Summer 2024

Adversarial Transferability and Generalization in Robust Deep Learning

Tao Wu
Missouri University of Science and Technology

ADVERSARIAL TRANSFERABILITY AND GENERALIZATION IN ROBUST DEEP LEARNING

by

TAO WU

A DISSERTATION

Presented to the Graduate Faculty of the

MISSOURI UNIVERSITY OF SCIENCE AND TECHNOLOGY

In Partial Fulfillment of the Requirements for the Degree

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

2024

Approved by:

Donald C. Wunsch II, Advisor
Tie Luo, Co-Advisor
Huiyuan Yang
Wenqing Hu
Venkata Sriram Siddhardh Nadendla

ABSTRACT

Despite its remarkable achievements across a multitude of benchmark tasks, deep learning (DL) models exhibit significant fragility to adversarial examples, i.e., subtle modifications applied to inputs during testing yet effective in misleading DL models. These meticulously crafted perturbations possess the remarkable property of transferability: an adversarial example that effectively fools one model often retains its effectiveness against another model, even if the two models were trained independently. This research delves into the characteristics influencing the transferability of adversarial examples from three distinct and complementary perspectives: data, model, and optimization. Firstly, from the data perspective, we propose a new method of crafting transferable AE based on random erasure (RE) which erase part of image with random noise which increases the diversity of adversarial perturbations and helps stabilize gradient fluctuations. Secondly, we explore from optimization perspective by penalizing the input gradient norm when optimizing the objective for generating AE, aim to find AE within flat regions of the loss landscape. Thirdly, we investigate from the model perspective and propose a novel strategy centered on transforming surrogate models by Lipschitz regularization. Finally, we introduce the normalized Hessian trace, a metric capable of accurately and consistently characterizing the curvature of loss landscapes, based on which we propose CR-SAM, a novel optimization technique that integrates curvature regularization into the Sharpness-Aware Minimization (SAM) optimizer aims to bolster the generalizability of deep neural networks across a range of image classification tasks.

In summary, this research presents three complementary techniques that provide a comprehensive and practical approach to generating highly transferable adversarial examples. Furthermore, our exploration of metrics aimed at describing the curvature of the loss landscape contributes to a deeper understanding of the optimization process and facilitates the enhancement of deep learning models' generalizability.

ACKNOWLEDGMENTS

I would like to express my gratitude to all those who have supported me during this research. Firstly, I am deeply thankful to my advisor, Dr. Donald Wunsch for giving me the opportunity to pursue this research and my co-advisor Dr. Tie Luo whose valuable insights and suggestions have helped me overcome many challenges throughout this work. I am grateful for their advice and guidance throughout my PhD program. Additionally, I would like to thank Dr. Huiyuan Yang, Dr. Wenqing Hu and Dr. Venkata Sriram Siddhardh Nadendla for serving on my PhD committee and taking the time to review my work.

Last but not the least, I extend a heartfelt sense of gratitude to my family for their unwavering support throughout my journey to pursue my dreams. Their encouragement has been a constant source of strength, and I am profoundly grateful for their presence in my life.

TABLE OF CONTENTS

Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

1. INTRODUCTION

Since the groundbreaking achievements of AlexNet [4] in the ImageNet [5] challenge, deep learning (DL) has demonstrated remarkable progress in numerous benchmark tasks and has witnessed a significant increase in its application to solve practical problems in the real world. The utilization of DL has expanded across various domains, showcasing its potential for revolutionizing industries and driving innovation in research and development. Nevertheless, recent findings have highlighted certain issues such as vulnerability to adversarial attacks, absence of interpretability, and difficulties in achieving out-of-distribution generalization. These emerging challenges highlight that relying solely on benchmark performance is insufficient for fully understanding the capabilities and limitations of deep learning algorithms.

1.1. ACHIEVEMENTS AND LIMITATIONS

We provide a brief overview of the achievements and limitations of deep learning, examining its significant breakthroughs in various fields such as computer vision and natural language processing, as well as its challenges.

1.1.1. Notable Achievements. Remarkable advancements have marked significant milestones within the field of deep learning, spanning various domains. *Breakthroughs in Computer Vision*: Breakthroughs in the field of Computer Vision have been significantly influenced by the advancements in DL, which have brought about a revolutionary transformation in the accuracy and efficiency of various computer vision tasks. Notably, Convolutional Neural Networks (CNNs) have emerged as dominant models, showcasing unparalleled levels of performance in tasks such as image classification [6, 7], object detection [8, 9], and semantic segmentation [10, 11]. These advancements have paved the way for the development of cutting-edge applications like autonomous driving [12], medical imaging diagnostics [13], and facial recognition systems [14], thereby demonstrating the

far-reaching impact of deep learning in the realm of computer vision. *Natural Language Processing Advancements*: Deep learning has significantly advanced the field of natural language processing (NLP) by leveraging sophisticated models such as recurrent neural networks (RNNs), transformers [15], and pre-trained language models [16] to reach unprecedented levels of performance. These models have excelled in various tasks including machine translation, sentiment analysis, and text generation, setting new benchmarks in the realm of NLP. Groundbreaking technologies like Google's BERT [17] and OpenAI's GPT series [18, 19, 20, 21] have showcased exceptional language comprehension abilities, opening up avenues for the development of conversational AI systems and language-oriented applications in diverse domains. *Enhancements in Speech Recognition*: The utilization of deep learning techniques has been instrumental in the progression of speech recognition systems, facilitating the precise conversion of oral language into written text. Noteworthy advancements have been observed with the integration of models such as RNNs and DNNs, leading to a substantial enhancement in the accuracy levels of automatic speech recognition systems [22]. This progress has paved the way for the evolution of virtual assistants, voice-controlled devices, and various speech-to-text applications, demonstrating the significant impact of deep learning in the field of speech recognition.

1.1.2. Critical Limitations. Delving into the intricacies of deep learning models, a myriad of shortcomings that impede their performance and robustness have been discovered. *Vulnerability to Adversarial Attacks*: Deep learning models have shown vulnerability to adversarial examples [23, 24], a phenomenon in which barely noticeable alterations to input data can lead to substantial misclassifications. These adversarial examples present security concerns in various domains such as image recognition systems and autonomous vehicles, underscoring the delicate nature of deep learning models in the face of subtle manipulations. The susceptibility of deep learning models to adversarial attacks raises important questions about the robustness and reliability of these systems when deployed in real-world scenarios. *Limited Out-of-distribution Generalization*: Deep learning models

frequently encounter difficulties in effectively generalizing beyond the distribution of the training data, which can result in either overfitting or subpar performance when faced with unseen data instances. Such a constraint presents noteworthy obstacles when it comes to implementing deep learning solutions in practical settings, particularly those characterized by diverse or limited data availability, necessitating the development of resilient methodologies for domain adaptation and transfer learning. These challenges underscore the importance of devising strategies that enable deep learning models to adapt gracefully to new data domains, ensuring their reliable performance across a spectrum of real-world scenarios. *Interpretability and Transparency*: Deep learning models often face criticism for their lack of interpretability and transparency, presenting a significant challenge in understanding the reasoning behind their predictions. This opacity raises concerns, especially in critical sectors like healthcare and criminal justice, where the need for explainability and accountability in decision-making is crucial. In such contexts, it becomes essential to clarify and justify the outcomes of deep learning models, as opaque decision-making can have profound consequences for individuals and society.

1.2. BACKGROUND

1.2.1. Adversarial Examples. Adversarial examples, initially brought to prominence by the seminal work of [23], are inputs purposefully crafted by applying human imperceptible perturbation to the clean inputs which can lead to erroneous predictions by even the most sophisticated DNNs. As shown in Figure 1.1, starting with a correctly-classified input image (left), the addition of a meticulously calculateed imperceptible perturbation (center) leads to the creation of an adversarial example (right). Despite the model's high confidence in its prediction, the outcome is erroneous, highlighting the vulnerability of even advanced classifiers to subtle manipulations.

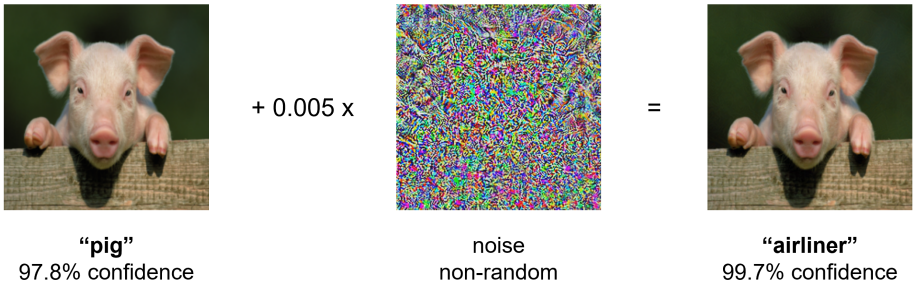


Figure 1.1. An adversarial example demonstrates the capability to deceive a state-of-the-art vision classifier ResNet50. In the world of deep learning, a subtle noise can make a pig fly.

Based on adversary's knowledge to the model, adversarial attacks can be grouped into *white-box attacks, black-box attacks*. In white-box setting, one assumes the attackers possess perfect knowledge about the target model, including the architecture, parameters, and gradient of the loss w.r.t. the input. Most methods adopt the gradient information of the target model to launch adversarial attacks under the white-box setting. However, white box attacks are almost unrealistic in real applications because the model structure and parameters are usually hidden from the attackers. Black-box attacks pose a greater real-world threat to DL systems, which can be grouped into three scenarios, including score-based, decision-based and transfer-based attack. Score-based black-box attacks can acquire the output probabilities by querying the target model, and the gradient can be estimated through queries. Decision-based black-box attacks can only solely rely on the predicted classes of the queries, this setting is more challenging since the target model only provides discrete hard-label predictions. Transfer-based black-box attacks require the least knowledge of the target model which are based on the transferability of adversarial examples [23]. Transfer-based black-box attacks are the main topic we study in this dissertation where we generate adversarial examples on surrogate models which are white-box to us, then the generated adversarial examples are transferred to black-box target models. In this setting, the most important aspect is to improve the transferability of adversarial examples so that transfer-based black-box attacks can be made more effective in real world scenarios.

To be more specific, let x be a benign image, y the corresponding true label and $f(x;\theta)$ the classifier with parameters θ and which outputs the prediction result. Let $\ell(x, y;\theta)$ denote the loss function (e.g., cross-entropy loss) of the classifier f. We define an adversarial attack as finding an adversarial example x^{adv} that satisfies $\|x^{adv} - x\|_p \leq \epsilon$ but incurs misclassification to the model, i.e., $f(x;\theta) \neq f(x^{adv};\theta)$. Here $\|\cdot\|_p$ denotes p-norm and we consider $p = \infty$ in this dissertation to be consistent with previous works. Mathematically, given a benign (clean) example x, we seek to find an AE x^{adv} as the solution to the following constrained optimization problem:

$$\arg\max_{x^{adv}} \ell(x^{adv}, y;\theta), \qquad \text{s.t. } \left\|x^{adv} - x\right\|_\infty \leq \epsilon \tag{1.1}$$

There have seen a large number of adversarial attack methods for solving the above problem, including gradient-based methods [1, 2, 24, 25, 26], optimization-based methods [23, 27], score-based methods [28, 29], and decision-based methods [30, 31]. In this dissertation, we focus mainly on gradient-based methods which have attracted the most attention.

Fast Gradient Sign Method (FGSM). FGSM [24] is the first gradient-based attack which crafts an adversarial example x^{adv} by attempting to maximize the loss function $\ell(x^{adv}, y;\theta)$ with a one-step gradient update:

$$x^{adv} = x + \epsilon \cdot \text{sign}(\nabla_x \ell(x, y;\theta)),$$

where $\nabla_x \ell(x, y;\theta)$ is the gradient of loss function with respect to x, $\text{sign}(\cdot)$ denotes the sign function, and ϵ denote the perturbation bound. Many subsequent methods have built upon and improved FGSM for enhancing adversarial tranferability.

Iterative Fast Gradient Sign Method (I-FGSM). I-FGSM [25] extends FGSM to an iterative version:

$$x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \text{sign}(\nabla_{x_t^{adv}} \ell(x_t^{adv}, y; \theta)), \tag{1.2}$$
$$x_0^{adv} = x,$$

where $\alpha = \epsilon/T$ is a small step size and T is the number of iterations.

Momentum Iterative Fast Gradient Sign Method (MI-FGSM). MI-FGSM [1] integrates a momentum term into I-FGSM and achieves much better transferability:

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_{x_t^{adv}} J(x_t^{adv}, y; \theta)}{\|\nabla_{x_t^{adv}} J(x_t^{adv}, y; \theta)\|_1}, \tag{1.3}$$
$$x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \text{sign}(g_{t+1}),$$

where $g_0 = 0$ and μ is a decay factor.

Nesterov Iterative Fast Gradient Sign Method (NI-FGSM). [2] integrates Nesterov's accelerated gradient (NAG) [32] into the iterative attack method, by replacing x_t^{adv} in (1.3) with $\tilde{x}_t^{adv}$ which is defined as

$$\tilde{x}_t^{adv} = x_t^{adv} + \alpha \cdot \mu \cdot g_t \tag{1.4}$$

Other notable methods for white-box adversarial attacks include Project Gradient Descent (PGD) [26] which extends FGSM by iteratively taking multiple small gradient steps and projecting the generated adversarial example onto the ϵ-sphere around the clean sample at each step, and Carlini and Wagner Attack (C&W) [27] which reformulates the constrained loss into an Lagrangian form and adopts Adam [33] for optimization.

1.2.2. Defending Against Adversarial Attacks. Due to the threat posed by adversarial examples, extensive research has been conducted to develop robust models capable of defending against various adversarial attacks. There are roughly three primary research directions in adversarial robustness.

The first direction is adversarial training [24, 26, 34], which involves injecting generated adversarial samples into the training data to help the model differentiate between adversarial and benign examples. For instance, [26] proposes augmenting the training data with adversarial examples crafted by the PGD attack, which remains the state-of-the-art defense to date. Despite its promise, adversarial training is computationally expensive and challenging to scale to large datasets [35].

The second approach involves input transformation. These methods preprocess input images to mitigate adversarial perturbations without compromising the classification accuracy on benign images. Examples of input transformation methods include random resizing and padding [36], JPEG compression [37], bit-depth reduction [38], total variance minimization [39], and autoencoder-based denoising [40]. However, these defenses can lead to shattered gradients or vanishing/exploding gradients, which adaptive attacks can exploit [41].

The third category is certified defenses, which are mathematically proven to be robust against the worst-case attacks under certain assumptions. The goal of certified defenses is to end the ongoing arms race between adversarial defenders and attackers. Recent certified defenses [42, 43] have been scaled to ImageNet, demonstrating the applicability of this defense type.

Additionally, model ensemble is another effective defense strategy that leverages the outputs from an ensemble of individual models [44, 45]. Model ensembles can be integrated with the aforementioned defenses, such as ensemble adversarial training [34], which significantly enhances the robustness of adversarial training.

