ABSTRACT

ZHANG, ZEKUN. Learning Ordered Top- k Adversarial Attacks via Adversarial Distillation. (Under the direction of Tianfu Wu.)

Deep Neural Networks (DNNs) are vulnerable to adversarial attacks, especially white-box targeted attacks. This paper studies the problem of how aggressive white-box targeted attacks can be to go beyond widely used Top-1 attacks. We propose to learn **ordered Top- k attacks** ($k \geq 1$), which enforce the Top- k predicted labels of an adversarial example to be the k (randomly) selected and ordered labels (the ground-truth label is exclusive). Two methods are presented. First, we extend the vanilla Carlini-Wagner (C&W) method and use it as a strong baseline. Second, we present an **Adversarial Distillation (AD)** framework consisting of two components: (i) Computing an adversarial probability distribution for a given ordered Top- k targeted labels. (ii) Learning adversarial examples by minimizing the Kullback-Leibler (KL) divergence between the adversarial distribution and the predicted distribution, together with the perturbation energy penalty. In computing adversarial distributions, we explore how to leverage label semantic similarities, leading to **knowledge-oriented attacks**. In experiments, we test Top- k ($k = 1, 2, 5, 10$) attacks in the ImageNet-1000 val dataset using three representative DNNs trained with the clean ImageNet-1000 train dataset, ResNet-50 [He16], DenseNet-121 [Hua17] and AOGNet-12M [Li19]. Overall, the proposed AD approach obtains the best results, especially by a large margin when computation budget is limited. It reduces the perturbation energy consistently with the same attack success rate on all the four k 's, and improve the attack success rate by a large margin against the modified C&W method for $k = 10$.

© Copyright 2020 by Zekun Zhang

All Rights Reserved

Learning Ordered Top- k Adversarial Attacks via Adversarial Distillation

by
Zekun Zhang

A thesis submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Electrical Engineering

Raleigh, North Carolina

2020

APPROVED BY:

Huaiyu Dai

Edgar Lobaton

Tianfu Wu
Chair of Advisory Committee

DEDICATION

To my parents, Yu Zhang and Juan Cheng, we love and support each other no matter what
To my advisor, Dr. Tianfu Wu, he is a kind friend, a great advisor and an excellent researcher

To my cat, Megatron, I hope you will live a happy life

To everyone I met during my time at NCSU, I love you

To myself

BIOGRAPHY

In June 2017, the author received his B.S. degree of Science in Electronic Science and Technology from Beijing Institute of Technology. In August of the same year, he enrolled in the graduate program in the Department of Electrical and Computer Engineering, at North Carolina State University. There, he worked as a teaching and research assistant supervised by Dr. Tianfu Wu. Also, he pursued his Master Thesis under the guidance of Dr. Wu. His research focuses on developing an efficient and explainable adversarial attack method for existing deep learning architectures.

ACKNOWLEDGEMENTS

First of all, I would like to thank my advisor, Dr. Tianfu Wu. He is a great advisor, and his patience and valuable advice are of great help to my research work. He is also a kind friend who cares about and helps my life. Also, I would like to extend my thanks to Dr. Huaiyu Dai and Dr. Edgar Lobaton for being my committee members. Finally, I would like to thank my group-mates who accompanied me during my graduate school journey: Wei Sun, Liang Dong, Xilai Li and Nan Xue.

TABLE OF CONTENTS

LIST OF TABLES	vi
LIST OF FIGURES	vii
Chapter 1 INTRODUCTION	1
1.1 Adversarial Attacks	1
1.1.1 Term Definitions	2
1.1.2 Adversarial Attacks in Deep Learning	3
1.2 Distillation	3
Chapter 2 Literature Review and Contribution Summary	6
2.1 Box-constrained L-BFGS	7
2.2 Fast Gradient Sign Method (FGSM)	8
2.3 Basic Iterative Method (BIM)	9
2.4 Attacks on ℓ_p constraint	10
2.5 Momentum Iterative Fast Gradient Sign Method (MIFGSM)	10
2.6 Carlini Wagner Attack (C&W)	11
2.7 The proposed work and Contributions	12
Chapter 3 Ordered Top-k Attack and Adversarial Distillation	14
3.1 Motivation	14
3.2 Method Overview	15
3.3 Problem Formulation	17
3.3.1 Background	18
3.3.2 The Proposed Ordered Top- k Attack Setting	19
3.3.3 Learning Ordered Top- k Attacks	19
Chapter 4 Experiments	21
4.1 Setup	21
4.1.1 Data	21
4.1.2 Settings	22
4.2 Results	24
4.2.1 Results for ResNet-50	24
4.2.2 Results for DenseNet-121 and AOGNet-12M	24
4.2.3 Are all Top- k targets equally difficult to attack?	24
Chapter 5 Conclusions	28
BIBLIOGRAPHY	30

LIST OF TABLES

Table 3.1	Results showing where the ground-truth (GT) labels are in the prediction of learned adversarial examples for different attack methods. The test is performed in ImageNet-1000 val dataset using ResNet-50 [He16] pretrained with clean images. The subscripts of methods indicate the computing budgets used. See Chapter 4 for experimental settings.	15
Table 4.1	Results and comparisons under the ordered Top- k targeted attack protocol using randomly selected and ordered k targets (GT exclusive) in ImageNet using ResNet-50 . For Top-1 attacks, we also compare with three state-of-the-art untargeted attack methods, FGSM [Goo15], PGD [Mad18] and MIFGSM [Don18]. 10 iterations are used for both PGD and MIFGSM.	22
Table 4.2	Results and comparisons using DenseNet-121 [Hua17] under the ordered Top- k targeted attack protocol using randomly selected and ordered 10 targets (GT exclusive). For Top-1 attacks, we also compare with three state-of-the-art untargeted attack methods, FGSM [Goo15], PGD [Mad18] and MIFGSM [Don18]. 10 iterations are used for both PGD and MIFGSM.	25
Table 4.3	Results and comparisons under the ordered Top- k targeted attack protocol using randomly selected and ordered 10 targets (GT exclusive) in ImageNet using AOGNet-12M	26
Table 4.4	Results of ordered Top-5 targeted attacks for ResNet-50 with targets being selected based on (Top) label similarity, which uses 5 most-like labels and 5 least-like labels as targets respectively, and (Bottom) prediction score of clean image, which uses 5 highest-score labels and 5-lowest score labels. In both cases, GT labels are exclusive.	27

LIST OF FIGURES

Figure 1.1	Simple example of adversarial attacks on image classification task, which is shown in paper [Goo15]. By adding a small perturbation to the clean panda image, the final adversarial example is still a panda image in human eyes, but the targeted model will recognize it as a gibbon with an extremely high confidence.	2
Figure 1.2	This figure is shown in [Su19], by applying the differential evolution, it searches pixels and the performance of the confidence decreasing when random changes are made on the pixels. Eventually, it will lower the confidence of correct class and successfully fool the targeted model.	4
Figure 1.3	Network Distillation [Hin15; Buc06] shows that the knowledge from the teacher model, which is a well pre-trained DNN model, can be transmitted to the student model, by adding the conditional distribution of teacher model to the loss function of student model.	5
Figure 2.1	Adversarial examples generated by [Sze14] on AlexNet [Kri12]. Left are the clean images which are classified correctly by the model. Center are the perturbation added to the clean image, and they are magnified 10× for visualization. Right are the final adversarial examples used to fool the model. . .	8
Figure 2.2	Adversarial examples are generated by one DNN classifier. Then those examples are printed in the real-world and fool another Tensorflow mobile app classifier successfully. It's even a real-world black-box attack.	10
Figure 3.1	Illustration of a Top- k attack on a 'Dogsled' image. The Top- k predicted labels of the target model on the adversarial example will be 10 ordered labels assigned before attack from 'Lesser Panda' to 'Beagle'.	15
Figure 3.2	Comparisons of Top- k attack success rates (ASR, higher is better) between the modified C&W method (CW^k) and the proposed Adversarial Distillation (AD) method ($k = 1, 2, 5, 10$). The thickness of plotted lines represents the reverse ℓ_2 energy of the learned perturbation (thicker is better). 9×30 and 9×1000 represent the computing budgets in learning (see Chapter 4 for experimental settings).	16
Figure 3.3	Learned adversarial examples of ordered Top-10 adversarial attacks for ResNet-50 [He16] pretrained with clean images. The proposed AD method has smaller perturbation energies and "cleaner" (lower-entropy) prediction distributions than the proposed modified C&W method (CW^k). Note that for Top-10 attacks, the 9×30 search scheme does not work (see Table. 4.1). See text for detail.	17
Figure 4.1	Learned adversarial examples of ordered Top-5 (top) and vanilla Top-1 (bottom) attacks for ResNet-50 [He16] pretrained with clean images. The proposed AD method has smaller perturbation energies and "cleaner" (lower-entropy) prediction distributions than both the modified CW^k method in Top-5 attacks and the vanilla C&W method in the Top-1 attacks.	23

CHAPTER

1

INTRODUCTION

Despite the recent dramatic progress, deep neural networks (DNNs) [LeC98; Kri12; He16; Sze16] trained for visual recognition tasks (*e.g.*, image classification) can be easily fooled by so-called **adversarial attacks** which utilize visually imperceptible, carefully-crafted perturbations to cause networks to misclassify inputs in arbitrarily chosen ways in the close set of labels used in training [Ngu15; Sze14; Ath17; Car17]. The existence of adversarial attacks hinders the deployment of DNNs-based visual recognition systems in a wide range of applications such as autonomous driving and smart medical diagnosis in the long-run.

