboxbench: A comprehensive benchmark of black-box adversarial attacks. *arXiv preprint arXiv:2312.16979*, 2023.

Bojia Zi, Shihao Zhao, Xingjun Ma, and Yu-Gang Jiang. Revisiting adversarial robustness distillation: Robust soft labels make student better. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 16443–16452, 2021.

A Appendix: Architecture of SmallCNN

```
SmallCNN(
  (features): Sequential(
    (0): Conv2d(3, 64, kernel_size=(3, 3), stride=(1, 1),
         padding=(1, 1))
    (1): ReLU(inplace)
    (2): MaxPool2d(kernel_size=2, stride=2, padding=0,
         dilation=1, ceil_mode=False)
    (3): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1),
         padding=(1, 1))
    (4): ReLU(inplace)
    (5): MaxPool2d(kernel_size=2, stride=2, padding=0,
         dilation=1, ceil_mode=False)
    (6): Conv2d(128, 256, kernel_size=(3, 3), stride=(1, 1),
         padding=(1, 1))
    (7): ReLU(inplace)
    (8): MaxPool2d(kernel_size=2, stride=2, padding=0,
         dilation=1, ceil_mode=False)
  )
  (classifier): Sequential(
    (0): Linear(in_features=4096, out_features=512, bias=True)
```

```
    (1): ReLU(inplace)
    (2): Linear(in_features=512, out_features=10 or 100,
         bias=True)
  )
)
```

B APPENDIX: HYPERPARAMETERS OF THE COMPARED ATTACK METHODS

Table 5: Hyperparameters of the compared attack methods

Method	**Hyperparameters**
ZOO attack	$\epsilon = 0.05$ num_iterations = 5000 learning_rate = 0.01
NES attack	$\epsilon = 0.1$ num_samples = 50 num_iterations = 300 $\sigma = 0.01$ $\alpha = 0.03$
Square attack	$\epsilon = 0.1$ num_queries = 5000 $p\_init = 0.8$
NP attack	$\epsilon = 0.05$ num_iterations = 1000 learning_rate = 0.01
MCG	down_sample_x = 1 down_sample_y = 1 finetune_grow = True finetune_reload = True finetune_perturbation = True