1.2.3. Adversarial Transferability. The transferability of adversarial examples enables transfer-based black-box attacks [23]. Such attacks require the least knowledge of target models and thus often pose the biggest threat to AI systems deployed in the real world. This black-box approach is to apply white-box attacks on surrogate models to find adversarial examples that can fool as many black-box target models as possible, known as transferability of the AE. Many works have been proposed to improve the transferability of AE. *Optimization-based approaches* focus on finding direction of the gradients towards optima that lead to better transferability. For example, Momentum Iterative Method (MIM) [1] integrates a momentum term into the gradient calculation to stabilize the update direction. Reverse Adversarial Perturbation (RAP) [46] seeks targeted AE located at a region with uniformly low loss value. *Smoothing-based approaches* smooth gradients by averaging gradients from multiple datapoints around the current AE. Diverse Inputs Method (DIM) [47] averages the gradients of randomly resized and padded inputs to generate AE. Translation-invariant Attack (TIM) [48], Scale Invariance Attack (SIM) [2], Smoothed Gradient Attack (SGM) [49], and Admix Attack (Admix) [50] also fall into this category. *Attention-based approaches* modify the important features in attention maps, motivated by the observation that different deep networks classify the same image based on similar important features. For instance, Attention guided Transfer Attack (ATA) [51] uses the gradients of an objective function w.r.t. neuron outputs to derive an attention map and seek AE that maximizes the difference between its attention map and the corresponding benign sample's map. Similar approaches include Jacobian based Saliency Map Attack (JSMA) [52], Attack on Attention (AoA) [53] and Activation attack (AA) [54]. *Ensemble-based approaches* take advantage of an ensemble of surrogate models with the belief that if an AE can attack multiple models, then it is more likely to transfer to other models as well. For instance, [55] proposes to generate AE on an ensemble of models with different architectures. Large Geometric Vicinity (LGV) [56] collects multiple checkpoints along the training trajectory, on which the attack

was performed on an ensemble of these models. [57] develops an ensemble attack from a Bayesian formulation which samples multiple models from the posterior distribution of parameter space.

1.2.4. Loss Landscape of DNNs. Understanding the loss landscape of deep neural networks (DNNs) plays a crucial role in optimizing model performance and generalization. The loss landscape, characterized by its complex geometry and optimization dynamics, influences the behavior and efficacy of optimization algorithms during training. Recent research into loss surface geometry underscores the strong correlation between generalization and the flatness of minima reached by DNN parameters. Among various mathematical definitions of flatness, including ϵ-sharpness [58], PAC-Bayes measure [59], Fisher Rao Norm [60], and entropy measures [61, 62], notable ones include Hessian-based metrics like Frobenius norm [63, 64], trace of the Hessian [65], largest eigenvalue of the Hessian [66], and effective dimensionality of the Hessian [67].

1.3. ORGANIZATION OF THE DISSERTATION

In section 2, from the data perspective, we propose a new method of crafting transferable AE which consists of two techniques: *elastic momentum* (EM) and *random erasure* (RE). This section is based on the work [68].

In section 3, we explore from optimization perspective and propose an approach, *gradient norm penalty (GNP)* by penalizing the input gradient norm to identify AE within flat regions of the loss landscape. This section is based on the work [69].

In section 4, we investigate from the model perspective and propose a novel strategy centered on transforming surrogate models by Lipschitz regularization. This section is based on the work [70].

In section 5, we introduce the normalized Hessian trace, a metric capable of accurately and consistently characterizing the curvature of loss landscapes. Leveraging this metric, we propose CR-SAM, a novel optimization technique that integrates curvature regularization into the Sharpness-Aware Minimization (SAM) optimizer to enhance the generalization of DNNs This section is based on the work [71].

Section 6 concludes this dissertation and presents the limitations and future work.

2. IMPROVING ADVERSARIAL TRANSFERABILITY FROM DATA PERSPECTIVE: ELASTIC MOMENTUM AND RANDOM ERASURE

2.1. INTRODUCTION

Deep Neural Networks (DNNs) have made resounding success in computer vision tasks. However, they are vulnerable to *adversarial examples* (AE), which are data samples (typically images) that are perturbed by human-imperceptible noises yet result in misclassifications. This can cause serious safety and security consequences in applications such as autonomous driving and medical diagnosis. The *transferability* of AE is an active research area [1, 2, 47, 48, 55, 72, 73, 74, 75, 76, 77] that studies how well an AE created to attack (fool) a "source" model can successfully fool other "target" models as well. The rationale of studying this is that (1) from an attacker's perspective, good transferability implies that one can launch *black-box attacks* on target models (without knowing their internal structure, algorithmic details, or parameters); (2) from a defender's perspective, studying it provides insight into understanding the failure and vulnerability of DNNs and how to design DNNs that are robust to AE.

The techniques proposed in the literature to improve the transferability of AE include gradient or momentum based methods [1, 2, 72, 73], ensemble methods [55, 74], image transformations based methods [2, 47, 48, 75], and network architecture alterations [76, 77]. A major issue of these techniques attempt to address is that AE created on a source model (in order to attack it) can be easily trapped into the exclusive blind spots of the source model and can hardly generalize to other (target) models; in other words, this can be viewed as an problem of AE *overfitting*.

In this section, we propose a new method of crafting AE and thereby improving their transferability. This method consists of two techniques: *elastic momentum* (EM) and *random erasure* (RE). We first introduce EM into the AE generation process to compute gradients in a much expedited manner insofar as the training will converge earlier than

reaching the overfitting region. We also propose to incorporate RE, which is a data augmentation technique, into the AE crafting procedure for the first time. The contributions of this section are summarized as follows:

- We introduce a new black-box approach of crafting transferable AE by proposing EM and a new usage of RE. EM generalizes the conventional momentum and the Nesterov's momentum methods by computing gradients over a flexible look-ahead horizon, and RE increases the diversity of adversarial perturbations and helps stabilize gradient fluctuations.
- Besides transferability, our proposed method is very flexible in that it can be applied to any existing gradient-based attacks to enhance their effectiveness.
- Through extensive evaluation with 5 recent baseline methods, 7 target deep learning models, and 9 advanced defense mechanisms, we demonstrate the superior transferability of our proposed black-box attack approach.

2.2. PROPOSED METHOD

In this section, we introduce our proposed method, which incorporates Elastic Momentum and Random Erasure in the generation of transferable adversarial examples.

2.2.1. Elastic Momentum. We make two key observations. First, the main reason why integrating momentum benefits AE computation is because the momentum essentially combines several steps of (potentially discounted) gradients together to help stablize gradient descent and obtain a more robust direction of convergence. Second, the reason why Nesterov's accelerated gradient can benefit it even further is because it computes the gradients based on an estimated next-step AE, rather than the last-step AE, which speeds up the training.

Thus, following the work [78], our basic idea is as follows. First, generalize the prediction of next-step AE, by allocating a *flexible look-ahead horizon* for computing an estimated future AE. Next, compute the gradient using that future AE to obtain a more *far-sighted* momentum, which accelerates the convergence (with reduced number of iterations) and thereby prevents overfitting.

Formally, an AE x^{adv} is computed iteratively as follows:

$$x_t^{em} = x_t^{adv} + \alpha \cdot \sigma \cdot g_t, \tag{2.1}$$

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_{x_t^{em}} J(x_t^{em}, y; \theta)}{\|\nabla_{x_t^{em}} J(x_t^{em}, y; \theta)\|_1}, \tag{2.2}$$

$$x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \text{sign}(g_{t+1}). \tag{2.3}$$

The momentum term g accumulates previous gradients with a decay factor μ, while the gradient is not computed based on the current AE x_t^{adv} but a future AE x_t^{em} estimated over a look-ahead horizon. The parameter σ is critical: although μ has to be a value extremely close to 1, as experimentally shown by [1], σ is independent of μ (as opposed to NI-FGSM) and could take a value much larger than 1, which essentially means that we can use g_t to approximate g_{t+1}, g_{t+2}, ... and tune the length of this look-ahead horizon to achieve the best transferability. For this reason, we call the momentum term g an elastic momentum (EM). Figure 2.1 illustrates our method EM as compared to NI-FGSM.

Our approach also generalizes MI-FGSM and NI-FGSM which can be viewed as special cases of ours: When $\sigma = 0$, we obtain the momentum iterative method MI-FGSM; when $\sigma = \mu$, we obtain Nesterov's momentum method NI-FGSM. Note, however, that we typically do *not* use these σ values in order to achieve acceleration and thus better performance. In fact, our method gives us flexibility to control the converging process via σ, in order to reach a local optimum before hitting the overfitting region, thereby obtaining better AE transferability.

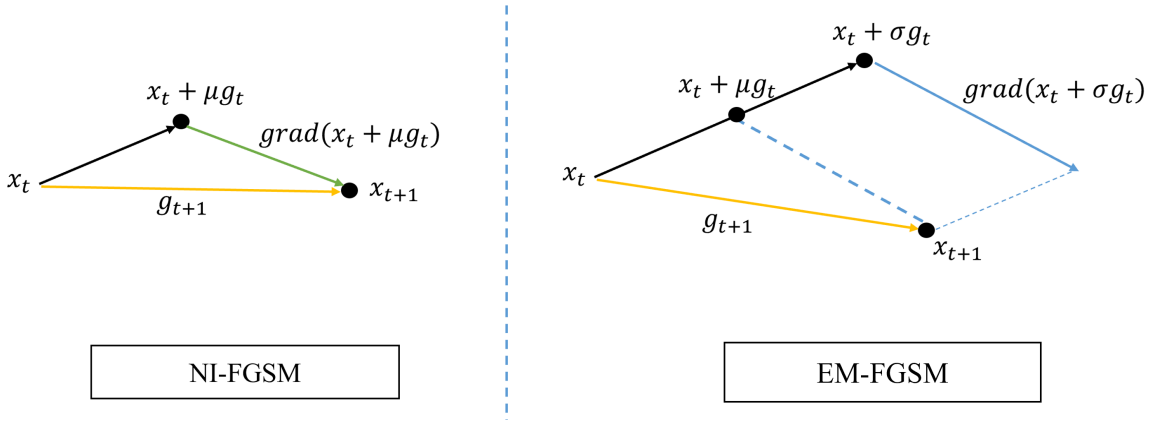


Figure 2.1. Illustration of EM as compared to NI-FGSM. when $\sigma = \mu$, we obtain Nesterov's momentum method NI-FGSM as a special case of EM-FGSM.

2.2.2. Random Erasure. Previous work [47] has demonstrated that random transformations of input images such as random resizing and random padding could boost the transferability of adversarial examples. However, what specific type of transformation is better remains an open question. In our work, we hypothesize that partial occlusion would make the resulting AE more transferable, and the rationale is as follows. A classification model usually examines different regions of an image to recognize its category, which is why a white-box attack could achieve near 100% attack success rate whereas black-box transferred AE are much less likely to fool target models since those models tend to ignore the adversarial regions. However, when an image is partially occluded, a model will classify it based on the overall object structure. Thus, if we use occluded adversarial images during AE generation, the AE generation process will make the non-occluded region of the object structure adversarial, and as a result, the generated AE will be more transferable and more likely to fool other target models. Similar techniques are also proposed in [79, 80] as a generic data augmentation technique for deep learning to address data insufficiency which bring benefits to the task of image classification, object detection and person re-identification. In this section, however, we apply RE to AE generation which has never been explored before. In addition, we identify that RE is the most suitable candidate for AE

transferability through our comparison with many other data augmentation techniques such as translation, scaling, rotation, resizing, padding, weighting, and even a nearest neighbor method that we created on our own.

Given an image I with width W and height H, we apply RE by randomly selecting a rectangle region I_e in I and removes the pixels in the region I_e. This region is determined as follows. Denoting by S_e the area of the region I_e, we randomly generate an erasure ratio s in the range $[0, s_h]$ where $s_h < 1$, and use $s = \frac{S_e}{S}$ to determine the value of S_e, where S is the area of the input image I, i.e., $S = W \times H$. Now, denote the aspect ratio of I_e by r_e. The height and width of I_e are therefore determined by $H_e = \sqrt{S_e \times r_e}$ and $W_e = \sqrt{\frac{S_e}{r_e}}$, respectively. To determine the location of I_e, we randomly pick a point $\mathcal{P} = (x_e, y_e) \in I$, until $x_e + W_e \leq W$ and $y_e + H_e \leq H$, upon which we finalize the coordinates of the erasure region $I_e = (x_e, y_e, x_e + W_e, y_e + H_e)$. An example is given in Figure 2.2.

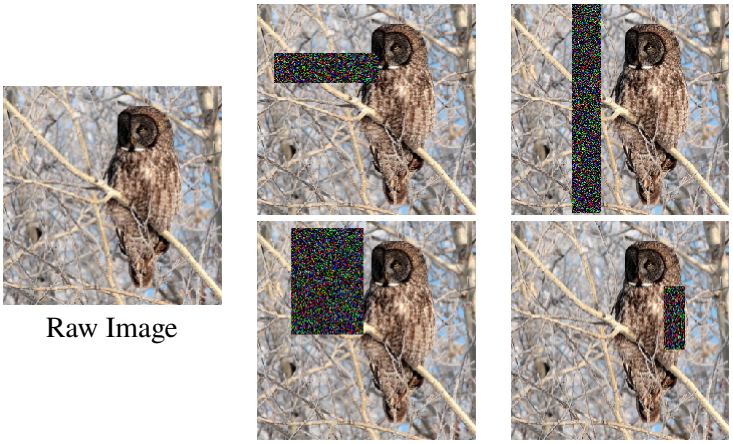


Figure 2.2. Applying RE to a raw image to generate four images with partial occlusions of varying sizes and positions.

To remove the pixels in the region I_e, there are three typical choices, namely using 0s, 1s and random noise, to fill the region. Our experiments show that they do not make a notable difference in performance. Hence, we adopt random noise in this section.

Now, we reformulate the objective function by incorporating RE, as

$$\begin{aligned} &\arg\max_{x^{adv}} \frac{1}{m}\sum_{i=0}^{m} \ell(RE_i(x^{adv}), y; \theta), \\ &\text{s.t.} \left\|x^{adv} - x\right\|_\infty \leq \epsilon, \end{aligned} \tag{2.4}$$

where m is the number of erasure copies, and $i = 0$ represents the input image without erasure.

Thus, our AE crafting process, integrated RE, is as follows. At each iteration t, with probability p, we apply RE to the input image x_t^{adv} to generate a collection of m erased images, and compute their losses and the average gradient $\frac{1}{m}\sum_{i=0}^{m} \nabla_x \ell(RE_i(x^{adv}), y; \theta)$, which will be used to compute the momentum g. With probability $1 - p$, we keep the input image x^{adv} intact.

To incorporate RE into EM, however, the above x^{adv} needs to be replaced by x^{em} defined by (2.1). Therefore, our final proposed method is formulated as

$$x_t^{em} = x_t^{adv} + \alpha \cdot \sigma \cdot g_t, \tag{2.5}$$

$$g_{t+1} = \mu \cdot g_t + \frac{\frac{1}{m}\sum_{i=0}^{m} \nabla_x \ell(RE_i(x_t^{em}), y; \theta)}{\|\frac{1}{m}\sum_{i=0}^{m} \nabla_x \ell(RE_i(x_t^{em}), y; \theta)\|_1}, \tag{2.6}$$

$$x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \text{sign}(g_{t+1}). \tag{2.7}$$

Algorithm 1 summarizes our proposed method, where $RE(x; p)$ denotes that we apply RE with probability p.