1.1 Adversarial Attacks

Adversarial attacks have quickly become an important problem to modern deep learning networks, since Szegedy [Sze14] first discovered the weakness of DNN in the field of image classification. They showed that by only adding a small perturbation, which is imperceptible to human eyes, to the clean image, the well-trained with high prediction accuracy DNN can be easily fooled. Figure 1.1 shows an example about how adversarial attacks work on image classification models. After that, many great researchers put their attention on adversarial attacks and proposed a lot of significant work. Furthermore, the interest on this topic quickly spread to the other areas of deep learning research. People started to study adversarial attacks in other field and with different limitations.

Figure 1.1 Simple example of adversarial attacks on image classification task, which is shown in paper [Goo15]. By adding a small perturbation to the clean panda image, the final adversarial example is still a panda image in human eyes, but the targeted model will recognize it as a gibbon with an extremely high confidence.

1.1.1 Term Definitions

Before further introduction of adversarial attacks, let’s first talk about some common terms widely used in adversarial attack field.

- **Adversarial Example/Image** is the final fake image which will be used to fool the model. And it is obtained by adding the perturbation on the clean image.
- **Adversarial Perturbation** is the noise added to the clean image.
- **Visually Imperceptible** means human eyes will not be able to tell the difference between clean image and adversarial example.
- **Untargeted Attack** fools the model to predict a wrong label, which could be any label but not the ground truth one. The final predicted label of an adversarial example generated by untargeted attack is not important as long as it’s not the correct one.
- **Targeted Attack** is the opposite of **Untargeted Attack**, the final predicted label should be exactly the label assigned before attack.
- **Targeted Model** is the model going to be fooled/attacked.
- **White-Box Attack** gives the attacker the full information about the targeted model, which means the attacker will know everything about the targeted model, including it’s architecture, parameters, training processing and training data.
- **Black-Box Attack** is the opposite of **White-Box Attack** which will not give the attacker any information about the targeted model. The attack should generate the attack examples without

any knowledge to the targeted model. Sometimes, an attacker is allowed only limited access to the model, such as 100 input times for each image. Moreover, in some cases, the attacker will be given some limited information, like training processing or training data. It's also named as **Gray-Box** or **Semi-Black-Box** attack.

- **Transferability** means an Adversarial example generated to fool one specific targeted model can also fool another model. Normally, the **Transferability** is only considered among models trained on the same or similar data.
- **Attack Success Rate** is the ratio of adversarial examples that successfully deceive the targeted model in all test data. It's also known as **ASR**.

1.1.2 Adversarial Attacks in Deep Learning

The growing ubiquity of DNNs in advanced machine learning and AI systems dramatically increases their capabilities, but also increases the potential for new vulnerabilities to attacks. Adversarial attacks are applied to nearly every field of deep learning. There are attacks on segmentation and detection [Xie17; Arn18; Son18], attacks on face recognition [Sha16; Don19a], attacks on audio [Qin19], attacks on reinforcement learning [Lin17; Gle20], attacks in real world [Kur17a; Son18] and other attacks [Kos18; Ebr18]. There are also attacks focusing on perceptible perturbation like [Bro17; Su19], by adding a small patch to the image or significantly changing several pixels, those attacks can fool the model successfully.

This situation has become critical as many powerful approaches have been developed where imperceptible perturbations to DNN inputs could deceive a well-trained DNN, significantly altering its prediction. Such results have initiated a rapidly proliferating field of research characterized by ever more complex attacks [Pap17b; Liu17; Xie19b; Don19b; Xu18; Che17; Bre18]. Those attacks focus on generating much stronger adversarial examples, which could have a higher ASR, a smaller perturbation, a better transferability, consuming much less computation budget or can do a black-box attack. These more efficient attacks prove increasingly strong against defensive countermeasures [Goo15; Kan18; Mad18; Xie19a].

Please refer to [Akh18; Cha18] for a comprehensive survey of attack methods in computer vision. We will review some related work that motivate our work in chapter 2.

1.2 Distillation

The central idea of our proposed AD method is built on distillation. Network distillation [Buc06; Hin15] is a powerful training scheme proposed to train a new, usually lightweight model (a.k.a., the student) to mimic another already trained model (a.k.a. the teacher). It takes a functional viewpoint of the knowledge learned by the teacher as the conditional distribution it produces over outputs given an input. The teacher teaches the student to keep up or emulate by adding some regularization terms to the loss in order to encourage the two models to be similar directly based on the distilled

Figure 1.2 This figure is shown in [Su19], by applying the differential evolution, it searches pixels and the performance of the confidence decreasing when random changes are made on the pixels. Eventually, it will lower the confidence of correct class and successfully fool the targeted model.

Figure 1.3 Network Distillation [Hin15; Buc06] shows that the knowledge from the teacher model, which is a well pre-trained DNN model, can be transmitted to the student model, by adding the conditional distribution of teacher model to the loss function of student model.

knowledge, replacing the training labels. Figure 1.3 shows the structure of Network Distillation, which motivates us to generate our own conditional distribution containing some knowledge that can help our AD method learn more powerful adversarial examples. Label smoothing [Sze15] can be treated as a simple hand-crafted knowledge to help improve model performance.

Distillation has been exploited to develop defense models [Pap16; Pap17a] to improve model robustness. Our proposed adversarial distillation method utilizes the distillation idea in an opposite direction, leveraging label semantic knowledge for learning ordered Top- k attacks and improving attack robustness.

CHAPTER

2

LITERATURE REVIEW AND CONTRIBUTION SUMMARY

For image classification tasks using DNNs, the discovery of the existence of visually-imperceptible adversarial attacks [Sze14] was a big shock in developing DNNs. Among all those attacks in image classification field, white-box attacks provide a powerful way of evaluating model brittleness. Assuming full access to DNNs pretrained with clean images, white-box targeted attacks are powerful ways of investigating the brittleness of DNNs and their sensitivity to non-robust yet well-generalizing features in the data, and of exploiting adversarial examples as useful features [Ily19]. In this paper, we are interested in studying visually-imperceptible targeted attacks under the white-box and limited computing budget setting in image classification tasks.

In a plain and loose explanation, DNNs are universal function approximator [Hor89] and capable of even fitting random labels [Zha16] in large scale classification tasks as ImageNet-1000 [Rus15]. Thus, adversarial attacks are generally learnable provided proper objective functions are given, especially when DNNs are trained with fully differentiable back-propagation. Many white-box attack methods focus on norm-ball constrained objective functions [Sze14; Kur17a; Car17; Don18; Ron19]. The C&W method investigates 7 different loss functions. The best performing loss function found by the C&W method has been applied in many attack methods and achieved strong results [Che17; Mad18; Che18]. By introducing momentum in the MIFGSM method [Don18] and the ℓ_p gradient projection in the PGD method [Mad18], both methods achieve better performance in generating adversarial examples. In the meanwhile, some other attack methods such as the StrAttack [Xu18] also investigate different loss functions for better interpretability of attacks.

In this chapter, we will take a review of some of these white-box attacks on image classifica-

tion introduced above. Note that most of the approaches introduced this chapter are the basis of adversarial attacks, not only in classification tasks, but also in other deep learning task and even real-world attacks.

2.1 Box-constrained L-BFGS

It is generally believed that [Sze14] is the first pioneering work in adversarial attacks. In this paper, Szegedy demonstrated that an adversarial example can easily fool a DNN by adding a small enough perturbation to the clean image. They formulated the problem as solving a box-constrained optimization problem:

$$\begin{aligned} & \text{minimize} \quad \|\delta\|_2, \\ & \text{subject to} \quad \arg \max_c f(x + \delta)_c = t, \\ & \quad \quad \quad x + \delta \in [0, 1]^n, \end{aligned} \tag{2.1}$$

where f is the classifier which maps the images to the label. For a given image x , $y = \arg \max_c f(x)_c$, y is the ground-truth label of image x . The given target label $t \neq y$. By minimizing $\|\delta\|_2$, the adversarial example $x + \delta$ would be the closest image to x but classied as target label t . However solving this problem becomes a hard problem, so that the authors proposed a box-constrained L-BFGS approximate solution. This is done by performing line-search to find the minimum $c > 0$ for which the minimizer δ of the following problem satisfies $t = \arg \max_c f(x + \delta)_c$:

$$\begin{aligned} & \text{minimize} \quad c\|\delta\| + L(x + \delta, t), \\ & \text{subject to} \quad x + \delta \in [0, 1]^n, \end{aligned} \tag{2.2}$$

where $L(\cdot)$ is the loss function of the classifier. And δ is known as the perturbation and will be directly added to the image to obtain the adversarial examples. Fig. 2.1 shows the perturbation generated by this method could be small enough in human eyes, and the final adversarial examples are similar to the clean images to human. It's also discovered by Szegedy that the adversarial examples generated to attack one DNN could also be able to attack another DNN, though the Attack Success Rate will drop. This property of adversarial examples is also known as transferability. In general, the discovery of this box-constrained L-BFGS attack shows the world how vulnerable DNNs can be, which has attracted the interest of researchers in deep learning field. And inspired by this work [Sze14], many further research on adversarial attack and defense have been proposed.