2.3. EXPERIMENTS

In this section, we illustrate our experiments, detailing the experimental settings and presenting the results obtained that validate the effectiveness of our proposed methods.

Algorithm 1 Proposed Method: EM-RE-FGSM

Input: A clean example x with ground-truth label y; a classifier f with loss function ℓ;
Input: Perturbation size ϵ; maximum iterations T; decay factor μ; look-ahead parameter σ; number of random erasure copies m; random erasure probability p.
Output: An adversarial example x^{adv}

$\alpha = \epsilon/T$
$g_0 = 0; x_0^{adv} = x$
for $t = 0$ to $T - 1$ **do**
    Compute $x_t^{em} = x_t^{adv} + \alpha \cdot \sigma \cdot g_t$
    $g = 0$
    for $i = 0$ to $m - 1$ **do**
        Compute gradient $\nabla_x \ell(RE_i(x_t^{em}; p), y; \theta)$
        Update $g = g + \nabla_x \ell(RE_i(x_t^{em}; p), y; \theta)$
    end for
    Average momentum as $g = \frac{g}{m}$
    Update g_{t+1} as $g_{t+1} = \mu \cdot g_t + \frac{g}{\|g\|_1}$
    Update $x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \text{sign}(g_{t+1})$
end for
return $x^{adv} = x_T^{adv}$

2.3.1. Experiment Setup. *Dataset.* We use an image dataset [5] which is a curated portion of ImageNet and is widely used such as by [2, 73]. This dataset randomly selects one clean and correctly classified images from each of the 1,000 categories of the ILSVRC 2012 validation dataset, and thus contains 1,000 good images with each of the size $299 \times 299 \times 3$.

Models to attack. We first consider four widely used state-of-the-art DNNs, namely Inception-v3 (Inc-v3) [3], Inception-v4 (Inc-v4)[81], Inception-Resnet-v2 (IncRes-v2) [81], and Resnet-v2-101 (Res-101) [82]. In addition, to increase the difficulty level, we also include three *adversarially trained* DNNs (and thus are more robust to AE), namely Inc-v3$_{ens3}$, Inc-v3$_{ens4}$ and IncRes-v2$_{ens1}$ [83]. The first two models are Inc-v3 trained on AE generated from an ensemble of 3 and 4 other pretrained DNNs, respectively, and the last is IncRes-v2 trained on AE generated from a single pretrained DNN (it is still called an "ensemble" in [83] so we have adopted the same naming convention).

Defenses to attack. To further increase the difficulty level, we also consider *defense mechanisms* in our evaluation. As pointed out by [42], many existing attacks underperform or even fail when target models are armed with defense mechanisms. Therefore, we select nine state-of-the-art advanced defenses: the top-3 winners in the NeurIPS defense strategy competition and 6 recently proposed defense methods. The first group consists of HGD (rank-1) [40], R&P (rank-2) [36], and NIPS-r3 (rank-3), and the second group consits of Bit-Red [38], JPEG [39], FD [84], ComDefend [85], RS [42] and NRP [86]. These 9 defense methods have been integrated into their respective DNNs.

Baseline AE-generation methods. We compare our method with five recently proposed attack methods, namely MI-FGSM [1], NI-FGSM [2], Diverse Inputs Method (DIM) [47], Translation-Invariant attack Method (TIM) [48], and Scale-Invariant attack Method (SIM) [2]. The first two are momentum-based attacks and the other three are image-transformation based attacks.

Versatile as a "plug-in". As mentioned, our method can be applied to any gradient-based attack method to form a new, stronger attack. We demonstrate this by integrating our method with DIM, TIM, and SIM, respectively, as well as all of them three combined together, to obtain two more attacks and include them in our evaluation as well.

Attack setup. We normalize image pixel values in $[-1, 1]$, and set the number of iterations $T = 10$, the maximum perturbation $\epsilon = 16/255$ as in [1]. For parameters related to EM, we set the decay factor $\mu = 1$ following [1] and the look-ahead parameter $\sigma = 2$ as indicated by our ablation study. For parameters related to RE, we set $s_h = 0.4$ and $r_e = 0.3$ following [79], the number of erasure copies $m = 5$ and probability $p = 0.5$. We use *attack success rate* (ASR) as our evaluation metric, which is the misclassification rate of a classifier when test samples are AE (we have verified that all the benign images are classified correctly in all the cases).

Table 2.1. The attack success rates (ASR) (%) on seven target models in the single-source-model setting, using EM alone. The AE are generated using a single source model Inc-v3, Inc-v4, IncRes-v2, or Res-101. '*' indicates white-box attack.

Source model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens1}$
Inc-v3	MI-FGSM	**100.0***	43.6	42.4	35.7	13.1	12.8	**6.2**
	NI-FGSM	**100.0***	51.7	50.3	41.3	**13.5**	**13.2**	6.0
	EM-FGSM	**100.0***	**55.0**	**52.7**	**44.6**	11.4	11.5	5.5
Inc-v4	MI-FGSM	56.3	99.7*	46.6	41.0	**16.3**	**14.8**	**7.5**
	NI-FGSM	63.1	**100.0***	51.8	45.8	15.4	13.6	6.7
	EM-FGSM	**66.9**	**100.0***	**54.4**	**47.6**	14.7	12.4	6.8
IncRes-v2	MI-FGSM	60.7	51.1	97.9*	46.8	**21.2**	**16.0**	**11.9**
	NI-FGSM	62.8	54.7	99.1*	46.0	20.0	15.1	9.6
	EM-FGSM	**65.2**	**56.2**	**99.2***	**48.7**	18.6	13.1	7.8
Res-101	MI-FGSM	58.1	51.6	50.5	99.3*	23.9	**21.5**	**12.7**
	NI-FGSM	65.6	58.3	57.0	**99.4***	**24.5**	21.4	11.7
	EM-FGSM	**65.7**	**60.9**	**61.1**	99.3*	20.8	17.6	10.0

2.3.2. Experimental Results. In this section, we present our experimental results, which include evaluations using a single source model, an ensemble of source models, attacks on advanced defense mechanisms, and an ablation study on hyper-parameters.

2.3.2.1. Single source model. In this section, we evaluate the case that AE are trained on a single source model and then used to attack multiple target models. We test four source models: Inc-v3, Inv-v4, IncRes-v2, and Res-101, and the target models are these four as well as the three ensemble models, i.e., Inc-v3$_{ens3}$, Inc-v3$_{ens4}$ and IncRes-v2$_{ens1}$.

Using EM alone (without RE). We first evaluate the EM approach only, without using RE. The results are presented in Table 2.1. First, under white-box attacks (source model is also the target model), all the methods achieve close to 100% ASR as expected. These mean that, although our method focuses on improving black-box performance (due to transferability), we do not sacrifice any white-box performance either. Second, let us look at black-box attacks (target model is different from the source model), which are more important since they particularly reflect the *transferability* of AE. We see that EM achieves the higher ASR in about 60% of the cases while MI-FGSM and NI-FGSM perform the best in about 30% and 10% of the cases, respectively. Note that we have not activated RE

Raw Image MI-FGSM NI-FGSM EM-FGSM

Figure 2.3. Adversarial images crafted by MI-FGSM [1], NI-FGSM [2] and our EM approach on the Inc-v3 model [3] with the maximum perturbation $\epsilon = 16/255$.

yet. Taking a closer look, one can observe that the cases where our proposed EM method outperforms MI-FGSM and NI-FGSM are normally trained models, and the cases in which it does not (but still keeps a comparable performance) are those three adversarially trained models. The reason behind this is that, for normally trained models, EM achieves better optimum in the constrained iterative steps and hence demonstrates better transferability; but on the other hand, the three ensemble adversarially trained models, i.e., Inc-v3$_{ens3}$, Inc-v3$_{ens4}$ and IncRes-v2$_{ens1}$, augmented their training data with AE crafted on other static pre-trained models, and hence were trained to resist transferable AE, making black-box attacks ineffective. Therefore, to achieve higher ASE against such adversarially trained models, we need to increase the diversity of perturbations in AE, which precisely motivated our introduction of our second technique, Ransom Erasure (RE). By combining with RE, our method achieves much higher ASR against ensemble adversarially trained models, as shown later. To offer a visual intuition, we also give some example AE images generated by all these methods, in Figure 2.3. It shows that all the adversarial images are very similar to the original raw image as perceived by human eyes.

Using RE alone (without EM). We then evaluate the RE approach only, without using EM. The results are presented in Table 2.2. The results indicate that in all the cases our proposed RE consistently outperforms DIM, TIM and SIM by a large margin, which means RE yields higher transferablity on all the black-box models while maintaining high

Table 2.2. The attack success rates (ASR) (%) on seven target models in the single-source-model setting, using RE alone. The AE are generated using a single source model Inc-v3, Inc-v4, IncRes-v2, or Res-101. '*' indicates white-box attack.

Source model	Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens1}$
Inc-v3	DIM	99.0*	64.3	60.9	53.2	19.9	18.3	9.3
	TIM	100.0*	48.8	43.6	39.5	24.8	21.3	13.2
	SIM	100.0*	69.4	67.3	62.7	32.5	30.7	17.3
	RE	**100.0***	**71.1**	**68.7**	**64.3**	**33.1**	**31.6**	**19.0**
Inc-v4	DIM	72.9	97.4*	65.1	56.5	20.2	21.1	11.6
	TIM	58.6	99.6*	46.5	42.3	26.2	23.4	17.2
	SIM	80.6	99.6*	74.2	68.8	47.8	44.8	29.1
	RE	**82.3**	**99.8***	**76.3**	**71.5**	**49.6**	**45.9**	**31.4**
IncRes-v2	DIM	70.1	63.4	93.5*	58.7	30.9	23.9	17.7
	TIM	62.2	55.4	97.4*	50.5	32.8	27.6	23.3
	SIM	84.7	81.1	99.0*	76.4	56.3	48.3	42.8
	RE	**86.2**	**83.3**	**99.4***	**78.7**	**59.1**	**50.6**	**46.2**
Res-101	DIM	75.8	69.5	70.0	98.0*	35.7	31.6	19.9
	TIM	59.3	52.1	51.8	99.3*	35.4	31.3	23.1
	SIM	75.2	68.9	69.0	99.7*	43.7	38.5	26.3
	RE	**78.2**	**71.5**	**72.8**	**99.8***	**45.2**	**39.8**	**28.7**

attack success rates on the white-box setting. For instance, if we craft adversarial examples on IncRes-v2 model where our white-box attack achieves 99.4% success rate, RE yields 78.7% ASR on Res-101 which is a black-box setting; in comparison, TIM only achieves an ASR of 97.4% and 50.5%, respectively, in the same two settings. This set of results validate the effectiveness of our proposed RE method.

2.3.2.2. Ensemble source model. Crafting AE on an ensemble of models has been shown to be effective to improve AE transferability [1, 55]. In this section, we evaluate the performance over an ensemble model of four: Inc-v3, Inc-v4, IncRes-v2 and Res-101, by averaging their logit outputs when calculating the gradients [1]. The results of using EM alone are summarized in Table 2.3. We observe that EM achieves the highest ASR in all the black-box attack scenarios.

Next, we apply both EM and RE to DIM, TIM and SIM, respectively, to form three new models. In addition, we create a new attack *Composite* by combining DIM, TIM and SIM together which forms the strongest baseline. On top of that, we apply EM and RE to

Table 2.3. ASR (%) on seven target models in the ensemble-source-model setting, using EM alone. The source model is the ensemble of {Inc-v3, Inc-v4, IncRes-v2, Res-101}. '*' indicates white-box attack.

Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens1}$
MI-FGSM	99.9*	98.2*	95.3*	99.9*	39.4	35.3	24.2
NI-FGSM	99.8*	99.8*	**98.9***	99.8*	41.0	33.5	23.1
EM-FGSM	**99.9***	**99.8***	98.4*	**99.9***	**43.6**	**36.1**	**25.9**

Table 2.4. ASR (%) on seven target models in the ensemble-source-model setting, using both EM and RE. The source model is the ensemble of {Inc-v3, Inc-v4, IncRes-v2, Res-101}. Composite model is the combination of DIM, TIM, and SIM. '*' indicates white-box attack.

Attack	Inc-v3	Inc-v4	IncRes-v2	Res-101	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens1}$
DIM	99.4*	97.4*	94.9*	99.8*	58.1	51.1	34.9
EM-RE-DIM	**99.7***	**99.1***	**97.5***	**99.8***	**64.3**	**59.7**	**41.6**
TIM	99.7*	98.9*	97.7*	99.9*	62.2	56.8	48.0
EM-RE-TIM	**99.9***	**99.3***	**98.9***	**100.0***	**68.9**	**64.1**	**56.4**
SIM	99.7*	99.0*	97.6*	**100.0***	78.8	73.9	59.5
EM-RE-SIM	**99.8***	**99.3***	**98.4***	**100.0***	**84.3**	**79.5**	**66.8**
Composite	99.6*	98.9*	97.8*	99.7*	91.1	90.3	86.8
EM-RE-Composite	**99.8***	**99.3***	**98.4***	**99.8***	**92.3**	**91.6**	**88.6**

Composite to obtain an enhanced attack using our method. We evaluate these 8 attacks and report their performance in Table 2.4. It shows that our proposed method again yields the best ASR in all the white-box and black-box attacks (4×7 cases), outperforming all the baselines by up to 17.5%.

2.3.2.3. Attacking advanced defense mechanisms. Although our proposed method exhibits superior performance on both regularly and adversarially trained deep models, there is still a question left as to whether it will perform well against models that are protected by more sophisticated mechanisms. As pointed out by [42], many existing attacks underperform or even fail when target models have additional defense mechanisms. Motivated by this, we select 9 advanced defense mechanisms to attack, as described in our experiment setup, for the purpose of a more thorough evaluation.

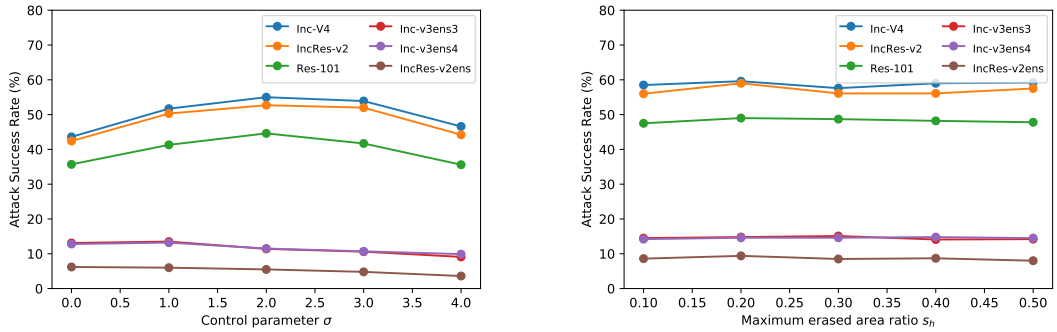
Table 2.5. ASR (%) on 9 advanced defense mechanisms. *Composite* refers to the combination of DIM, TIM, and SIM.

Source	Attack	HGD	R&P	NIPS-r3	Bit-Red	JPEG	FD	ComDefend	RS	NRP	Average
Inc-v3	MI-Composite	56.6	44.9	52.5	36.2	77.3	60.0	80.1	40.3	29.3	53.0
	NI-Composite	50.4	39.4	47.4	34.3	76.0	58.6	77.7	36.9	24.8	49.5
	EM-RE-Composite	**59.6**	**48.3**	**55.9**	**39.6**	**81.1**	**65.5**	**82.3**	**45.4**	**33.1**	**56.8**
Ensemble	MI-Composite	91.0	87.7	89.0	75.9	94.2	88.8	95.1	68.1	76.1	85.1
	NI-Composite	91.3	85.6	89.0	72.3	95.9	89.5	95.4	63.2	69.5	83.5
	EM-RE-Composite	**92.9**	**89.6**	**91.8**	**79.3**	**96.9**	**92.4**	**96.4**	**74.3**	**80.1**	**88.2**

We use Inc-v3 and the ensemble of {Inc-v3, Inc-v4, IncRes-v2, Res-101} as the source models to train AE, and attack the above 9 advanced defense mechanisms. We further create more baseline attacks by combining MI and NI respectively with the *Composite* (MI and NI do have this similar "plug-in" kind of advantage as our method, but most other methods in the literature do not have). The results are given in Table 2.5. In this case, there is no white-box attack and all attacks are black-box. We observe that our proposed EM-RE approach is the best performer in all the scenarios, with a substantial winning margin, up to 25.4%.

2.3.2.4. Ablation study on hyper-parameters. We also conduct ablation experiments to study the impact of the hyper-parameters on the performance of our approach. Two key parameters are σ which determines the look-ahead horizon in EM, and s_h which determines the maximum erasure area in RE. In this ablation study, the source model is chosen to be Inc-v3 and the generated AE are then used to attack the other six models, and hence all the attacks are black-box.

The results as shown in Fig. 2.4, where we vary σ from 0.0 to 4.0 with step size 1.0, and vary s_h from 0.1 to 0.5 with step size 0.1. The perturbation $\epsilon = 16/255$ and the number of iterations $T = 10$. The results indicate that the best ASR is achieved at $\sigma = 2.0$, yet is *insensitive* to the choice of s_h (which is a good thing since it implies robustness of our erasure). Therefore, we have chosen $\sigma = 2.0$ and $s_h = 0.4$ in our experiments.



(a) Impact of σ on ASR.

(b) Impact of s_h on ASR.

Figure 2.4. Ablation study on σ (look-ahead horizon) and s_h (max. erasure area). The source model is Inc-v3 and the 6 target models under attack are indicated by the legend.

2.4. SUMMARY

In this section, we propose a new black-box approach of crafting transferable adversarial examples (AE) to attack deep learning based image classifiers. As such deep models are increasingly being deployed in autonomous driving, medical diagnosis, and many other computer vision applications, studying this topic plays an important role in deepening our understanding of AI security. Our proposed method consists of a gradient-based elastic momentum (EM) technique, and a random erasure (RE) data augmentation technique. EM introduces a flexible look-ahead horizon to estimate future momentum during AE computation, which speeds up the process of finding local optima and thus prevents hitting the overfitting region. RE creates an ensemble of transformed images that increases the diversity of perturbations and helps stablize gradient updates, which optimize the adversarial perturbations. We have performed extensive experiments to evaluate our proposed EM-RE method by attacking 7 modern deep learning classifiers and 9 advanced defense mechanisms, in comparison with 5 recently proposed baseline methods (and an additional Composite method). The results demonstrate superior transferability of the adversarial examples generated by our proposed method for black-box attacks.

3. IMPROVING ADVERSARIAL TRANSFERABILITY FROM OPTIMIZATION PERSPECTIVE: GRADIENT NOEM PENALTY

3.1. INTRODUCTION

Deep Neural Networks (DNNs) are the workhorse of a broad variety of computer vision tasks but are vulnerable to adversarial examples (AE), which are data samples (typically images) that are perturbed by human-imperceptible noises yet result in odd misclassifications. This lack of adversarial robustness curtails and often even prevents deep learning models from being deployed in security or safety critical domains such as healthcare, neuroscience, finance, and self-driving cars, to name a few.

Adversarial examples are commonly studied under two settings, white-box and black-box attacks. In the white-box setting, adversaries have full knowledge of victim models, including model structures, parameters and weights, and loss functions used to train the models. Therefore, they can directly obtain the gradients of the victim models and seek adversarial examples by misleading the loss function toward incorrect predictions. White-box attacks are important for evaluating and developing robust models and serve as the backend method for many black-box attacks, but is limited in use due to its requirement of having to know the internal details of target models. In the black-box setting, adversaries do not need specific knowledge about victim models other than their external properties (type of input and output). Two types of approaches, query-based and transfer-based, are commonly studied for black-box attacks. The query-based approach attempts to estimate the gradients of a victim model by querying it with a large number of input samples and inspecting the outputs. Due to the large number of queries, it can be easily detected and defended. The transfer-based approach uses *surrogate* models to generate *transferable* AE which can attack a range of models instead of a single victim model. Hence it is a more attractive approach to black-box attacks.

This section takes the second approach and focuses on designing a new and effective method to improve the *transferability* of AE. Several directions for boosting adversarial transferability have appeared. Dong et al. [1] proposed momentum based methods. Attention-guided transfer attack (ATA) [51] uses attention maps to identify common features for attacking. Diverse Input Method (DIM) [47] calculates the average gradients of augmented images. [55] generates transferable AE using an ensemble of multiple models.

Despite the efforts of previous works, there still exists a large gap of attack success rate between the transfer-based setting and the ideal white-box setting. In this section, we propose a novel method to boost adversarial transferability from an *optimization* perspective. Inspired by the concept of "flat minima" in the optimization theory [87] which improves the generalization of DNNs, we seek to generate AE that lie in flat regions where the input gradient norm is small, so as to "generalize" to other victim models that AE are *not* generated on. In a nutshell, this section makes the following contributions:

- We present a novel transfer-based black-box attack that targets adversarial examples (AE) from a novel perspective, seeking to locate them in flat regions of the loss landscape by penalizing the input gradient norm.

- Our method, the input gradient norm penalty (GNP), enhances adversarial transferability across a broad spectrum of deep networks. Through extensive experiments, we have shown that GNP consistently outperforms existing methods, establishing a new benchmark for transfer-based black-box attacks.

- Furthermore, we demonstrate that GNP can be effortlessly integrated with existing transfer-based attacks, yielding even better performance. This indicates not only the superior effectiveness of our method but also its exceptional flexibility and compatibility with a wide range of existing attack techniques.

3.2. METHOD

Given a classification model $f(x) : x \in \mathcal{X} \rightarrow y \in \mathcal{Y}$ that outputs a label y as the prediction for an input x, we aim to craft an adversarial example x^* which is visually indistinguishable from x but will be misclassified by the classifier, i.e., $f(x^*) \neq y$. The generation of AE can be formulated as the following optimization problem:

$$\arg\max_{x^*} \ell(x^*, y), \quad \text{s.t. } \|x^* - x\|_p \leq \epsilon, \tag{3.1}$$

where the loss function $\ell(\cdot, \cdot)$ is often the cross-entropy loss, and the l_p-norm measures the discrepancy between x and x^*. In this work, we use $p = \infty$ which is commonly adopted in the literature. Optimizing Eq. (4.1) needs to calculate the gradient of the loss function, but this is not feasible in the black-box setting. Therefore, we aim to create transferable AE on a source model yet can attack many other target models.

3.2.1. Motivation. We develop a new method to boost adversarial transferability from a perspective inspired by "flat optima" in optimization theory. See Figure 3.1. If an AE is located at a sharp local maximum, it will be sensitive to the difference of decision boundaries between the source model and target models. In contrast, if it is located at a flat maximum region, it is much more likely to result in a similar high loss on other models (which is desired).

Thus, we seek to generate AE in flat regions. To this end, we introduce a *gradient norm penalty* (GNP) term into the loss function, which penalizes the gradient norm of the loss function with respect to input. The reason is that flat regions are characterized by small gradient norms, hence penalizing the gradient norm will encourage the optimizer to find an AE that lies in a flat region. We thus enhance the adversarial transferability since a minor shift of decision boundary will not significantly change the loss value (prior work has shown that different networks often share similar decision boundaries).

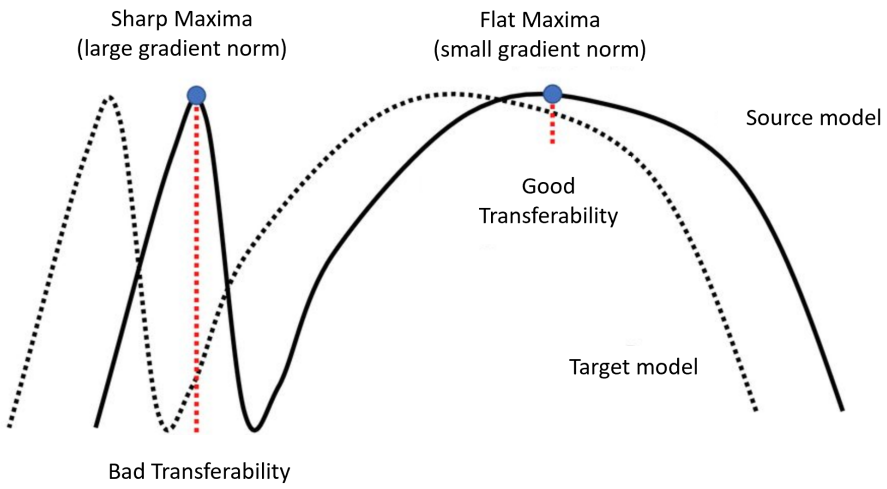


Figure 3.1. The loss function landscape: sharpness vs. flatness, which leads to different levels of transferability.

3.2.2. GNP Attack. As explained in method, we aim to guide the loss function optimization process to move into a flat local optimal region. To this end, we introduce GNP to penalize large gradient norm, as

$$L(x, y) = \ell(x, y) - \lambda \left\|\nabla_x \ell(x, y)\right\|_2 \tag{3.2}$$

where $\ell(\cdot)$ is the original loss function of the source model, and the regularization term is our GNP, which encourages small gradient norm when finding local maxima.

For gradient based attacks (e.g., FGSM, I-FGSM, MI-FGSM, etc.), we need to calculate the gradient of the new loss (3.2). To simplify notation, we omit y in the loss function since we are calculating gradient with respect to x. Using the chain rule, we have

$$\nabla_x L(x) = \nabla_x \ell(x) - \lambda \nabla_x^2 \ell(x) \frac{\nabla_x \ell(x)}{\|\nabla_x \ell(x)\|} \tag{3.3}$$

This equation involves the calculation of Hessian matrix $H = \nabla_x^2 \ell(x)$. This is often infeasible because of the curse of dimensionality (such a Hessian matrix in DNNs tends to be too large due to the often large input dimension). Therefore, we take the first-order Taylor expansion

Algorithm 2 I-FGSM+GNP

Input: A clean sample x with ground-truth label y; source model $f(\cdot)$ with loss function $\ell(\cdot)$;
Input: Perturbation size ϵ; maximum iterations T; step length r; regularization coefficient β
Output: A transferable AE x^{adv}

$\alpha = \epsilon/T$; $g_0 = 0$; $x_0^{adv} = x$
for $t = 0$ to $T - 1$ **do**
    $g_1 = \nabla_x \ell(x)$
    Compute $r\frac{\nabla_x \ell(x)}{||\nabla_x \ell(x)||}$
    $g_2 = \nabla_x \ell\left(x + r\frac{\nabla_x \ell(x)}{||\nabla_x \ell(x)||}\right)$
    $g_t = (1+\beta)g_1 - \beta g_2$
    Update $x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \text{sign}(g_t)$
end for
return $x^{adv} = x_T^{adv}$

together with the finite difference method (FDM) to approximate the following gradient:

$$\nabla_x L(x + r\Delta x) \approx \nabla_x \ell(x) + Hr\Delta x \tag{3.4}$$

where $\Delta x = \frac{\nabla_x \ell(x)}{\|\nabla_x \ell(x)\|}$, and r is the step length to control the neighborhood size. Thus we obtain the regularization term of (3.3) as:

$$H\frac{\nabla_x \ell(x)}{\|\nabla_x \ell(x)\|} \approx \frac{\nabla_x \ell\left(x + r\frac{\nabla_x \ell(x)}{\|\nabla_x \ell(x)\|}\right) - \nabla_x \ell(x)}{r} \tag{3.5}$$

Inserting (3.5) back into (3.3), we obtain the gradient of the regularized loss function as:

$$\nabla_x L(x) = (1+\beta)\nabla_x \ell(x) - \beta \nabla_x \ell\left(x + r\frac{\nabla_x \ell(x)}{\|\nabla_x \ell(x)\|}\right) \tag{3.6}$$

where $\beta = \frac{\lambda}{r}$ is the regularization coefficient. We summarize the algorithm of how GNP is integrated into I-FGSM in Algorithm 3, but I-FGSM can be replaced by any gradient based attack.

Table 3.1. Attack success rates when GNP is integrated with baselines to attack 11 target models ('*' denotes white-box attack).

Method	ϵ	ResNet50*	VGG19	ResNet152	Inc v3	DenseNet	MobileNet	SENet	ResNeXt	WRN	PNASNet	MNASNet	Average
I-FGSM	16/255	100.00%	61.50%	52.82%	30.86%	57.36%	58.92%	38.12%	48.88%	48.92%	28.92%	57.20%	48.35%
	8/255	100.00%	38.90%	29.36%	15.36%	34.86%	37.66%	17.76%	26.30%	26.26%	13.04%	35.08%	27.46%
	4/255	100.00%	18.86%	11.28%	6.66%	15.44%	18.36%	5.72%	9.58%	9.98%	4.14%	17.02%	11.70%
I-FGSM +GNP	16/255	100.00%	**75.96%**	**68.89%**	**48.23%**	**73.68%**	**74.05%**	**55.46%**	**62.36%**	**70.60%**	**45.06%**	**76.98%**	**67.12%**
	8/255	99.96%	**68.56%**	**60.65%**	**38.58%**	**62.05%**	**63.23%**	**43.69%**	**50.36%**	**59.32%**	**33.62%**	**60.28%**	**53.97%**
	4/255	99.98%	**25.96%**	**22.35%**	**15.86%**	**26.89%**	**28.66%**	**15.62%**	**21.93%**	**23.06%**	**13.69%**	**30.21%**	**22.38%**
MI-FGSM	16/255	100.00%	73.01%	67.62%	47.51%	73.16%	72.42%	54.53%	61.78%	60.96%	44.10%	71.46%	62.75%
	8/255	100.00%	52.50%	41.52%	25.56%	47.25%	48.96%	28.06%	35.81%	37.56%	20.41%	47.62%	38.53%
	4/255	99.94%	25.74%	16.68%	9.95%	22.54%	24.89%	9.56%	14.20%	15.38%	7.23%	23.27%	16.94%
MI-FGSM +GNP	16/255	100%	**89.65%**	**83.69%**	**65.86%**	**87.96%**	**90.06%**	**69.74%**	**79.12%**	**77.36%**	**58.60%**	**88.25%**	**79.04%**
	8/255	99.91%	**65.28%**	**55.63%**	**39.69%**	**61.42%**	**63.26%**	**42.03%**	**48.65%**	**51.07%**	**35.03%**	**58.93%**	**52.20%**
	4/255	100.00%	**39.62%**	**33.25%**	**15.62%**	**37.96%**	**40.04%**	**20.35%**	**30.27%**	**30.05%**	**15.23%**	**37.92%**	**30.03%**

3.3. EXPERIMENTS

In this section, we detail our experiments, describing the experimental settings and presenting the results that validate the effectiveness of our proposed methods.