Figure 2.1 Adversarial examples generated by [Sze14] on AlexNet [Kri12]. Left are the clean images which are classified correctly by the model. Center are the perturbation added to the clean image, and they are magnified 10× for visualization. Right are the final adversarial examples used to fool the model.

2.2 Fast Gradient Sign Method (FGSM)

Inspired by [Sze14] and in order to efficiently generate adversarial examples, Goodfellow [Goo15] proposed a new and effective method by solving this problem:

$$\delta = \epsilon \cdot \text{sign}(\nabla_x L(\theta, x, y)), \quad (2.3)$$

where θ is the parameters of the targeted DNN model, x is the input image to the model, y is the ground-truth label. And $L(\cdot)$ is the cost function used to train the target model, $\nabla L(\theta, x, y)$ is the gradient of the cost function w.r.t input x . $\text{sign}(\cdot)$ is the sign function and ϵ is a small coefficient used to limit the size of the perturbation so that they are imperceptible to human eyes. The final adversarial example is $x' = (x + \delta)$. This method of generating adversarial attacks is named as Fast Gradient Sign Method, also known as FGSM. Fig. 1.1 shows an adversarial example generated by FGSM.

Note that the original formulation Eqn. 2.3 only focus on simply trying to increase the cost of the correct class, it is set up for untargeted attack. However, with a small modification, [Kur17b] shows that equation Eqn. 2.3 can be used to generate targeted attack perturbation:

$$\delta = -\epsilon \cdot \text{sign}(\nabla_x L(\theta, x, t)), \quad (2.4)$$

where t is the assigned target label. FGSM doesn't require an iterative procedure to generate adversarial examples, it's an 'one-step' method, and thus it is much faster than other methods. Meanwhile, FGSM still fools a variety of models into misclassification. This small computation budget of FGSM also improves the efficiency of adversarial training. Note that the applying $\text{sign}(\cdot)$ function can be

seen as an ℓ_∞ constraint, which inspired other attacks focusing on norm-ball constraints.

2.3 Basic Iterative Method (BIM)

By taking one single step along the direction increasing the loss, FGSM can generate the adversarial examples. Just like optimization problems, a straightforward improvement of FGSM would be taking multiple steps with a smaller stepsize. Basic Iterative Method, also known as Iterative FGSM, is proposed in [Kur17a]. BIM also clips the pixel values after each step to restrict them to ϵ -neighbour of clean image. :

$$x^{i+1} = \text{Clip}_{x,\epsilon}\{x^i + \alpha \cdot \text{sign}(\nabla_x L(\theta, x^i, y))\}, \quad (2.5)$$

where x^i denotes the adversarial example at i^{th} step. $\text{Clip}_\epsilon\{\cdot\}$ is the clip function that clips the image in ϵ -neighbour of last step image. BIM algorithm starts with clean image, $x^0 = x$. The exact formulation for $\text{Clip}_\epsilon\{\cdot\}$ is as follows:

$$\text{Clip}_{x,\epsilon}\{p^{i+1}\} = \min\{255, p^i + \epsilon, \max\{0, p^i - \epsilon, p^{i+1}\}\}, \quad (2.6)$$

where p^i is the pixel value in i^{th} step. Simply speaking, $\text{Clip}_\epsilon\{\cdot\}$ cuts the pixel changed over the ϵ -neighbour of last step to a maximum ϵ change.

By sacrificing few computation budget, BIM achieves a better performance than FGSM. By modifying the parameters ϵ and α and the iteration times, the performance and computation budget of BIM can be controlled. Due to this flexibility, BIM and methods based on it are widely used in adversarial training. Moreover, this paper [Kur17a] also shows the ability of adversarial attacks can still work in physic real-world, Fig. 2.2 shows an example of the physic attack.

Another interesting finding in this [Kur17a] work is so called *least-likely class*. In order to creating a more interesting mistake, the author modified the original untargeted attack Eqn. 2.5 to a targeted one:

$$x^{i+1} = \text{Clip}_{x,\epsilon}\{x^i - \alpha \cdot \text{sign}(\nabla_x L(\theta, x^i, y_{LL}))\}, \quad (2.7)$$

where y_{LL} is the least-likely class according to the prediction of the trained network on image x :

$$y_{LL} = \arg \min_y p\{y|x\}, \quad (2.8)$$

These *least-likely class* can be seen as the most dissimilar class from the ground-truth class, when the model is well-trained. And as an result, [Kur17a] found out that adversarial examples generated by Eqn. 2.7 are more harmful to the DNN models. It decreases the targeted model's prediction accuracy the most compared with FGSM and BIM, even when ϵ is relatively small. This finding also inspires our research to some extent.

Figure 2.2 Adversarial examples are generated by one DNN classifier. Then those examples are printed in the real-world and fool another Tensorflow mobile app classifier successfully. It's even a real-world black-box attack

2.4 Attacks on ℓ_p constraint

Shortly after the discovery of FGSM [Goo15] and BIM [Kur17a], many research has been proposed based on them. As introduced above, the $\text{sign}(\cdot)$ function can be seen as an ℓ_∞ constraint, a straightforward idea derived from this point is to use other ℓ_p constraints:

$$\delta = \epsilon \cdot \frac{\nabla_x L(\theta, x, y)}{\|\nabla L(\theta, x, y)\|_p}, \quad (2.9)$$

The gradient computed from the loss function is normalized with ℓ_p norm. Many work have contributed to this extension of FGSM based attacks [Goo15; Kur17a; Don18; Mad18; Kur17b; Miy19]. Another famous work is the Projected Gradient Descent (PGD) attack [Mad18], which is essentially the same as BIM. Some may like to use the term **PGD** to refer these generation of attack instead of using **BIM** or **BIM** ℓ_p . The difference is that PGD initializes the attack from a random point decided by ℓ_p norm, and BIM initializes from the clean input. And the author [Mad18] also pointed out that BIM is the ℓ_∞ version of PGD. As well known, Projected Gradient Descent is a standard convex optimization method.

2.5 Momentum Iterative Fast Gradient Sign Method (MIFGSM)

As introduced above, the main procedure of those FGSM [Goo15] based iterative attacks can be seen as the optimization problem. An intuitive idea of improvement is to modify the optimization procedure, thus momentum is introduced to generate adversarial attacks [Don18]. By adding a

velocity vector in the direction of gradient to the loss during each iterations, momentum methods help accelerating the gradient decent algorithms and achieving a better performance. Just like normal gradient decent algorithms, BIM [Kur17a] might drop into the local maxima during the iteration steps and "overfit" the model. Hence, the author [Don18] integrated momentum into BIM [Kur17a]. First, accumulating the velocity vector in the gradient direction:

$$\mathbf{g}_{i+1} = \mu \cdot \mathbf{g}_i + \frac{\nabla_x L(\theta, x^i, y)}{\|\nabla L(\theta, x^i, y)\|_1}, \quad (2.10)$$

Then updating adversarial example x^{i+1} after gradient computation:

$$x^{i+1} = x^i + \alpha \cdot \text{sign}(\mathbf{g}_{i+1}), \quad (2.11)$$

Since MIFGSM prevents the normal BIM methods from the local maxima and the "overfitting" on the targeted model, the adversarial examples generated by MIFGSM demonstrate a powerful capability of transferability. This property of transferability makes MIFGSM a strong black-box attack, especially for the model trained with a defense mechanism.

2.6 Carlini Wagner Attack (C&W)

In [Car17], the author proposed a method to generate adversarial attack by solving the following problem:

$$\begin{aligned} &\text{minimize} \quad \mathcal{E}(\delta) = \|\delta\|_p, \\ &\text{subject to} \quad t = \arg\max_c f(x + \delta)_c, \\ &\quad \quad \quad x + \delta \in [0, 1]^n, \end{aligned} \quad (2.12)$$

where $\mathcal{E}(\cdot)$ is defined by a ℓ_p norm (e.g., the ℓ_2 norm) and n the size of the input domain (e.g., the number of pixels).

The Eqn 2.12 is really challenging because it involves solving the non-linear, non-convex constraints. To overcome the difficulty of directly solving Eqn. 2.12, the C&W method expresses it in a different form by designing some loss functions $L(x') = L(x + \delta)$ such that the first constraint $t = \arg\max_c f(x')_c$ is satisfied if and only if $L(x') \leq 0$. Then, the learning problem is formulated by:

$$\begin{aligned} &\text{minimize} \quad \|\delta\|_p + \lambda \cdot L(x + \delta), \\ &\text{subject to} \quad x + \delta \in [0, 1]^n, \end{aligned} \quad (2.13)$$

which can be solved via back-propagation with the constraint satisfied via introducing a change-of-

variables by a tanh layer:

$$\delta_i = \frac{1}{2}(\tanh(w_i) + 1) - x_i, \quad (2.14)$$

$$(2.15)$$

Since $-1 \leq \tanh(w_i) \leq 1$, this change-of-variables can satisfy the constraint $0 \leq \delta_i + x_i \leq 1$, then the final solution will be automatically valid in the range $[0, 1]^n$.