3.3.1. Experiment Setup. In this section, we provide an overview of our experiment setup, including details about the dataset, the models employed, and the implementation specifics.

Dataset and Models. We randomly sample 5,000 test images that can be correctly classified by all the models, from the ImageNet [5] validation set. We consider 11 SOTA DNN-based image classifiers: ResNet50 [88], VGG-19 [89], ResNet-152 [88], Inc v3 [90], DenseNet [91], MobileNet v2 [92], SENet [93], ResNeXt [94], WRN [95], PNASNet [96], and MNASNet [97]. Following the work in [98], we choose ResNet50 as the source model and the remining 10 models as target models.

Implementation Details. In experiments, the pixel values of all images are scaled to [0, 1]. The adversarial perturbation is restricted by 3 scales $\epsilon = 4/255, 8/255, 16/255$. The step length is set as $r = 0.01$ and regularization coefficient $\beta = 0.8$, we run 100 iterations for all attacks and evaluate model misclassification as attack success rate.

3.3.2. Experimental Results. In this section, we present our experimental results, which include evaluations of integration with baseline attacks, integration with existing transfer-based attacks and attacking "secured" models.

3.3.2.1. Integration with baseline attacks. We first evaluate the performance of GNP by integrating it with baseline attacks including I-FGSM and MI-FGSM. The results are shown in Table 3.1. We use a pre-trained ResNet50 as the source model and evaluate the attack success rate (ASR) of the generated AE on a variety of target models under different scales of perturbation ϵ. GNP achieves significant and consistent improvement in all the cases. For instance, taking the average ASR of all the 10 target models under perturbation $\epsilon = 8/255$, GNP outperforms I-FGSM and MI-FGSM by 26.51% and 13.67%, respectively. In addition, the improvements of the attack success rates on a single model can be achieved by a large margin of 33.06%.

3.3.2.2. Integration with existing transfer-based attacks. Here we also evaluate the effectiveness of GNP when incorporated into other transfer-based attacks such as DIM [48] and TIM [47]. The results are given in Table 3.2 and show that DIM+GNP and TIM+GNP are clear winners over DIM and TIM alone, respectively. Specifically, DIM+GNP achieves an average success rate of 91.95% under $\epsilon = 16/255$ for the 10 target models, and TIM+GNP outperform TIM by a large margin of 16.28% under $\epsilon = 8/255$. We note that we only present the integration of GNP with two typical methods here, but our method also apply to other more powerful gradient-based attack methods.

3.3.2.3. Attacking "secured" models. For a more thorough evaluation, we also investigate how GNP will perform when attacking DNN models that have been *adversarially trained* (and hence are much harder to attack). We choose three such advanced defense methods to attack, namely, JPEG [39], R&P [36] and NRP [86]. In addition, we choose another three *ensemble* adversarially trained (AT) models, which are even harder than regular AT models, and attack them: Inc-v3$_{ens3}$, Inc-v3$_{ens4}$ and IncRes-v2$_{ens1}$ [83]. We craft AE on the ResNet50 surrogate model with $\epsilon = 16/255$, and use DIM+TIM as the

Table 3.2. ASR when GNP is integrated with transfer-based attacks to attack 11 target models ('*' denotes white-box attack).

Method	ϵ	ResNet50*	VGG19	ResNet152	Inc v3	DenseNet	MobileNet	SENet	ResNeXt	WRN	PNASNet	MNASNet	Average
DIM	16/255	100.00%	93.70%	93.62%	72.96%	94.32%	91.68%	79.41%	91.65%	91.17%	76.34%	89.07%	87.47%
	8/255	100.00%	74.01%	71.32%	40.58%	74.65%	71.63%	44.32%	63.38%	64.32%	40.29%	67.27%	61.28%
	4/255	100.00 %	39.21%	31.65%	15.93%	38.35%	36.74%	15.42%	25.53%	28.68%	12.40%	33.56%	27.76%
DIM +GNP	16/255	100.00%	**96.49%**	**97.38%**	**76.89%**	**97.86%**	**95.73%**	**84.56%**	**95.38%**	**96.04%**	**81.69%**	**93.51%**	**91.95%**
	8/255	100.00%	**85.63%**	**84.21%**	**49.65%**	**85.32%**	**80.59%**	**56.24%**	**72.39%**	**75.52%**	**51.68%**	**78.16%**	**72.24%**
	4/255	100.00%	**51.36%**	**45.69%**	**27.96%**	**51.39%**	**49.29%**	**28.13%**	**40.08%**	**39.64%**	**25.97%**	**45.23%**	**40.96%**
TIM	16/255	100.00%	79.90%	76.28%	54.41%	85.42%	77.68%	55.02%	74.15%	73.86%	62.07%	74.38%	73.34%
	8/255	100.00%	54.91%	44.76%	28.29%	58.17%	51.02%	24.16%	41.70%	46.08%	29.05%	48.92%	41.71%
	4/255	99.92%	24.31%	17.23%	12.67%	28.42%	23.24%	6.56%	15.03%	18.25%	9.94%	22.76%	18.95%
TIM +GNP	16/255	100.00%	**93.61%**	**90.39%**	**68.43%**	**96.89%**	**91.23%**	**69.01%**	**87.32%**	**84.69%**	**76.25%**	**85.39%**	**84.30%**
	8/255	100.00%	**70.03%**	**61.29%**	**45.12%**	**71.35%**	**66.23%**	**41.03%**	**55.46%**	**60.12%**	**46.20%**	**62.97%**	**57.99%**
	4/255	100.00%	**35.96%**	**35.03%**	**25.16%**	**43.17%**	**36.95%**	**20.36%**	**30.31%**	**32.01%**	**23.68%**	**39.05%**	**32.27%**

Table 3.3. Attacking 6 "secured" models either are adversarially trained or with advanced defense strategies.

Source model	Attack	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens1}$	JPEG	R&P	NRP
ResNet50	DIM+TIM	52.13%	48.79%	38.96%	54.85%	49.75%	39.44%
	DIM+TIM+GNP	**65.69%**	**63.16%**	**52.89%**	**66.31%**	**62.04%**	**52.81%**

"backbone" to apply GNP. The results are presented in Table 3.3, where we can see that GNP again boosts ASR significantly against the six "secured" models, achieving consistent performance improvements of 11.46–14.37%.

3.3.3. Ablation Study. We conduct ablation study on the hyper-parameters of the proposed GNP attack, i.e., step length r and regularization coefficient β. Since r represents the radius of neighborhood that is flat around current AE, a larger r is preferred; on the other hand, setting it too large will increase the approximation error of Taylor expansion and thus mislead the AE update direction. The β is to balance the goal of fooling the surrogate model and finding flat optima. 4.3 reports the results of our ablation study, where ASR is averaged over 10 target models (excluding the source ResNet50) attacked by I-FGSM + GNP with $\epsilon = 8/255$. We observe that adding the GNP regularization term clearly improves performance (as compared to $\beta = 0$) and the performance gain is rather consistent for β in

a wide range of 0.6–1.6. The step length r does not affect the performance gain too much either, and $r = 0.01$ seems to be the most stable. Thus, the ablation study reveals that GNP is not hyper-parameter sensitive and works well in a variety of conditions.

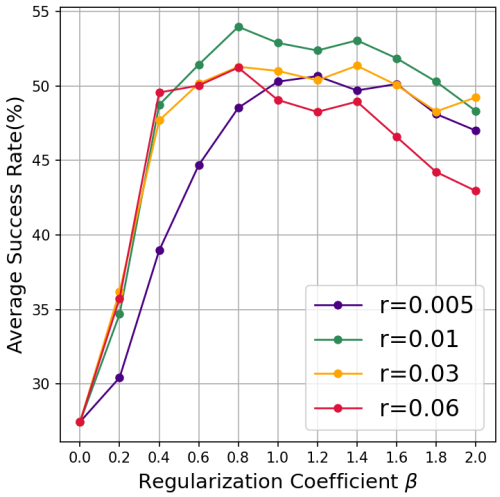


Figure 3.2. Average attack success rate (ASR) under different values of hyperparameters step length r and regularization coefficient β.

3.4. SUMMARY

In this section, we have proposed a new method for improving the transferability of AE from an optimization perspective, by seeking AE located at flat optima. We achieve this by introducing an input gradient norm penalty (GNP) which guides the AE search toward flat regions of the loss function. This GNP method is very flexible as it can be used with any gradient based AE generation methods. We conduct comprehensive experimental study and demonstrate that our method can boost the transferability of AE significantly. This section focuses on untargeted attacks, but GNP can be rather easily applied to targeted attacks as well, by making a small change to the loss function.

4. IMPROVING ADVERSARIAL TRANSFERABILITY FROM MODEL PERSPECTIVE: LIPSCHITZ REGULARIZED SURROGATE

4.1. INTRODUCTION

Adversarial attacks are commonly launched under two settings, white-box and black-box attacks. In the white-box setting, adversaries have full knowledge of target models, including model structures, parameters and weights, data and loss functions used to train the models. Therefore, they can add such perturbation to benign images that the loss on the perturbed images is maximized. An efficient way to do this involves iteratively incorporating the gradient of the loss w.r.t. input [24, 26] into perturbations. White-box attacks are important for evaluating and developing robust models, and also serve as the backend method for many black-box attacks. However, they are limited by the requirement of having to know the internal details of target models. In the black-box setting, adversaries do not need insider knowledge about target models other than their external interface (type/format of input and output), and usually take two types of approaches, query-based or transfer-based. Query-based approaches attempt to estimate the gradients of a target model's loss function by querying it with a large number of input samples and inspecting the outputs. Such frequent queries make it easy to be detected and defend them. On the other hand, transfer-based approaches use *surrogate* models to generate *transferable* AE which can attack a wide range of models, and hence are more effective to form stronger and more covert black-box attacks.

The *transferability* of AE is of central importance for transfer-based attacks. Unveiling principles of adversarial transferability provides insight into understanding the working mechanism of DNNs and designing robust DNNs. In the literature, several directions have been explored to improve the transferability of AE from the attackers' perspective. These include optimization-based [1, 2], smoothing-based [47, 48, 76, 77], attention-based [51], and ensemble-based [74] methods. Despite these efforts, a large gap of attack success rate

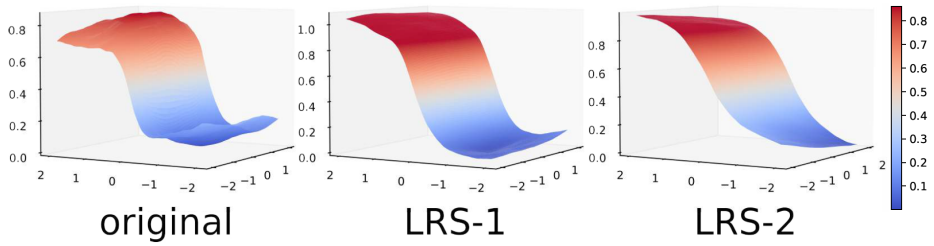


Figure 4.1. The loss landscape of original and transformed surrogate model: corrugated vs. smooth. Transformed surrogate models offer more stable input gradients and make the generated AE more generalizable, enabling more potent attacks.

still exists between the transfer-based black-box setting and the ideal white-box setting. The major reason is that AE created on a surrogate model can easily be trapped into the surrogate model's exclusive blind spots, resulting in poor generalization to fool other target models — a phenomenon known as AE *overfitting*.

Prior work on boosting adversarial transferability has focused on the AE crafting process itself, either by (1) manipulating the input images [47] or their attention maps [54], or (2) tuning the AE optimization steps such as applying momentum [1] or variance reduction [99]. However, the surrogate model, on which AE crafting is hinged, has been taken as given and not adequately explored. Specifically, what internal properties of a surrogate model are important to produce transferable AE, and (how) are they achievable? Answering this question points toward a new direction to adversarial machine learning.

We were inspired by the intricate terrain of the loss landscape w.r.t. inputs, which is characterized by peaks, valleys, and plateaus, profoundly influencing the behavior of optimization algorithms that generate AE. Thus, we propose to impose local Lipschitz regularization on the loss landscape of *surrogate models*, striving to alleviate notorious challenges in optimization posed by sharp gradients, vanishing or exploding gradients, and chaotic oscillations of gradient descent within the loss landscape. Upon that, the optimization process can traverse terrains with ease, not encountering steep slopes, cliffs, narrow valleys, etc., thereby allowing for creating stronger (i.e., more generalizable) transfer-

based black-box attacks. As shown in Figure 4.1, such regularized surrogate models offer more stable input gradients and flatter local optima which help avoid AE overfitting and create more transferable AE.

The contributions of this section are summarized below:

- Unlike prior work which all focuses on the AE generation process per se, we transform surrogate models on which that process is based, such that any existing transfer-based black-box AE generation methods can simply run on our LRS-transformed surrogate models, like a "cushion", without any change yet achieving much better performance.
- To the best of our knowledge, this is the first work that establishes a connection between the inner properties of surrogate models and AE transferability. We identify three such properties that would favor adversarial transferability, namely smaller local Lipschitz constant, smoother loss landscape, and stronger adversarial robustness, offering further insights into understanding adversarial transferability.
- We conduct extensive evaluation on ImageNet and demonstrate that, by applying LRS to a basic AE generation method (PGD), it yields superior adversarial transferability for 7 state-of-the-art black-box attacks on 10 target models.

4.2. METHODOLOGY

Given a classification model $f(x) : x \in \mathcal{X} \rightarrow y \in \mathcal{Y}$ that outputs a predicted label y for an input x, we aim to craft an adversarial example x^* which is visually indistinguishable from x but will be misclassified by the classifier, i.e., $f(x^*) \neq y$. This objective can be formulated as the following optimization problem:

$$\arg\max_{x^*} \ell(x^*, y), \quad \text{s.t. } \|x^* - x\|_p \leq \epsilon, \tag{4.1}$$

where the loss function $\ell(\cdot, \cdot)$ is often the cross-entropy loss, and the l_p-norm measures the discrepancy between x and x^*. We adopt $p = \infty$ as is common in the literature. Optimizing Eq. (4.1) needs to calculate the gradient of the loss function, which unfortunately is not accessible in the black-box setting. Therefore, we seek a surrogate model on which we aim to create transferable AE that can attack many other unknown target models.