In order to find the best loss function for $L(\cdot)$, the author proposed a list of possible choices of loss functions in the paper as follows:

$$L_1(x') = -\text{loss}_{f,t}(x') + 1, \quad (2.16)$$

$$L_2(x') = (\max_{i \neq t} f(x')_i - f(x')_t)^+, \quad (2.17)$$

$$L_3(x') = \text{softplus}(\max_{c \neq t} f(x')_c - f(x')_t) - \log(2), \quad (2.18)$$

$$L_4(x') = (0.5 - f(x')_t)^+, \quad (2.19)$$

$$L_5(x') = -\log(2f(x')_t - 2), \quad (2.20)$$

$$L_6(x') = (\max_{i \neq t} z(x')_i - z(x')_t)^+, \quad (2.21)$$

$$L_7(x') = \text{softplus}(\max_{c \neq t} z(x')_c - z(x')_t) - \log(2), \quad (2.22)$$

where $\text{loss}_{f,t}$ is the cross entropy loss for x , $(e)^+$ is a short-hand for $\max(e, 0)$, and $\text{softplus}(x)$ is $\log(1 + e^x)$. After experiments and tests, the best loss function among 7 functions proposed by the C&W method is defined by the hinge loss equation Eqn. 2.21. Extend this function by $(e)^+$, the final loss function chosen is:

$$L_{CW}(x') = \max(0, \max_{c \neq t} z(x')_c - z(x')_t), \quad (2.23)$$

which induces penalties when the logit of the target label is not the maximum among all labels.

In our work, we extend this C&W method as a strong baseline for our ordered Top- k attack task, which will be introduced in next Chapter 3.

2.7 The proposed work and Contributions

Three main contributions to the field of learning adversarial attacks are introduced in our work:

- *The problem in study is novel.* We propose to learn **ordered Top- k attacks** ($k \geq 1$), which enforce the Top- k predicted labels of an adversarial example to be the k (randomly) selected and ordered labels (the ground-truth label is exclusive). This ordered Top- k adversarial attacks is an important problem that reflects the robustness of attacks themselves, but has not been addressed in the literature.

- The proposed *adversarial distillation framework* is effective. In order to solve the **ordered Top- k attacks** proposed above, we present an **Adversarial Distillation (AD)** framework. This AD framework consists of two components: (i) Computing an adversarial probability distribution for a given ordered Top- k targeted labels. (ii) Learning adversarial examples by minimizing the perturbation energy penalty together with the Kullback-Leibler (KL) divergence between the adversarial distribution and the predicted distribution.
- The proposed *knowledge-oriented adversarial distillation* is novel and effective. In order to compute adversarial distributions mentioned in our **Adversarial Distillation** framework, we explore how to leverage label semantic similarities, leading to **knowledge-oriented attacks**. And our knowledge-oriented method achieves a better performance compared to the baseline method, especially when k is large (such as $k = 5, 10$). It worth exploring the existing distillation framework for a novel problem (ordered Top- k adversarial attacks) with some novel modifications (knowledge-oriented target distributions as “teachers”).

ORDERED TOP- K ATTACK AND ADVERSARIAL DISTILLATION

3.1 Motivation

In the literature reviewed in Chapter 2, most methods address targeted attacks in the Top-1 manner, in which an adversarial attack is said to be successful if a randomly selected label (not the ground-truth label) is predicted as the Top-1 label with the added perturbation satisfying to be visually-imperceptible. One question arises,

- *The “robustness” of an attack method itself:* How far is the attack method able to push the underlying ground-truth label in the prediction of the learned adversarial examples?

Table 3.1 shows the evaluation results of the “robustness” of different attack methods. The widely used C&W method [Car17] does not push the GT labels very far, especially when smaller perturbation energy is aimed using larger search range (*e.g.*, the average rank of the GT label is 2.6 for C&W_{9×1000}). Consider Top-5, if the ground-truth labels of adversarial examples still largely appear in the Top-5 of the prediction, we may be over-confident about the 100% ASR, especially when some downstream modules may rely on Top-5 predictions in their decision making. On the contrary, the three untargeted attack approaches are much better in terms of pushing the GT labels since they usually move against the GT label explicitly in the optimization, but the energies of learned perturbation are usually much larger since no explicit constraints are posed. As we shall show, more “robust” attack methods can be developed by harnessing the advantages of the two types of attack

Table 3.1 Results showing where the ground-truth (GT) labels are in the prediction of learned adversarial examples for different attack methods. The test is performed in ImageNet-1000 val dataset using ResNet-50 [He16] pretrained with clean images. The subscripts of methods indicate the computing budgets used. See Chapter 4 for experimental settings.

Method	ASR	Proportion of GT Labels in Top- k (smaller is better)					Average Rank of GT Labels (larger is better)
		Top-3	Top-5	Top-10	Top-50	Top-100	
C&W _{9×30} [Car17]	99.9	36.9	50.5	66.3	90.0	95.1	20.4
C&W _{9×1000} [Car17]	100	71.9	87.0	96.1	99.9	100	2.6
FGSM [Goo15]	80.7	25.5	37.8	52.8	81.2	89.2	44.2
PGD ₁₀ [Mad18]	100	3.3	6.7	12	34.7	43.9	306.5
MIFGSM ₁₀ [Don18]	99.9	0.7	1.9	6.0	22.5	32.3	404.4

Figure 3.1 Illustration of a Top- k attack on a 'Dogsled' image. The Top- k predicted labels of the target model on the adversarial example will be 10 ordered labels assigned before attack from 'Lesser Panda' to 'Beagle'.

methods. In addition, the targeted Top-1 attack setting could limit the flexibility of attacks, and may lead to less rich perturbations.

To facilitate explicit control of targeted attacks and enable more “robust” attack methods, one natural solution, which is *the focus of this paper*, is to develop **ordered Top- k targeted attacks** which enforce the Top- k predicted labels of an adversarial example to be the k (randomly) selected and ordered labels ($k \geq 1$, the GT label is exclusive). Fig 3.1 is an example of a Top-10 attack on a 'Dogsled' image.

3.2 Method Overview

In order to learn ordered Top- k attacks, we *present two methods*. The basic idea is to design proper adversarial objective functions that result in imperceptible perturbations for a testing image (whose original prediction by a model is correct) through iterative gradient-based back-propagation. *First*, we extend the vanilla Carlini-Wagner (C&W) method [Car17], denoted by CW^k , and use it as a

Figure 3.2 Comparisons of Top- k attack success rates (ASR, higher is better) between the modified C&W method (CW^k) and the proposed Adversarial Distillation (AD) method ($k = 1, 2, 5, 10$). The thickness of plotted lines represents the reverse ℓ_2 energy of the learned perturbation (thicker is better). 9×30 and 9×1000 represent the computing budgets in learning (see Chapter 4 for experimental settings).

strong baseline. *Second*, we present an **Adversarial Distillation (AD)** framework consisting of two components: (i) Computing an adversarial probability distribution for any given ordered Top- k targeted labels. (ii) Learning adversarial examples by minimizing the Kullback-Leibler (KL) divergence between the adversarial distribution and the predicted distribution, together with the perturbation energy penalty.

The proposed AD framework can be viewed as applying the network distillation frameworks [Hin15; Buc06; Pap16] for “the bad” induced by target adversarial distributions. To compute a proper adversarial distribution for a given ordered Top- k targeted labels, the AD framework is motivated by two aspects: (i) The difference between the objective functions used by the C&W method and the three untargted attack methods (Table 3.1) respectively. The former maximizes the margin of the logits between the target and the runner-up (either GT or not), while the latter maximizes the cross-entropy between the prediction probabilities (softmax of logits) and the one-hot distribution of the ground-truth. (ii) The label smoothing methods [Sze15; Per17], which are often used to improve the performance of DNNs by addressing the over-confidence issue in the one-hot vector encoding of labels. We explore how to leverage label semantic similarities in computing “smoothed” adversarial distributions, leading to **knowledge-oriented attacks**. We measure label semantic similarities using the cosine distance between some off-the-shelf word2vec embedding of labels such as the pretrained Glove embedding [Pen14]. Along this direction, another question of interest is further investigated: *Are all Top- k targets equally challenging for an attack approach?* The answer is no (as intuitively perceived), and we observe some meaningful “blind spots” in experiments.

Figure 3.3 Learned adversarial examples of ordered Top-10 adversarial attacks for ResNet-50 [He16] pre-trained with clean images. The proposed AD method has smaller perturbation energies and “cleaner” (lower-entropy) prediction distributions than the proposed modified C&W method (CW^k). Note that for Top-10 attacks, the 9 × 30 search scheme does not work (see Table. 4.1). See text for detail.

In experiments, we test Top- k ($k = 1, 2, 5, 10$) in the ImageNet-1000 [Rus15] val dataset using three representative DNNs trained with clean ImageNet-1000 train dataset, ResNet-50 [He16], DenseNet-121 [Hua17] and AOGNet-12M [Li19] respectively. Overall, the proposed AD approach obtains the best results. It reduces the perturbation energy consistently with the same attack success rate on all the four k ’s, and improve the attack success rate by a large margin against the modified CW^k method for $k = 10$ (Fig. 3.2 and learned adversarial examples in Fig. 3.3). We observe that Top- k targets that are distant from the GT label in terms of either label semantic distance or prediction scores of clean images are actually more difficult to attack. In summary, *not only can ordered Top- k attacks improve the “robustness” of attacks, but also they provide insights on how aggressive adversarial attacks can be (under limited computing budgets).*

3.3 Problem Formulation

In this section, we first define the white-box attack setting and the widely used C&W method [Car17] under the Top-1 protocol, to be self-contained. Then we present the ordered Top- k attack formulation. To learn ordered Top- k attacks, we present details of the modified C&W method, CW^k, as a strong baseline and the proposed AD framework.

3.3.1 Background

We focus on image classification tasks using DNNs. Denote by (x, y) a clean input, $x \in \mathcal{X}$ and its ground-truth label $y \in \mathcal{Y}$. For example, in the ImageNet-1000 classification task, x represents a RGB image defined in the lattice of 224×224 and we have $\mathcal{X} \triangleq R^{3 \times 224 \times 224}$. y is the category label and we have $\mathcal{Y} \triangleq \{1, \dots, 1000\}$. Let $f(\cdot; \Theta)$ be a DNN pretrained with clean training images where Θ collects all estimated parameters and is **frozen** in learning adversarial examples. For notation simplicity, we denote by $f(\cdot)$ a pretrained DNN. The prediction for an input x from $f(\cdot)$ is usually defined using softmax function by,

$$P = f(x) = \text{softmax}(z(x)), \quad (3.1)$$

where $P \in R^{|\mathcal{Y}|}$ represents the estimated confidence/probability vector ($P_c \geq 0$ and $\sum_c P_c = 1$) and $z(x)$ is the logit vector. The predicted label is then inferred by $\hat{y} = \arg \max_{c \in [1, |\mathcal{Y}|]} P_c$.

*The traditional **Top-1 protocol** of learning targeted attacks.* For an input (x, y) , given a target label $t \neq y$, we seek to compute some visually-imperceptible perturbation $\delta(x, t, f)$ using the pretrained and fixed DNN $f(\cdot)$ under the white-box setting. **White-box attacks** assume the complete knowledge of the pretrained DNN f , including its parameter values, architecture, training method, etc. The perturbed example is defined by,

$$x' = x + \delta(x, t, f), \quad (3.2)$$

which is called **an adversarial example** of x if $t = \hat{y}' = \arg \max_c f(x')_c$ and the perturbation $\delta(x, t, f)$ is sufficiently small according to some energy metric.

The C&W Method [Car17]. As we introduced in Chapter 2, C&W method formulate the problem as:

$$\begin{aligned} & \text{minimize} \quad \mathcal{E}(\delta) = \|\delta\|_p, \\ & \text{subject to} \quad t = \arg \max_c f(x + \delta)_c, \\ & \quad \quad \quad x + \delta \in [0, 1]^n, \end{aligned} \quad (3.3)$$

To overcome the difficulties caused by the constraints, the problem is re-formulated by:

$$\begin{aligned} & \text{minimize} \quad \|\delta\|_p + \lambda \cdot L(x + \delta), \\ & \text{subject to} \quad x + \delta \in [0, 1]^n, \end{aligned} \quad (3.4)$$

where $L(\cdot)$ is the loss which has the best performance among all 7 testing loss functions:

$$L_{CW}(x') = \max(0, \max_{c \neq t} z(x')_c - z(x')_t). \quad (3.5)$$

Computing Budget: For the trade-off parameter λ (Eqn. 3.4), a binary search will be performed during the learning. For example, typically, 9×30 and 9×1000 are used by the C&W method, which

represents 9 tries of different values for λ and 30 or 1000 back-propagation iterations used for each λ .

3.3.2 The Proposed Ordered Top- k Attack Setting

We extend Eqn. 3.3 for learning ordered Top- k attacks ($k \geq 1$). Denote by $(t_1, \dots, t_k)$ the ordered Top- k targets ($t_i \neq y$). We have,

$$\begin{aligned} & \text{minimize} \quad \mathcal{E}(\delta) = \|\delta\|_p, \\ & \text{subject to} \quad t_i = \arg \max_{c \in [1, |\mathcal{Y}|], c \notin \{t_1, \dots, t_{i-1}\}} f(x + \delta)_c, \\ & \quad \quad \quad i \in \{1, \dots, k\}, \\ & \quad \quad \quad x + \delta \in [0, 1]^n. \end{aligned} \tag{3.6}$$

3.3.3 Learning Ordered Top- k Attacks

3.3.3.1 The Modified C&W Method, CW^k

We modify the loss function (Eqn. 3.5) of the C&W method accordingly to solve Eqn. 3.6. We have,

$$L_{CW}^{(k)}(x') = \sum_{i=1}^k \max \left(0, \max_{j \notin \{t_1, \dots, t_i\}} z(x')_j - \min_{l \in \{t_1, \dots, t_i\}} z(x')_l \right), \tag{3.7}$$

which covers the vanilla C&W loss (Eqn. 3.5), , when $k = 1$, $L_{CW}(x') = L_{CW}^{(1)}(x')$. The C&W loss function does not care where the underlying GT label will be as long as it is not in the Top- k . On the one hand, it is powerful in terms of attack success rate. On the other hand, the GT label may be very close to the Top- k , leading to over-confident attacks (Tabel. 3.1). In addition, it is generic for any given Top- k targets. As we shall show, they are less effective if we select the Top- k targets from the sub-set of labels which are least like the ground-truth label in terms of label semantics.

3.3.3.2 Knowledge-Oriented Adversarial Distillation

To overcome the shortcomings of the C&W loss function and In our adversarial distillation framework, we adopt the view of point proposed in the network distillation method [Hin15] that the full confidence/probability distribution summarizes the knowledge of a trained DNN. We hypothesize that we can leverage the network distillation framework to learn the ordered Top- k attacks by designing a proper adversarial probability distribution across the entire set of labels that satisfies the specification of the given ordered Top- k targets, and facilitates explicit control of placing the GT label, as well as top-down integration of label semantics.

Consider a given set of Top- k targets, $\{t_1, \dots, t_k\}$, denoted by P^{AD} the adversarial probability distribution in which $P_{t_i}^{AD} > P_{t_j}^{AD} (\forall i < j)$ and $P_{t_i}^{AD} > P_l^{AD} (\forall l \notin \{t_1, \dots, t_k\})$. The space of candidate distributions are huge. We present a simple knowledge-oriented approach to define the adversarial

distribution. We first specify the logit distribution and then compute the probability distribution using softmax. Denote by Z the maximum logit (*e.g.*, $Z = 10$ in our experiments).

We define the adversarial logits for the ordered Top- k targets by,

$$z_{t_i}^{AD} = Z - (i - 1) \times \gamma, \quad i \in [1, \dots, k], \quad (3.8)$$

where γ is an empirically chosen decreasing factor (*e.g.*, $\gamma = 0.3$ in our experiments).

For the remaining categories $l \notin \{t_1, \dots, t_k\}$, we define the adversarial logit by,

$$z_l^{AD} = \alpha \times \frac{1}{k} \sum_{i=1}^k s(t_i, l) + \epsilon, \quad (3.9)$$

where $0 \leq \alpha < z_{t_k}^{AD}$ is the maximum logit that can be assigned to any j , $s(a, b)$ is the semantic similarity between the label a and label b , and ϵ is a small position for numerical consideration (*e.g.*, $\epsilon = 1e-5$). We compute $s(a, b)$ using the cosine distance between the Glove [Pen14] embedding vectors of category names and $-1 \leq s(a, b) \leq 1$. Here, when $\alpha = 0$, we discard the semantic knowledge and treat all the remaining categories equally. Note that our design of P^{AD} is similar in spirit to the label smoothing technique and its variants [Sze15; Per17] except that we target attack labels and exploit label semantic knowledge. The design choice is still preliminary, although we observe its effectiveness in experiments. We hope this can simulate more sophisticated work to be explored.

With the adversarial probability distribution P^{AD} defined above as the target, we use the KL divergence as the loss function in our adversarial distillation framework as done in network distillation [Hin15] and we have,

$$L_{AD}^{(k)}(x') = KL(f(x') || P^{AD}), \quad (3.10)$$

and then we follow the same optimization scheme as done in the C&W method (Eqn. 3.4).