The choice of surrogate model plays a critical role in generating transferable AE. However, previous works have focused on *selecting* pretrained surrogate models in terms of network architecture, model capacity and accuracy [49], and attacking them *as given*. Those models' internal properties such as loss geometry and robustness have been overlooked. In our work, we set to alter any given surrogate model towards desired internal properties that favor adversarial transferability.

4.2.1. LRS-1: Lipschitz Regularization on the First Order of Loss Landscape.

Definition 1 *A function $f(x)$ is locally L_c-*Lipschitz continuous *on an open set $\Omega \subset \mathbb{R}^m$ if there exists a constant $0 \leq L_c < \infty$ satisfying*

$$\forall x_1, x_2 \in \Omega,\ \|f(x_1) - f(x_2)\|_2 \leq L_c \|x_1 - x_2\|_2.$$

The smallest L_c for which the above inequality is satisfied is called the *Lipschitz constant* of $f(\cdot)$. Without loss of generality, we assume that the loss function of surrogate model is a locally Lipschitz function around a datapoint x (i.e., in the neighborhood $\mathcal{B}_\epsilon(x) = \{x' : \|x - x'\|_2 \leq \epsilon\}$). Our aim is to restrict the local Lipschitz constant L_c. The rationale is that if the loss function of the surrogate model has a small local Lipschitz constant L_c, the change of loss will be small in the neighborhood of x; thus for any adversarial examples x^* that incurs a large loss $\ell(x^*)$, datapoints around x^* are also likely to incur large loss, and hence tend to be adversarial on *other* unknown target models as well since neural network classifiers generally share similar decision boundaries and loss landscape [55].

To constrain the local Lipschitz constant L_c, we derive a regularization term that is specifically designed to modify the loss landscape of surrogate models. This regularization term aims to ensure that the surrogate models adhere to the desired smoothness properties, thereby controlling the sensitivity of the model's predictions to small changes in the input. By reshaping the loss landscape, the regularization term helps in achieving more stable and reliable predictions, ultimately aligning the model's behavior with the goal of maintaining a controlled Lipschitz constant. According to the mean value theorem, for all $x' \in \mathcal{B}_\epsilon(x)$,

$$\|\ell\left(x'\right) - \ell\left(x\right)\|_2 = \|\nabla\ell(\zeta)\left(x' - x\right)\|_2\,, \tag{4.2}$$

where $\zeta = cx + (1-c)x', c \in [0,1]$. Then the Cauchy-Schwarz inequality gives that

$$\|\ell\left(x'\right) - \ell\left(x\right)\|_2 \leq \|\nabla\ell(\zeta)\|_2 \|(x' - x)\|_2\,. \tag{4.3}$$

When $x' \to x$, the corresponding Lipschitz constant $L_c = \|\nabla\ell(\zeta)\|_2$ approximates to $\|\nabla\ell\left(x\right)\|_2$. Therefore, we transform our original aim of constraining the Lipschitz constant L_c into constraining $\|\nabla\ell\left(x\right)\|_2$ so that the crafted AE would reach a smoother and flatter optimum when maximizing the loss.

$$L_c\|\nabla f(x)\|_2 \geq f(x + \nabla f(x)) - f(x) \geq \langle\nabla f(x), \nabla f(x)\rangle = \|\nabla f(x)\|_2^2 \tag{4.4}$$

To this end, we impose the constraint of small Lipscitz constant to the loss of surrogate model by optimizing the following new objective:

$$L(x, y) = \ell(x, y) + \lambda_1 \|\nabla_x\ell(x, y)\|_2^2 \tag{4.5}$$

where $\ell(\cdot)$ is the original loss function of the surrogate model, and we square the gradient norm in order to penalize more on larger norms.

4.2.2. LRS-2: Lipschitz Regularization on the Second Order of Loss Landscape.

Definition 2 *A function $f(x)$ is said to have a Lipschitz continuous gradient on an open set $\Omega \subset \mathbb{R}^m$ if there exists a constant $0 \leq L_s < \infty$ satisfying*

$$\forall x_1, x_2 \in \Omega,\ \|\nabla f(x_1) - \nabla f(x_2)\|_2 \leq L_s \|x_1 - x_2\|_2.$$

From the convex optimization theory, we know that for a twice differentiable strongly convex $f(\cdot)$, the largest eigenvalue of the Hessian of f is uniformly upper bounded by L_s everywhere on Ω. That is,

$$L_s I \geq \nabla^2 f(x) \tag{4.6}$$

Also we have

$$\sum_{i=1}^{n} \lambda_i^2 = \left\|\nabla^2 f(x)\right\|_F^2 \tag{4.7}$$

Our aim is to restrict the Lipschitz continuous gradient of f, such that the largest eigenvalue of the Hessian of f will be small. The rationale is that the local curvature geometry of a function is measured by its Hessian, whose eigenvectors and eigenvalues describe the directions of principal curvature and the amount of curvature in each direction, respectively. Thus, limiting the eigenvalues will lead to smaller curvature which translates to a more linear behaviour of the surrogate network. Besides, this regularization penalizes a steep loss surface, encouraging the optimization to move towards regions of flatter curvature, where the generated AE will have a better ability to generalize to new, unseen models [46].

Thus, we propose a regularization on the second-order of the loss landscape as follows by "linearlizing" the surrogate model:

$$L(x, y) = \ell(x, y) + \lambda_2 \left\|\nabla_x^2 \ell(x, y)\right\|_F^2 . \tag{4.8}$$

Remark: Note that the above two regularization formulations (4.5) (4.8) concern local Lipschitzness with respect to the *input space* instead of the *parameter space*. This is an important distinction from conventional neural network optimization.

4.2.3. Optimizing the Regularized Loss. In view of practical implementation, we also consider reducing the computational overhead and make our attack scalable to large neural networks and datasets. To this end, instead of training a surrogate model using our proposed regularized objective from scratch, we propose to fine-tune a pretrained network with only a few extra epochs (10 epochs in our implementation).

To efficiently calculate the regularization terms in (4.5) and (4.8), we approximate them with finite difference methods (FDM), because computing the full Hessian matrix would incur prohibitive cost for high-dimensional datasets.

Let d be the input gradient direction,gradient direction i.e., $d = \text{sign}(\nabla_x \ell(x, y))$, h be the finite difference step size. Then, the input gradient norm is approximated by

$$\|\nabla_x \ell(x, y)\|_2^2 \approx \left(\frac{\ell(x + h_1 d, y) - \ell(x, y)}{h_1} \right)^2 \tag{4.9}$$

Similary,

$$\left\|\nabla_x^2 \ell(x, y)\right\|_F^2 \approx \left(\frac{\nabla_x \ell(x + h_2 d, y) - \nabla_x \ell(x, y)}{h_2} \right)^2 \tag{4.10}$$

This approximation significantly reduces the computational overhead associated with directly calculating the Hessian matrix. Moreover, it provides the additional benefit of controlling large variations in both the loss function and its gradient. This is achieved through the step size h, which defines the neighborhood size around the datapoint x. By adjusting h, we can effectively manage the extent of variations, leading to a more stable and efficient optimization process.

Algorithm 3 LRS-1 (using PGD as an example base)

Input: A clean sample x with ground-truth label y; a pretrained surrogate model $f(\cdot)$;
Hyper-parameters: Finetune epochs n; batch size m; learning rate η; training dataset D; step size h; perturbation size ϵ; maximum iterations T; regularization coefficient λ
Output: A transferable AE x^{adv}

Pretrained surrogate model f_0 with weight w_0
for epoch = 0 to $n-1$ **do**
  for t = 0 to $len(D)/m$ **do**
    sample minibatch $\{(x_i, y_i)\}_{i=1,\ldots,m}$
    $g_i = \nabla_x \ell\left(x_i, y_i; w_t\right)$
    $d_i = \text{sign}(g_i)$
    $z_i = x_i + h d_i$
    $\mathcal{L}(w_t) = \sum_{i=1}^{m} \ell\left(x_i, y_i; w_t\right)$
    $\mathcal{R}(w_t) = \sum_{i=1}^{m} \left(\ell\left(z_i, y_i; w_t\right) - \ell\left(x_i, y_i; w_t\right)\right)^2$
    $w_{t+1} = w_t - \frac{1}{m}\eta \nabla_w \left(\mathcal{L}\left(w_t\right) + \frac{1}{h^2}\lambda \mathcal{R}\left(w_t\right)\right)$
  end for
end for
save finetuned surrogate model f_n with weight w_n

$\alpha = \epsilon/T$; $x_0^{adv} = x$
for $t = 0$ to $T-1$ **do**
  $g_t = \nabla_x \ell(x, w_n)$
  $x_{t+1}^{adv} = x_t^{adv} + \alpha \cdot \text{sign}(g_t)$
  $x_{t+1}^{adv} = \text{clip}\left(x_{t+1}^{adv}, 0, 1\right)$
end for
return $x^{adv} = x_T^{adv}$

Algorithm 3 presents LRS-1 in its entirety, employing Projected Gradient Descent (PGD) [26] as a simple base to substantiate the attack. Notably, LRS serves as a versatile "cushion" where any transfer-based black-box attack can run on top of it (applied to that attack's chosen surrogate model) without change, yet reaping performance gains. The application of LRS-2 mirrors that of LRS-1.

The LRS approach is flexible whereby it allows the combined use of LRS-1 and LRS-2 as a "double cushion." Achieving this simply involves a weighted sum of the two regularization terms applied to the loss function. We refer to this scenario as LRS-F. In our experiments, we demonstrate the enhanced performance of LRS-F.

4.3. EXPERIMENTS

Datasets. We test untargeted ℓ_∞ black-box attacks on CIFAR-10 [100] and ImageNet [5] datasets as the common benchmark [1, 48, 57, 77]. For CIFAR-10, we perform attacks on all test data. For ImageNet, we randomly sample 5,000 test images that are correctly classified by all the target models from the ImageNet validation set. Inputs to all models are re-scaled to $[0.0, 1.0]$.

Models under attack. We take CIFAR-10 dataset for quick experiments verification, DenseNet [91] is chosen as surrogate model due to its small model size and high classification accuracy, and five other networks serving as target (victim) models: VGG-19 with batch normalization [89], ResNet-18 [88], WRN [95], ResNeXt [94], PyramidNet [101]. For ImageNet, we choose ResNet-50 [88] as the surrogate model and 10 state-of-the-art classifiers as target victim models: VGG-19 [89], ResNet-152 [88], Inception v3 [90], DenseNet [91], MobileNet v2 [92], SENet [93], ResNeXt [94], WRN [95], PNASNet [96], and MNASNet [97]. For the above victim models, we follow their official pre-processing pipelines in our evaluation.

Implementation details on ImageNet. For LRS-1 regularization, we set $\lambda_1 = 5.0$, $h_1 = 0.01$. For LRS-2 regularization, we set $\lambda_2 = 5.0$, $h_2 = 1.5$. When use LRS-F as regularization, we keep the same λ and h values. We use an SGD optimizer with momentum 0.9 and weight decay 0.0005, the learning rate is fixed at 0.001, and the surrogate model is run for 10 epochs which is a tradeoff between efficiency and efficacy. With PGD as the back-end method, we run it for 50 iterations on ImageNet with perturbation range 8/255 and step size of 2/255. All experiments are performed on an NVIDIA V100 GPU.

Table 4.1. Attack success rates of adversarial examples crafted on CIFAR10 dataset using original and transformed surrogate model under ℓ_∞ constraint with $\epsilon = 4/255$ and $\epsilon = 8/255$, PGD serves as the backbone method. '*' denotes white-box attacks.

ϵ	Transformed?	DenseNet*	VGG19	ResNet18	WRN	ResNeXt	PyramidNet	Average
4/255	No	100.00%	29.79%	19.04%	54.41%	69.41%	21.53%	38.84%
	LRS-1	99.73%	55.97%	42.16%	72.66%	80.93%	42.64%	58.87%
	LRS-2	99.82%	59.86%	48.98%	77.81%	88.63%	46.78%	64.21%
	LRS-F	99.93%	**65.16%**	**54.23%**	**81.49%**	**92.76%**	**51.07%**	**68.94%**
8/255	No	100.00%	60.13%	35.54%	86.41%	95.60%	46.62%	64.85%
	LRS-1	100.00%	93.41%	77.82%	98.79%	99.75%	88.09%	91.57%
	LRS-2	100.00%	95.26%	81.43%	99.27%	**99.87** %	92.69%	93.71%
	LRS-F	100.00%	**96.21%**	**86.41%**	**99.45%**	99.84%	**95.46%**	**95.48**%

4.3.1. Experimental Results. We conducted several sets of experiments in order to thoroughly evaluate the proposed approach.

Validation on small scale. We first experiment on the relatively smaller CIFAR-10 using DenseNet as surrogate to evaluate LRS. In Table 4.1, we compare adversarial transferability over the original pretrained surrogate model and that over LRS-transformed surrogate models, all using PGD as the base attack. The evaluation involved two perturbation scales ϵ (4.1). We observe that: (1) overall, applying LRS results in clear improvement by large margins; (2) LRS-2 boosts adversarial transferability more than LRS-1; (3) the best surrogate model is achieved by using both the first and second order regularization together, i.e., LRS-F, while at the cost of slightly higher computation overhead. Specifically, when $\epsilon = 4/255$, we see an absolute value increase in the average attack success rate (ASR) of 20.03%, 25.37% and 30.10% when the surrogate model is transformed by LRS-1, LRS-2 and LRS-F, respectively; when $\epsilon = 8/255$, the corresponding improvements are 26.72%, 28.86% and 30.63%, respectively. All of these are significant enhancements. In particular, when attacking PyramidNet with LRS-F transformed surrogate model under $\epsilon = 8/255$, we achieved an increase of ASR by a remarkable 48.84% in absolute value.

Table 4.2. Attack success rates of SOTA transfer-based untargeted attacks on ImageNet using ResNet-50 as the surrogate model and PGD as the backend attack method, under the ℓ_∞ constraint with $\epsilon = 8/255$. '*' denotes white-box attack.

Method	ResNet-50*	VGG-19	ResNet-152	Inception v3	DenseNet	MobileNet
PGD (2018)	100.00%	39.22%	29.18%	15.60%	35.58%	37.90%
TIM (2019)	100.00%	44.98%	35.14%	22.21%	46.19%	42.67%
SIM (2020)	100.00%	53.30%	46.80%	27.04%	54.16%	52.54%
LinBP (2020)	100.00%	72.00%	58.62%	29.98%	63.70%	64.08%
Admix (2021)	100.00%	57.95%	45.82%	23.59%	52.00%	55.36%
TAIG (2022)	100.00%	54.32%	45.32%	28.52%	53.34%	55.18%
ILA++ (2022)	99.96%	74.94%	69.64%	41.56%	71.28%	71.84%
LRS-1 (ours)	100.00%	76.02%	72.36%	42.01%	71.23%	69.36%
LRS-2 (ours)	100.00%	78.24%	75.96%	46.14%	73.01%	73.45%
LRS-F (ours)	100.00%	**80.64%**	**78.21%**	**50.10%**	**75.19%**	**76.24%**

Method	SENet	ResNeXt	WRN	PNASNet	MNASNet	**Average**
PGD (2018)	17.66%	26.18%	27.18%	12.80%	35.58%	27.69%
TIM (2019)	22.47%	32.11%	33.26%	21.09%	39.85%	34.00%
SIM (2020)	27.04%	41.28%	42.66%	21.74%	50.36%	41.69%
LinBP (2020)	41.02%	51.02%	54.16%	29.72%	62.18%	52.65%
Admix (2021)	30.28%	41.94%	42.78%	21.91%	52.32%	42.40%
TAIG (2022)	24.82%	38.36%	42.16%	17.20%	54.90%	41.41%
ILA++ (2022)	53.12%	65.92%	65.64%	44.56%	70.40%	62.89%
LRS-1 (ours)	54.27%	66.85%	67.21%	45.29%	72.03%	64.53%
LRS-2 (ours)	57.19%	69.48%	71.13%	48.39%	75.68%	67.57%
LRS-F (ours)	**59.68%**	**71.96%**	**74.61%**	**52.43%**	**76.87%**	**69.91%**

Comparison with SOTA on large scale. We compare the attacking performance of LRS on 10 target models with state-of-the-art (SOTA) attacking methods, on the relatively large ImageNet dataset (the same comparison on CIFAR10 is reported in supplementary material). The SOTA attack methods for comparison include TIM [48], SIM [2], LinBP [77], Admix [50], TAIG [102] and ILA++ [103]. The results are presented in Table 4.2, which shows that all the LRS-cushioned attacking methods (LRS-1, LRS-2, LRS-F) outperform all the SOTA methods considerably. For example, looking at the Average ASR column of Table 4.2, LRS-F achieves an improvement over all the SOTA methods of between 7.02–35.91%.

Table 4.3. Attack success rates by combining SOTA transfer-based untargeted attacks with our methods, on CIFAR-10 using DenseNet as the surrogate model and PGD as the backbone attack method, under the ℓ_∞ constraint with $\epsilon = 4/255$. '*' denotes white-box attack.

Method	DenseNet*	VGG19	ResNet18	WRN	ResNeXt	PyramidNet	Average
TIM (2019)	100.00%	33.96%	23.46%	56.49%	72.38%	23.14%	41.89%
TIM+LRS-1	100.00%	64.23%	53.19%	81.03%	86.95%	50.62%	67.80%
TIM+LRS-2	100.00%	69.21%	57.39%	86.98%	90.12%	55.13%	71.17%
TIM+LRS-F	100.00%	**73.86%**	**61.48%**	**90.11%**	**93.48%**	**60.42%**	**75.87%**
Admix (2021)	100.00%	44.09%	34.80%	64.36%	76.24%	27.65%	49.43%
Admix+LRS-1	100.00%	66.49%	58.96%	85.69%	89.65%	55.48%	71.05%
Admix+LRS-2	100.00%	74.39%	63.59%	88.94%	93.56%	62.47%	76.39%
Admix+LRS-F	100.00%	**78.12%**	**68.04%**	**94.23%**	**95.37%**	**67.96%**	**80.14%**
TAIG (2022)	100.00%	41.69%	30.23%	64.12%	75.89%	25.96%	47.78%
TAIG+LRS-1	100.00%	62.38%	51.29%	80.33%	84.68%	51.46%	66.03%
TAIG+LRS-2	100.00%	73.18%	62.08%	84.39%	92.04%	60.03%	74.34%
TAIG+LRS-F	100.00%	**75.98%**	**65.21%**	**89.11%**	**93.16%**	**63.49%**	**77.99%**

Easily integrating with and supporting other attacks. As previously noted, LRS is a flexible "cushion" on which any other transfer-based black-box attack can execute without any change. In Table 4.3, we report the results when applying LSR to TIM, Admix and ILA++ (besides PGD which has been shown). It can be seen that the transferability is enhanced significantly by 20–34% on average due to the use of LRS.

Attacking "secure" models. Besides attacking regularly trained models, we also conduct experiments on attacking adversarially trained models and models equipped with advanced defense methods—which are hence more "secure"—for a thorougher evaluation of our proposed LRS method. We consider six such secure models, Inc-v 3_{ens3}, Inc-v 3_{ens4} and IncRes-v 2_{ens} [83], JPEG [39], R&P [36] and NRP [86]. Specifically, we craft adversarial examples on ResNet50 with $\epsilon = 16/255$ on our selected ImageNet dataset and test the transferability on the six secure target models. The results are shown in Table 4.4. It shows that LRS remarkably improves the transferability of the backbone attack methods

Table 4.4. Attacking "secure" models which underwent adversarial training or are equipped with advanced defense methods.

Source model	Attack	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes-v2$_{ens1}$	JPEG	R&P	NRP	Average
ResNet50	PGD	4.3%	5.0%	2.3%	6.9%	2.2%	2.5%	3.87%
	PGD + LRS-1	25.1%	24.7%	21.9%	38.6%	26.3%	29.4%	27.67%
	PGD + LRS-2	26.7%	25.3%	23.0%	41.2%	28.4%	31.8%	29.40%
	PGD + LRS-F	**28.3%**	**27.6%**	**25.2%**	**44.1%**	**29.8%**	**33.1%**	**31.35%**
ResNet50	TIM	21.3%	19.8%	12.6%	33.6%	18.4%	15.2%	20.15%
	TIM + LRS-1	40.8%	41.0%	26.9%	51.4%	39.6%	34.1%	38.97%
	TIM + LRS-2	43.6%	45.2%	30.4%	55.3%	43.5%	37.1%	42.52%
	TIM + LRS-F	**45.7%**	**48.2%**	**32.8%**	**59.4%**	**45.8%**	**40.3%**	**45.37%**

on all the six presumably more robust models. On average, the performance is improved by 23.80–27.48% and 18.82–25.22%, respectively, when using I-FGSM and TIM as the backbone attack methods.

4.3.2. Exploring Further: Factors Enhancing Adversarial Transferability in Regularized Surrogate Models. In this section, we explore the factors that enhance adversarial transferability in regularized surrogate models. Our analysis delves into the mechanisms and conditions that improve the effectiveness of adversarial attacks across different model architectures.

Smaller local Lipschitz constant. A reduced Lipschitz constant indicates a smoother classifier. Therefore, we delve into whether our transformed surrogate models indeed exhibit increased smoothness through a smaller local Lipschitz constant. While computing the precise Lipschitz constant remains a open challenge, we can empirically gauge the surrogate models' local Lipschitzness using the empirical Lipschitz constant [104]:

$$L_{emp} = \frac{1}{n}\sum_{i=1}^{n} \max_{\boldsymbol{x}_i' \in \mathbb{B}_\infty(\boldsymbol{x}_i, \varepsilon)} \frac{\left\| f\left(\boldsymbol{x}_i\right) - f\left(\boldsymbol{x}_i'\right) \right\|_2}{\left\| \boldsymbol{x}_i - \boldsymbol{x}_i' \right\|_2} \tag{4.11}$$

We estimate this value using a PGD-like approach and calculate the average estimation across all test data points. Refer to Table 4.5 for the empirical local Lipschitz constants. It clearly shows that our transformed surrogate models display significantly re-

Table 4.5. Empirical local Lipschitz constant of surrogate model estimated via Eq. (4.11). The constants of DenseNet and ResNet50 are evaluated on CIFAR10 and ImageNet, respectively.

Surrogate model	DenseNet100	ResNet50
Original pretrained	5.53	976.59
Transformed by LRS-1	0.79	57.62
Transformed LRS-2	0.67	53.21
Transformed LRS-F	**0.59**	**49.64**

duced local Lipschitz constants (by *more than an order of magnitude*). This contributes to a notably smoother loss landscape, minimizing the likelihood of the AE generation process being confined to undesirable local optima. Such optima yield low loss values yet possess complex non-smooth geometries that are challenging to navigate away from.

Smoother loss landscape. Research has extensively explored flat optima's role in model generalization [62, 105, 106], highlighting how optimizing weights toward flat optima can improve neural network generalization due to their robustness against shifts in the loss function between training and test data. In our context of developing first-order Lipschitz regularization, we propose that *adversarial examples positioned within flat optima exhibit robustness against shifts in the loss function between surrogate and target models*, thus enhancing AE transferability.

To verify our hypothesis, we visualize the loss landscape of a surrogate model before and after transformation in Figure 4.1. The original pretrained surrogate model features a highly non-linear and jagged loss surface. Conversely, regularization results in a notably smoother loss surface with flatter local optima. This visualization confirms our regularization strategy's effectiveness in smoothing out sharp optima in the loss landscape. Consequently, AEs generated using regularized surrogate models are more likely to reside within flat optima, boosting their transferability.

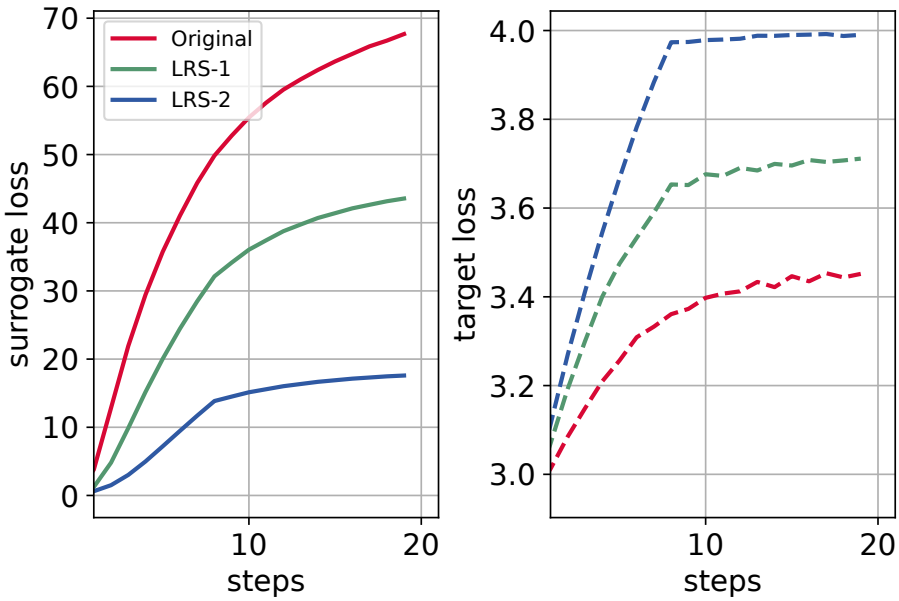


Figure 4.2. The loss of surrogate model (DenseNet) and target model (ResNet18), w.r.t. PGD-generated AE. It reveals that LRS-transformed models demonstrate more robustness and enable more transferable attacks.

More robust against attacks. Another perspective explaining the favorability of Lipschitz-regularized surrogate models for adversarial transferability is their increased robustness against adversarial attacks. When a neural network possesses a small Lipschitz constant, it signifies a strict control over changes in network output amidst input perturbations, leading to certified robustness guarantees [107, 108]. Consequently, generating AEs on such robust surrogate models enhances the effectiveness of the resulting AEs in deceiving less robust target models. The robustness contributes to adversarial transferability. [109] also demonstrate that enhancing the robustness of the source classifier against small-magnitude adversarial examples, significantly enhances the transferability of targeted adversarial attacks.

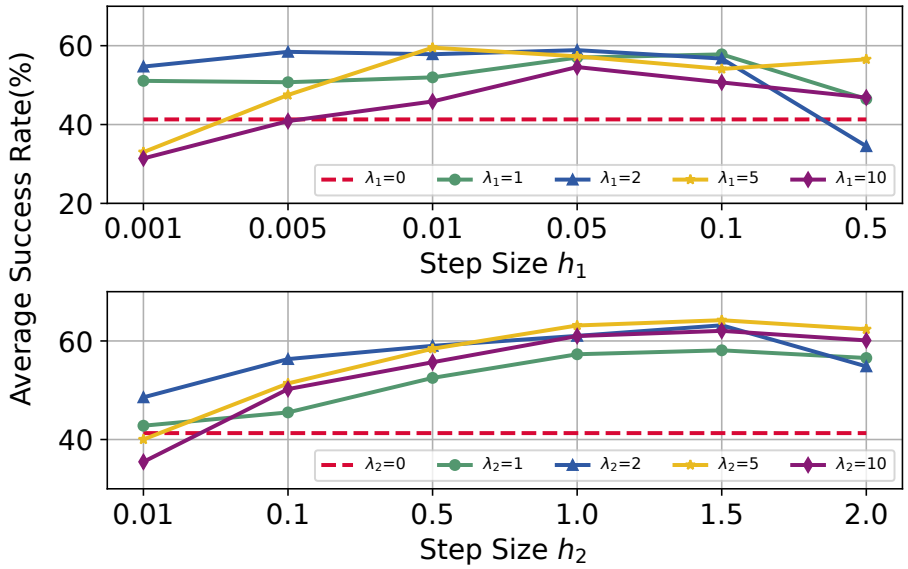


Figure 4.3. Ablation studies on average ASR under different hyperparameters h and λ, the performance gains are consistent in a wide range of hyper-parameter values.

In line with this notion, Figure 4.2 illustrates that adversarial examples generated by PGD yield significantly lower losses on regularized surrogate models, indicating their enhanced robustness. However, their loss on target models is higher, signifying stronger black-box attacks and improved transferability.

4.3.3. Ablation Studies. We perform ablation studies on two crucial hyperparameters in our proposed LRS approach: the *step size* and *regularization coefficient*, denoted as h_1 and λ_1 for LRS-1, and h_2 and λ_2 for LRS-2. These parameters influence the locally enforced Lipschitz radius around the current AE. Larger values of h_1 and h_2 are generally preferred to enhance the neighborhood radius. Conversely, excessively large values can introduce finite difference method approximation errors, potentially misleading the AE update direction. The values of λ_1 and λ_2 serve to balance the trade-off between model accuracy and Lipschitz regularization.

Figure 4.3 presents the outcomes of our ablation studies. The ASR is computed by averaging over 5 target models on CIFAR10. The attacks are executed using PGD as the backend method with $\epsilon = 4/255$. Our observations indicate that AE generated using

LRS have significantly enhanced transferability compared to the case with $\lambda = 0$. These performance improvements remain consistent across a reasonably broad range of λ and h values. This ablation study underscores the non-sensitive nature of LRS to hyperparameters, establishing its effectiveness across diverse conditions.

4.4. SUMMARY

This section introduces a novel approach to enhancing adversarial transferability by transforming surrogate models via regularization, unlike in previous research where a pretrained model is chosen as is to serve as the (fixed) surrogate. We present Lipschitz Regularized Surrogate (LRS), a technique that imposes Lipschitz regularization to surrogate models for just a few training epochs. We show that this technique enables *any* existing transfer-based black-box AE generation method to produce highly transferable adversarial examples. This is validated through comprehensive experiments involving comparisons with numerous benchmark models, attack methods, and datasets. Our findings affirm the remarkable efficacy and superiority of LRS. Moreover, we offer insights into what and how properties of surrogate models promote adversarial transferability.