CHAPTER

4

EXPERIMENTS

4.1 Setup

In this section, we evaluate ordered Top- k attacks with $k = 1, 2, 5, 10$ in the ImageNet-1000 benchmark [Rus15] using three representative pretrained DNNs: (i) ResNet-50 [He16]. ResNets are the most widely used DNNs. (ii) DenseNet-121 [Hua17]. DenseNets are also popular in practice. (iii) AOGNet-12M [Li19]. AOGNets are grammar-guided networks with the vanilla Bottleneck building block, which represent an interesting direction of network architecture engineering. So, the attacking results by the proposed methods will be broadly useful for ResNets and DenseNets based deployment in practice and potentially insightful for on-going and future development of more powerful and robust DNNs. Due to computational cost, we choose the small versions of the three different DNNs. Pretrained models of ResNet-50 and DenseNet-121 are from the PyTorch model zoo¹. Pretrained AOGNet-12M is from their Github repo². We implement the proposed two methods using the AdverTorch toolkit³. *Our reproducible source code will be released.*

4.1.1 Data.

In ImageNet-1000 [Rus15], there are 50,000 images for validation. To study attacks, we utilize the subset of images for which the predictions of all the three networks, ResNet-50, DenseNet-121 and AOGNet-12M, are correct. To reduce the computational demand, we randomly sample a smaller subset, as commonly done in the literature. We first randomly select 500 categories to enlarge the

¹<https://github.com/pytorch/vision/tree/master/torchvision/models>

²<https://github.com/iVMCL/AOGNets>

³<https://github.com/BorealisAI/advertorch>

Table 4.1 Results and comparisons under the ordered Top- k targeted attack protocol using randomly selected and ordered k **targets** (GT exclusive) in ImageNet using **ResNet-50**. For Top-1 attacks, we also compare with three state-of-the-art untargeted attack methods, FGSM [Goo15], PGD [Mad18] and MIFGSM [Don18]. 10 iterations are used for both PGD and MIFGSM.

Protocol	Attack Method	Best Case				Average Case				Worst Case			
		ASR	ℓ_1	ℓ_2	ℓ_∞	ASR	ℓ_1	ℓ_2	ℓ_∞	ASR	ℓ_1	ℓ_2	ℓ_∞
Top-10	CW $_{9 \times 30}^k$	0	N.A.	N.A.	N.A.	0	N.A.	N.A.	N.A.	0	N.A.	N.A.	N.A.
	AD $_{9 \times 30}$	0.8	2579	8.18	0.096	0.16	2579	8.18	0.096	0	N.A.	N.A.	N.A.
	CW $_{9 \times 100}^k$	43.4	2336	7.83	0.109	11.8	2330	7.82	0.109	0.1	2479	8.26	0.119
	AD $_{9 \times 100}$	91.8	1677	5.56	0.088	51.2	1867	6.14	0.098	5.6	2021	6.62	0.110
	CW $_{9 \times 1000}^k$	97.7	1525	5.26	0.092	64.5	1742	5.99	0.103	20.4	1898	6.61	0.120
	AD $_{9 \times 1000}$	99.8	678	2.45	0.060	98.4	974	3.45	0.081	94.3	1278	4.48	0.103
Improvement		2.1 (3.0%)				33.9 (52.6%)				73.9 (362.3%)			
Top-5	CW $_{9 \times 30}^k$	75.8	2370	7.76	0.083	29.34	2425	7.94	0.086	0.7	2553	8.37	0.094
	AD $_{9 \times 30}$	96.1	1060	3.58	0.056	80.68	1568	5.13	0.070	49.8	2215	7.07	0.087
	CW $_{9 \times 1000}^k$	100	437	1.59	0.044	100	600	2.16	0.058	100	779	2.77	0.074
	AD $_{9 \times 1000}$	100	285	1.09	0.034	100	359	1.35	0.043	100	456	1.68	0.055
Top-2	CW $_{9 \times 30}^k$	99.9	1002	3.40	0.037	99.36	1504	4.95	0.050	97.9	2007	6.52	0.065
	AD $_{9 \times 30}$	99.9	308	1.12	0.028	99.5	561	1.94	0.037	98.4	873	2.92	0.049
	CW $_{9 \times 1000}^k$	100	185	0.72	0.025	100	241	0.91	0.033	100	303	1.12	0.042
	AD $_{9 \times 1000}$	100	137	0.56	0.022	100	174	0.70	0.028	100	220	0.85	0.035
Top-1	C&W $_{9 \times 30}$	100	209.7	0.777	0.022	99.92	354.1	1.273	0.031	99.9	560.9	1.987	0.042
	AD $_{9 \times 30}$	100	140.9	0.542	0.018	99.9	184.6	0.696	0.025	99.9	238.6	0.880	0.032
	C&W $_{9 \times 1000}$	100	95.6	0.408	0.017	100	127.2	0.516	0.023	100	164.1	0.635	0.030
	AD $_{9 \times 1000}$	100	81.3	0.380	0.016	100	109.6	0.472	0.023	100	143.9	0.579	0.029
	FGSM	2.3	9299	24.1	0.063	0.46	9299	24.1	0.063	0	N.A.	N.A.	N.A.
	PGD $_{10}$	99.6	4691	14.1	0.063	88.1	4714	14.2	0.063	57.1	4748	14.3	0.063
MIFGSM $_{10}$		100	5961	17.4	0.063	99.98	6082	17.6	0.063	99.9	6211	17.9	0.063

coverage of categories, and then randomly chose 2 images per selected categories, resulting in 1000 test images in total.

4.1.2 Settings.

We follow the protocol used in the C&W method. We only test ℓ_2 norm as the energy penalty for perturbations in learning (Eqn. 3.4). But, we evaluate learned adversarial examples in terms of three norms (ℓ_1 , ℓ_2 and ℓ_∞). We test two search schema for the trade-off parameter λ in optimization: both use 9 steps of binary search, and 30 and 1000 iterations of optimization are performed for each trial of λ . In practice, computation budget is an important factor and less computationally expensive ones are usually preferred. Only $\alpha = 1$ is used in Eqn. 3.9 in experiments for simplicity due to computational demand. We compare the results under three scenarios proposed in the C&W method [Car17]: *The Best Case* settings test the attack against all incorrect classes, and report the target class(es) that was least difficult to attack. *The Worst Case* settings test the attack against all incorrect classes, and report the target class(es) that was most difficult to attack. *The Average Case* settings select the target class(es) uniformly at random among the labels that are not the GT.

Figure 4.1 Learned adversarial examples of ordered Top-5 (top) and vanilla Top-1 (bottom) attacks for ResNet-50 [He16] pretrained with clean images. The proposed AD method has smaller perturbation energies and “cleaner” (lower-entropy) prediction distributions than both the modified CW^k method in Top-5 attacks and the vanilla C&W method in the Top-1 attacks.

4.2 Results

4.2.1 Results for ResNet-50

We first test ordered Top- k attacks using ResNet-50 for the four selected k 's. Fig. 4.1 shows some learned adversarial examples of ordered Top-5 and Top-1 attacks (Top-10 attack examples in Fig. 3.3). The visualizations are similar across different networks. Table. 4.1 summarizes the quantitative results and comparisons. **For Top-10 attacks**, the proposed AD method obtains significantly better results in terms of both ASR and the ℓ_2 energy of the added perturbation. For example, the proposed AD method has *relative 362.3% ASR improvement* over the strong C&W baseline for the worst case setting. **For Top-5 attacks**, the AD method obtains significantly better results when the search budget is relatively low (i.e., 9×30). **For Top- k ($k = 1, 2$) attacks**, both the C&W method and the AD method can achieve 100% ASR, but the AD method has consistently lower energies of the added perturbation, i.e., finding more effective attacks and richer perturbations.

4.2.2 Results for DenseNet-121 and AOGNet-12M

Overall, we obtain similar results for DenseNet-121 summarized in Table. 4.2. For AOGNet-12M (Table. 4.3), the proposed AD does not show improvement as significant as for the other two networks, especially for Top-10.

4.2.3 Are all Top- k targets equally difficult to attack?

Intuitively, we understand that they should not be equally difficult. We conduct some experiments to test this hypothesis. In particular, we test whether the label semantic knowledge can help identify the weak spots of different attack methods, and whether the proposed AD method can gain more in those weak spots. We test Top-5 using ResNet-50. Table. 4.4 summarizes the results. Similar results are observed for DenseNets and AOGNets. We observe that for the 9×30 search budget, attacks are more challenging if the Top-5 targets are selected from the least-like set in terms of the label semantic similarity (Eqn. 3.9), or from the lowest-score set in terms of prediction scores on clean images.

Table 4.2 Results and comparisons using **DenseNet-121** [Hua17] under the ordered Top- k targeted attack protocol using randomly selected and ordered 10 targets (GT exclusive). For Top-1 attacks, we also compare with three state-of-the-art untargeted attack methods, FGSM [Goo15], PGD [Mad18] and MIFGSM [Don18]. 10 iterations are used for both PGD and MIFGSM.

Protocol	Method	Best Case				Average Case				Worst Case			
		ASR	ℓ_1	ℓ_2	ℓ_∞	ASR	ℓ_1	ℓ_2	ℓ_∞	ASR	ℓ_1	ℓ_2	ℓ_∞
Top-10	CW $^k_{9 \times 30}$	0	N.A.	N.A.	N.A.	0	N.A.	N.A.	N.A.	0	N.A.	N.A.	N.A.
	AD $_{9 \times 30}$	1.3	2674	8.48	0.102	0.26	2674	8.48	0.102	0	N.A.	N.A.	N.A.
	CW $^k_{9 \times 100}$	45.3	2320	7.77	0.108	12.42	2338	7.83	0.109	0	N.A.	N.A.	N.A.
	AD $_{9 \times 100}$	96.9	1864	6.21	0.101	61.4	2331	7.71	0.123	13.2	2878	9.58	0.156
	CW $^k_{9 \times 1000}$	100	1163	4.09	0.085	97.16	1640	5.67	0.107	85.8	2157	7.39	0.130
	AD $_{9 \times 1000}$	100	627	2.26	0.058	100	900	3.19	0.076	100	1250	4.37	0.100
	Improvement			<i>1.83 (44.7%)</i>				<i>2.48 (43.7%)</i>				<i>3.02 (40.9%)</i>	
Top-5	CW $^k_{9 \times 30}$	96.6	2161	7.09	0.071	73.68	2329	7.65	0.080	35.6	2530	8.28	0.088
	AD $_{9 \times 30}$	97.7	6413	2.14	0.043	92.66	1063	3.57	0.057	83.3	1636	5.35	0.072
	CW $^k_{9 \times 1000}$	100	392	1.42	0.040	100	527	1.89	0.052	100	669	2.37	0.065
	AD $_{9 \times 1000}$	100	273	1.05	0.033	100	344	1.29	0.042	100	425	1.57	0.052
Top-2	CW $^k_{9 \times 30}$	99.9	549	1.92	0.033	99.72	1058	3.54	0.042	99.4	1640	5.35	0.051
	AD $_{9 \times 30}$	99.9	199	0.74	0.023	99.8	249	0.92	0.029	99.7	308	1.12	0.037
	CW $^k_{9 \times 1000}$	100	146	0.58	0.022	100	187	0.72	0.029	100	230	0.86	0.037
	AD $_{9 \times 1000}$	100	121	0.52	0.021	100	153	0.62	0.027	100	187	0.74	0.034
Top-1	C&W $_{9 \times 30}$	99.9	188.6	0.694	0.019	99.9	279.4	1.008	0.028	99.9	396.5	1.404	0.037
	AD $_{9 \times 30}$	99.9	136.4	0.523	0.017	99.9	181.8	0.678	0.024	99.9	240.0	0.870	0.031
	C&W $_{9 \times 1000}$	100	98.5	0.415	0.016	100	132.3	0.528	0.023	100	174.8	0.657	0.030
	AD $_{9 \times 1000}$	100	83.8	0.384	0.016	100	115.9	0.485	0.023	100	158.69	0.610	0.030
	FGSM	6.4	9263	24.0	0.063	1.44	9270	24.0	0.063	0	N.A.	N.A.	N.A.
	PGD $_{10}$	100	4617	14.2	0.063	97.2	4716	14.2	0.063	87.6	4716	14.2	0.063
	MIFGSM $_{10}$	100	5979	17.6	0.063	100	6095	17.6	0.063	100	6218	17.9	0.063

Table 4.3 Results and comparisons under the ordered Top- k targeted attack protocol using randomly selected and ordered 10 targets (GT exclusive) in ImageNet using **AOGNet-12M**.

Protocol	Method	Best Case				Average Case				Worst Case			
		ASR	ℓ_1	ℓ_2	ℓ_∞	ASR	ℓ_1	ℓ_2	ℓ_∞	ASR	ℓ_1	ℓ_2	ℓ_∞
Top-10	CW $^k_{9 \times 30}$	0	N.A.	N.A.	N.A.	0	N.A.	N.A.	N.A.	0	N.A.	N.A.	N.A.
	AD $_{9 \times 30}$	0	N.A.	N.A.	N.A.	0	N.A.	N.A.	N.A.	0	N.A.	N.A.	N.A.
	CW $^k_{9 \times 100}$	43.5	2221	7.50	0.11	12.56	2246	7.57	0.11	0.1	2657	8.56	0.13
	AD $_{9 \times 100}$	42.8	1920	6.47	0.11	11.64	1947	6.56	0.11	0	N.A.	N.A.	N.A.
	CW $^k_{9 \times 1000}$	100	1018	3.68	0.084	99.94	1393	4.93	0.10	99.9	1816	6.31	0.12
	AD $_{9 \times 1000}$	100	651	2.40	0.065	99.84	947	3.43	0.088	99.3	1317	4.72	0.12
Top-5	CW $^k_{9 \times 30}$	37.1	1809	6.08	0.078	10.84	1918	6.41	0.079	0.1	2290	7.80	0.090
	AD $_{9 \times 30}$	82.7	1126	3.84	0.063	46.26	1599	5.28	0.074	10.2	2396	7.71	0.093
	CW $^k_{9 \times 1000}$	100	355	1.33	0.041	100	468	1.73	0.053	100	588	2.15	0.068
	AD $_{9 \times 1000}$	100	199	0.81	0.029	100	258	1.01	0.037	100	327	1.26	0.047
Top-2	CW $^k_{9 \times 30}$	99.6	644	2.33	0.046	97.22	1176	4.04	0.059	91	1918	6.38	0.074
	AD $_{9 \times 30}$	99.9	205	0.78	0.024	98.72	283	1.06	0.032	96.2	391	1.43	0.041
	CW $^k_{9 \times 1000}$	100	131	0.53	0.021	100	178	0.70	0.029	100	232	0.90	0.038
	AD $_{9 \times 1000}$	100	96	0.44	0.019	100	124	0.54	0.025	100	154	0.65	0.032
Top-1	C&W $_{9 \times 30}$	99.9	256	0.946	0.025	99.64	532	1.90	0.039	99.1	932	3.25	0.054
	AD $_{9 \times 30}$	99.9	141	0.541	0.019	99.8	190	0.720	0.026	99.5	246	0.922	0.034
	C&W $_{9 \times 1000}$	100	80	0.36	0.016	100	114	0.48	0.023	100	153	0.61	0.030
	AD $_{9 \times 1000}$	100	62	0.312	0.016	100	86	0.398	0.021	100	115	0.496	0.028
	FGSM	1.7	9254	24.0	0.0625	0.34	9254	24.0	0.0625	0	N.A.	N.A.	N.A.
	PGD $_{10}$	100	4685	14.1	0.0625	98.16	4698	14.2	0.0625	91.2	4714	14.2	0.0625
	MIFGSM $_{10}$	100	5940	17.3	0.0625	99.92	6046	17.5	0.0625	99.6	6165	17.8	0.0625

Table 4.4 Results of ordered Top-5 targeted attacks for ResNet-50 with targets being selected based on (Top) label similarity, which uses 5 most-like labels and 5 least-like labels as targets respectively, and (Bottom) prediction score of clean image, which uses 5 highest-score labels and 5-lowest score labels. In both cases, GT labels are exclusive.

Protocol	Similarity	Method	ASR	ℓ_1	ℓ_2	ℓ_∞
Label similarity	Most like	$CW^k_{9 \times 30}$	80	1922	6.30	0.066
		$AD_{9 \times 30}$	96.5	1286	4.20	0.054
		$CW^k_{9 \times 1000}$	100	392	1.43	0.042
		$AD_{9 \times 1000}$	100	277	1.05	0.035
	Least like	$CW^k_{9 \times 30}$	27.1	2418	7.90	0.085
		$AD_{9 \times 30}$	77.1	1635	5.35	0.072
		$CW^k_{9 \times 1000}$	100	596	2.15	0.060
		$AD_{9 \times 1000}$	100	370	1.39	0.045
Prediction Score	Highest	$CW^k_{9 \times 30}$	93	1546	4.98	0.042
		$AD_{9 \times 30}$	99.9	1182	3.78	0.039
		$CW^k_{9 \times 1000}$	100	205	0.75	0.025
		$AD_{9 \times 1000}$	100	170	0.65	0.023
	Lowest	$CW^k_{9 \times 30}$	13.4	2231	7.30	0.082
		$AD_{9 \times 30}$	68.6	1791	5.86	0.077
		$CW^k_{9 \times 1000}$	100	621	2.25	0.064
		$AD_{9 \times 1000}$	100	392	1.47	0.047

CHAPTER

5

CONCLUSIONS

This paper proposes to extend the traditional Top-1 targeted attack setting to the ordered Top- k setting ($k \geq 1$) under the white-box attack protocol. The ordered Top- k targeted attacks can improve the robustness of attacks themselves. To our knowledge, it is the first work studying this ordered Top- k attacks. To learn the ordered Top- k attacks, we present a conceptually simple yet effective adversarial distillation framework motivated by network distillation. We also develop a modified C&W method as the strong baseline for the ordered Top- k targeted attacks. In experiments, the proposed method is tested in ImageNet-1000 using two popular DNNs, ResNet-50 and DenseNet-121, with consistently better results obtained. We investigate the effectiveness of label semantic knowledge in designing the adversarial distribution for distilling the ordered Top- k targeted attacks.

Discussions. We have shown that the proposed AD method is generally applicable to learn ordered Top- k attacks. But, we note that the two components of the AD framework are in their simplest forms in this paper, and need to be more thoroughly studied: designing more informative adversarial distributions to guide the optimization to learn adversarial examples better and faster, and investigating loss functions other than KL divergence such as the Jensen-Shannon (JS) divergence or the Earth-Mover distance. On the other hand, we observed that the proposed AD method is more effective when computation budget is limited (*e.g.*, using the 9×30 search scheme). This leads to the theoretically and computationally interesting question whether different attack methods all will work comparably well if the computation budget is not limited. Of course, in practice, we prefer more powerful ones when only limited computation budget is allowed. Furthermore, we observed that both the modified C&W method and the AD method largely do not work in learning Top- k ($k \geq 20$) attacks with the two search schema (9×30 and 9×1000). We are working on addressing the aforementioned issues to test the Top- k ($k \geq 20$) cases, thus providing a thorough empirical answer

to the question: *how aggressive can adversarial attacks be?*

BIBLIOGRAPHY

- [Akh18] Akhtar, N. & Mian, A. S. “Threat of Adversarial Attacks on Deep Learning in Computer Vision: A Survey”. *IEEE Access* **6** (2018), pp. 14410–14430.
- [Arn18] Arnab, A. et al. “On the Robustness of Semantic Segmentation Models to Adversarial Attacks”. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.
- [Ath17] Athalye, A. & Sutskever, I. “Synthesizing Robust Adversarial Examples”. *CoRRabs/1707.07397* (2017).
- [Bre18] Brendel, W. et al. “Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models”. *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.
- [Bro17] Brown, T. B. et al. “Adversarial patch”. *arXiv preprint arXiv:1712.09665* (2017).
- [Buc06] Bucila, C. et al. “Model compression”. *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Philadelphia, PA, USA, August 20-23, 2006*. 2006, pp. 535–541.
- [Car17] Carlini, N. & Wagner, D. A. “Towards Evaluating the Robustness of Neural Networks”. *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*. 2017, pp. 39–57.
- [Cha18] Chakraborty, A. et al. “Adversarial Attacks and Defences: A Survey”. *CoRRabs/1810.00069* (2018). arXiv: 1810.00069.
- [Che17] Chen, P. et al. “ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks without Training Substitute Models”. *AISeC@CCS*. ACM, 2017, pp. 15–26.
- [Che18] Chen, P. et al. “EAD: Elastic-Net Attacks to Deep Neural Networks via Adversarial Examples”. *AAAI*. AAAI Press, 2018, pp. 10–17.
- [Don18] Dong, Y. et al. “Boosting Adversarial Attacks With Momentum”. *CVPR*. IEEE Computer Society, 2018, pp. 9185–9193.
- [Don19a] Dong, Y. et al. “Efficient Decision-Based Black-Box Adversarial Attacks on Face Recognition”. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019.
- [Don19b] Dong, Y. et al. “Evading Defenses to Transferable Adversarial Examples by Translation-Invariant Attacks”. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. 2019, pp. 4312–4321.
- [Ebr18] Ebrahimi, J. et al. “HotFlip: White-Box Adversarial Examples for Text Classification”. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*,

ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 2: Short Papers. 2018, pp. 31–36.

- [Gle20] Gleave, A. et al. “Adversarial Policies: Attacking Deep Reinforcement Learning”. *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.
- [Goo15] Goodfellow, I. J. et al. “Explaining and Harnessing Adversarial Examples”. *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*. 2015.
- [He16] He, K. et al. “Deep Residual Learning for Image Recognition”. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.
- [Hin15] Hinton, G. E. et al. “Distilling the Knowledge in a Neural Network”. *CoRR* **abs/1503.02531** (2015). arXiv: 1503.02531.
- [Hor89] Hornik, K. et al. “Multilayer feedforward networks are universal approximators”. *Neural Networks* **2.5** (1989), pp. 359–366.
- [Hua17] Huang, G. et al. “Densely connected convolutional networks”. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.
- [Ily19] Ilyas, A. et al. “Adversarial Examples Are Not Bugs, They Are Features”. *CoRR* **abs/1905.02175** (2019). arXiv: 1905.02175.
- [Kan18] Kannan, H. et al. “Adversarial Logit Pairing”. *CoRR* **abs/1803.06373** (2018). arXiv: 1803.06373.
- [Kos18] Kos, J. et al. “Adversarial Examples for Generative Models”. *2018 IEEE Security and Privacy Workshops, SP Workshops 2018, San Francisco, CA, USA, May 24, 2018*. 2018, pp. 36–42.
- [Kri12] Krizhevsky, A. et al. “ImageNet Classification with Deep Convolutional Neural Networks”. *Neural Information Processing Systems (NIPS)*. 2012, pp. 1106–1114.
- [Kur17a] Kurakin, A. et al. “Adversarial examples in the physical world”. *ICLR (Workshop)*. OpenReview.net, 2017.
- [Kur17b] Kurakin, A. et al. “Adversarial Machine Learning at Scale”. *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.
- [LeC98] LeCun, Y. et al. “Gradient-based learning applied to document recognition”. *Proceedings of the IEEE* **86.11** (1998), pp. 2278–2324.
- [Li19] Li, X. et al. “AOGNets: Compositional Grammatical Architectures for Deep Learning”. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. 2019, pp. 6220–6230.

- [Lin17] Lin, Y. et al. “Tactics of Adversarial Attack on Deep Reinforcement Learning Agents”. *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*. 2017, pp. 3756–3762.
- [Liu17] Liu, Y. et al. “Delving into Transferable Adversarial Examples and Black-box Attacks”. *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. 2017.
- [Mad18] Madry, A. et al. “Towards Deep Learning Models Resistant to Adversarial Attacks”. *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. 2018.
- [Miy19] Miyato, T. et al. “Virtual Adversarial Training: A Regularization Method for Supervised and Semi-Supervised Learning”. *IEEE Trans. Pattern Anal. Mach. Intell.* **41.8** (2019), pp. 1979–1993.
- [Ngu15] Nguyen, A. M. et al. “Deep neural networks are easily fooled: High confidence predictions for unrecognizable images”. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*. 2015, pp. 427–436.
- [Pap17a] Papernot, N. & McDaniel, P. D. “Extending Defensive Distillation”. *CoRR* **abs/1705.05264** (2017). arXiv: 1705.05264.
- [Pap16] Papernot, N. et al. “Distillation as a Defense to Adversarial Perturbations Against Deep Neural Networks”. *IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2016, pp. 582–597.
- [Pap17b] Papernot, N. et al. “Practical Black-Box Attacks against Machine Learning”. *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017*. 2017, pp. 506–519.
- [Pen14] Pennington, J. et al. “Glove: Global vectors for word representation”. In *EMNLP*. 2014.
- [Per17] Pereyra, G. et al. “Regularizing Neural Networks by Penalizing Confident Output Distributions”. *CoRR* **abs/1701.06548** (2017). arXiv: 1701.06548.
- [Qin19] Qin, Y. et al. “Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition”. *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*. 2019, pp. 5231–5240.
- [Ron19] Rony, J. et al. “Decoupling Direction and Norm for Efficient Gradient-Based L2 Adversarial Attacks and Defenses”. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. 2019, pp. 4322–4330.
- [Rus15] Russakovsky, O. et al. “ImageNet Large Scale Visual Recognition Challenge”. *International Journal of Computer Vision (IJCV)* **115.3** (2015), pp. 211–252.
- [Sha16] Sharif, M. et al. “Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition”. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*. 2016, pp. 1528–1540.

- [Son18] Song, D. et al. “Physical Adversarial Examples for Object Detectors”. *12th USENIX Workshop on Offensive Technologies (WOOT 18)*. Baltimore, MD: USENIX Association, 2018.
- [Su19] Su, J. et al. “One Pixel Attack for Fooling Deep Neural Networks”. *IEEE Trans. Evolutionary Computation* **23.5** (2019), pp. 828–841.
- [Sze14] Szegedy, C. et al. “Intriguing properties of neural networks”. *ICLR*. 2014.
- [Sze15] Szegedy, C. et al. “Rethinking the Inception Architecture for Computer Vision”. *CoRR abs/1512.00567* (2015). arXiv: 1512.00567.
- [Sze16] Szegedy, C. et al. “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning”. *CoRR abs/1602.07261* (2016).
- [Xie17] Xie, C. et al. “Adversarial Examples for Semantic Segmentation and Object Detection”. *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*. 2017, pp. 1378–1387.
- [Xie19a] Xie, C. et al. “Feature Denoising for Improving Adversarial Robustness”. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. 2019, pp. 501–509.
- [Xie19b] Xie, C. et al. “Improving Transferability of Adversarial Examples With Input Diversity”. *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*. 2019, pp. 2730–2739.
- [Xu18] Xu, K. et al. “Structured Adversarial Attack: Towards General Implementation and Better Interpretability”. *CoRR abs/1808.01664* (2018).
- [Zha16] Zhang, C. et al. “Understanding deep learning requires rethinking generalization”. *CoRR abs/1611.03530* (2016). arXiv: 1611.03530.