5. IMPROVING GENERALIZATION OF DNNS: CURVATURE REGULARIZED SHARPNESS-AWARE MINIMIZATION

5.1. INTRODUCTION

Over the past decade, rapid advancements in deep neural networks (DNNs) have significantly reshaped various pattern recognition domains including computer vision [6], speech recognition [110], and natural language processing [111]. However, the success of DNNs hinges on their capacity to generalize—how well they would perform on new, unseen data. With their intricate multilayer structures and non-linear characteristics, modern DNNs possess highly non-convex loss landscapes that remain only partially understood. Prior landscape analysis has linked flat local minima to better generalization [58, 59, 112, 113, 114]. In particular, [59] conducted a comprehensive empirical study on various generalization metrics, revealing that measures based on sharpness exhibit the highest correlation with generalization performance. Recently, [87] introduced Sharpness-Aware Minimization (SAM), an efficient technique for minimizing loss landscape sharpness. This method has proven highly effective in enhancing DNN generalization across diverse scenarios. Given SAM's remarkable success and the significance of DNN generalization, a substantial body of subsequent research has emerged [115, 116, 117, 118, 119].

Specifically, the SAM approach formulates the optimization of neural networks as a minimax problem, where it aims to minimize the maximum loss within a small radius ρ around the parameter $\boldsymbol{w}$. Given that the inner maximization problem is NP-hard, SAM employs a practical sharpness calculation method that utilizes one-step gradient ascent as an approximation. However, our experimentation reveals a notable decline in the accuracy of this one-step approximation as training progresses (see 5.1). This phenomenon likely stems from the heightened non-linearity within the loss landscape during later stages of training. Our further investigation highlights a limitation in conventional curvature measures like

the Hessian trace and the top eigenvalue of the Hessian matrix. These measures diminish as training advances, incorrectly suggesting reduced curvature and overlooking the actual non-linear characteristics.

Consequently, we posit that the escalating non-linearity in SAM training undermines the precision of approximating and effectiveness of mitigating sharpness. Building upon these insights, we introduce the concept of a *normalized Hessian trace*. This novel metric serves as a dependable indicator of loss landscape non-linearity and behaves consistently across training and testing datasets. Guided by this metric, we propose *Curvature Regularized SAM* (CR-SAM), a novel regularization approach for SAM training. CR-SAM incorporates the normalized Hessian trace to counteract excessive non-linearity effectively.

To calculate the normalized Hessian trace, we present a computationally efficient strategy based on finite differences (FD). This approach enables parallel execution without additional computational burden. Through both theoretical analysis and empirical evaluation, we demonstrate that CR-SAM training converges towards flatter minima, resulting in substantially enhanced generalization performance.

The main contributions of this section can be summarized as follows:

- We identify that the one-step gradient ascent approximation becomes less effective during the later stages of SAM training. In response, we introduce normalized Hessian trace, a metric that can accurately and consistently characterize the non-linearity of neural network loss landscapes.

- We propose CR-SAM, a novel algorithm that infuses curvature minimization into SAM and thereby enhance the generalizability of deep neural networks. For scalable computation, we devise an efficient technique to approximate the Hessian trace using finite differences (FD). This technique involves only independent function evaluations and can be executed in parallel without additional overhead. Moreover, we also theoretically show the efficacy of CR-SAM in reducing generalization error, leveraging PAC-Bayes bounds.

- Our comprehensive evaluation of CR-SAM spans a diverse range of contemporary DNN architectures. The empirical findings affirm that CR-SAM consistently outperforms both SAM and SGD in terms of improving model generalizability, across multiple datasets including CIFAR10/100 and ImageNet-1k/-C/-R.

5.2. BACKGROUND AND RELATED WORK

Empirical risk minimization (ERM) is a fundamental principle in machine learning for model training on observed data. Given a training dataset $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^n$ drawn i.i.d. from an underlying unknown distribution $\mathcal{D}$, we denote by $f(x; \boldsymbol{w})$ a deep neural network model with trainable parameters $\boldsymbol{w} \in \mathbb{R}^p$, where a differentiable loss function w.r.t. an input x_i is given by $\ell\left(f\left(x_i; \boldsymbol{w}\right), y_i\right)$ and is taken to be the cross entropy loss in this section. The *empirical loss* can be written as $L_{\mathcal{S}}(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^n \ell\left(f\left(x_i; \boldsymbol{w}\right), y_i\right)$ whereas the *population loss* is defined as $L_{\mathcal{D}}(\boldsymbol{w}) = \mathbb{E}_{(x,y)\sim D}[\ell\left(f\left(x; \boldsymbol{w}\right), y\right)]$. The generalization error is defined as the difference between $L_{\mathcal{D}}(\boldsymbol{w})$ and $L_{\mathcal{S}}(\boldsymbol{w})$, i.e., $e(f) = L_{\mathcal{D}}(\boldsymbol{w}) - L_{\mathcal{S}}(\boldsymbol{w})$.

5.2.1. SAM and Variants. Sharpness-Aware Minimization (SAM) [87] is a novel optimization algorithm that directs the search for model parameters within flat regions. Training DNNs with this method has demonstrated remarkable efficacy in enhancing generalization, especially on transformers. SAM introduces a new objective that aims to minimize the maximum loss in the vicinity of weight $\boldsymbol{w}$ within a radius ρ:

$$\min_{\boldsymbol{w}} L^{\mathrm{SAM}}(\boldsymbol{w}) \quad \text{where} \quad L^{\mathrm{SAM}}(\boldsymbol{w}) = \max_{\|\boldsymbol{v}\|_2 \leq 1} L_{\mathcal{S}}(\boldsymbol{w} + \rho \boldsymbol{v}). \tag{5.1}$$

Through the minimization of the SAM objective, the neural network's weights undergo updates that shift them towards a smoother loss landscape. As a result, the model's generalization performance is improved. To ensure practical feasibility, SAM adopts two approximations: (1) employs one-step gradient ascent to approximate the inner maximization; (2) simplifies gradient calculation by omitting the second and higher-order

terms, i.e.,

$$\nabla L^{\mathrm{SAM}}(\boldsymbol{w}) \approx \nabla L_{\mathcal{S}}\left(\boldsymbol{w} + \rho \frac{\nabla L_{\mathcal{S}}(\boldsymbol{w})}{\|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2}\right). \tag{5.2}$$

Nevertheless, behind the empirical successes of SAM in training computer vision models [87, 120] and natural language processing models [121], there are two inherent limitations.

Firstly, SAM introduces a twofold computational overhead to the base optimizer (e.g., SGD) due to the inner maximization process. In response, recent solutions such as LookSAM [115], Efficient SAM (ESAM) [116], Sparse SAM (SSAM) [122], Sharpness-Aware Training for Free (SAF) [117], and Adaptive policy SAM (AE-SAM) [118] have emerged, which propose various strategies to reduce the added overhead.

5.2.2. Regularization Methods for Generalization. The work [123] contends that model generalization hinges primarily on two traits: the model's *support* and its *inductive biases*. Given the broad applicability of modern DNNs to various datasets, the inductive biases is the remaining crucial factor for guiding a model towards the true data distribution. From a Bayesian standpoint, inductive bias can be viewed as a prior distribution over the parameter space. Classical ℓ_1 and ℓ_2 regularization, for instance, correspond to Laplacian and Gaussian prior distributions respectively. In practice, one can employ regularization techniques to instill intended inductive biases, thereby enhancing model generalization. Such regularization can be applied to three core components of modern deep learning models: data, model architecture, and optimization.

Data-based regularization involves transforming raw data or generating augmented data to combat overfitting. Methods like label smoothing [124], Cutout [125], Mixup [126], and RandAugment [127] fall under this category. **Model-based regularization** aids feature extraction and includes techniques such as dropout [128], skip connections [6], and batch normalization [129]. Lastly, **optimization-based regularization** imparts desired properties like sparsity or complexity into the model. Common methods include weight decay [130],

gradient norm penalty [131, 132], Jacobian regularization [133], and confidence penalty [61]. Our proposed curvature regularizer in this work aligns with the optimization-based strategies, fostering flatter loss landscapes.

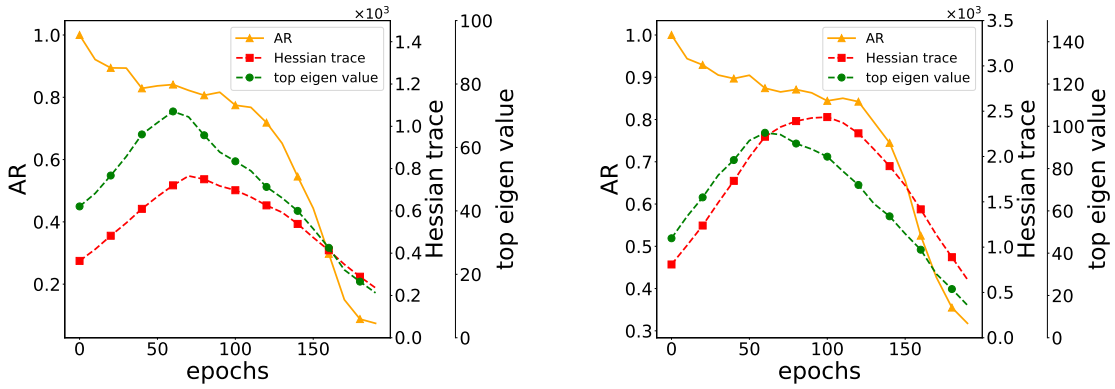
5.2.3. Flat Minima. Recent research into loss surface geometry underscores the strong correlation between generalization and the flatness of minima reached by DNN parameters. Among various mathematical definitions of flatness, including ϵ-sharpness [58], PAC-Bayes measure [59], Fisher Rao Norm [60], and entropy measures [61, 62], notable ones include Hessian-based metrics like Frobenius norm [63, 64], trace of the Hessian [65], largest eigenvalue of the Hessian [66], and effective dimensionality of the Hessian [67]. In this work, our focus is on exploring the Hessian trace and its connection to generalization. Akin to our objective, [134] also proposes Hessian trace regularization for DNNs. However, [134] utilizes the computationally demanding Hutchinson method [135] with dropout as an unbiased estimator for Hessian trace. In contrast, our method employs finite difference (FD), offering greater computational efficiency and numerical stability. Moreover, our rationale and regularization approach significantly differ from [134].

5.3. METHODOLOGY

In this section, we provide a detailed description of our methodology. We begin by discussing two key observations derived from our analysis of Sharpness-Aware Minimization (SAM) training. These observations highlight specific challenges and opportunities within the SAM framework that informed our approach.

5.3.1. Our Empirical Findings about SAM Training. In this section, we discuss two key empirical findings related to Sharpness-Aware Minimization (SAM) training.

Declining accuracy of one-step approximation. The optimal solution to the inner maximization in SAM's objective is intractable, which led SAM to resort to an approximation using one-step gradient ascent. However, we found that this approximation's accuracy diminishes progressively as training advances. To show this, we introduce the *approxima-*



(a) ResNet-18 on CIFAR10 (b) ResNet-18 on CIFAR100

Figure 5.1. The evolution of approximation ratio (AR), Hessian trace and top eigenvalue of Hessian (the two Y axes on the right) during SAM training on CIFAR10 and CIFAR100 datasets.

tion ratio (AR) for sharpness, approximated by one-step gradient ascent, defined as:

$$\mathrm{AR} = \mathbb{E}_{(x,y)\sim D}\left[\frac{\ell\left(f\left(x;\boldsymbol{w}+\boldsymbol{\delta}\right),y\right)-\ell\left(f\left(x;\boldsymbol{w}\right),y\right)}{\ell\left(f\left(x;\boldsymbol{w}+\boldsymbol{\delta}^{*}\right),y\right)-\ell\left(f\left(x;\boldsymbol{w}\right),y\right)}\right] \tag{5.3}$$

where $\boldsymbol{\delta}$ represents one-step gradient ascent perturbation, and $\boldsymbol{\delta}^*$ denotes the optimal perturbation. An AR closer to 1 indicates a better approximation. Given the infeasibility of obtaining the optimal $\boldsymbol{\delta}^*$, we employ the perturbation from a 20-step gradient ascent as $\boldsymbol{\delta}^*$ and approximate its expectation by sampling 5000 data points from the training set and calculating their average. Our assessment of AR through multiple experiments, illustrated in Figure 5.1, reveals its progression during training. The continuously decreasing AR indicates an enlarging curvature whereas both of the Hessian-based curvature metrics (which are expected to continuously increase) fail to capture the true curvature of model loss landscape. Notably, the one-step ascent approximation for sharpness demonstrates diminishing accuracy as training unfolds, with a significant decline in the later stages. This suggests an increasing curvature of the loss landscape as training advances. In the realm of DNNs, the curvature of a function at a specific point is commonly assessed through the Hessian matrix calculated at that point. However, the dependence on gradient scale make Hessian metrics

fail to measure the curvature precisely. Specifically, models near convergence of training exhibit smaller gradient norms and inherently correspond to reduced Hessian norms, but does not imply a more linear model.

We show the evolution of conventional curvature metrics like Hessian trace and the top eigenvalue of the Hessian in 5.1, both metrics increase initially and then decrease, which fail to capture true loss landscape curvature since AR's consistent decline implies a higher curvature. This phenomenon also verify their dependence on the scaling of model gradients; as gradients decrease near convergence, Hessian-based curvature metrics like Hessian trace and top eigenvalue of the Hessian also decrease.

The degrading effectiveness of the one-step gradient ascent approximation can be theoretically confirmed through a Taylor expansion. The sharpness optimized by SAM in practice is represented as:

$$\begin{aligned} R^{\mathrm{SAM}}(\boldsymbol{w}) &= L_{\mathcal{S}}\left(\boldsymbol{w} + \rho \frac{\nabla L_{\mathcal{S}}(\boldsymbol{w})}{\|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2}\right) - L_{\mathcal{S}}(\boldsymbol{w}) \\ &= \rho \|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2 + O\left(\rho^2\right) \end{aligned} \tag{5.4}$$

Eq. (5.4) highlights that as training nears convergence, the gradient $\nabla L_{\mathcal{S}}(\boldsymbol{w})$ tends toward 0, causing $R^{\mathrm{SAM}}(\boldsymbol{w})$ to approach 0 as well. Consequently, sharpness ceases to be effectively captured, and SAM training mirrors standard training behavior.

A new metric for accurate curvature characterization. Our initial observation underscores the limitations of the top Hessian eigenvalue and Hessian trace in capturing loss landscape curvature during SAM training. These metrics suffer from sensitivity to gradient scaling, prompting the need for a more precise curvature characterization. To address this challenge, we introduce a novel curvature metric, *normalized Hessian trace*, defined as follows:

$$C(\boldsymbol{w}) = \frac{\mathrm{Tr}\left(\nabla^2 L_{\mathcal{S}}\left(\boldsymbol{w}\right)\right)}{\|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2} \tag{5.5}$$

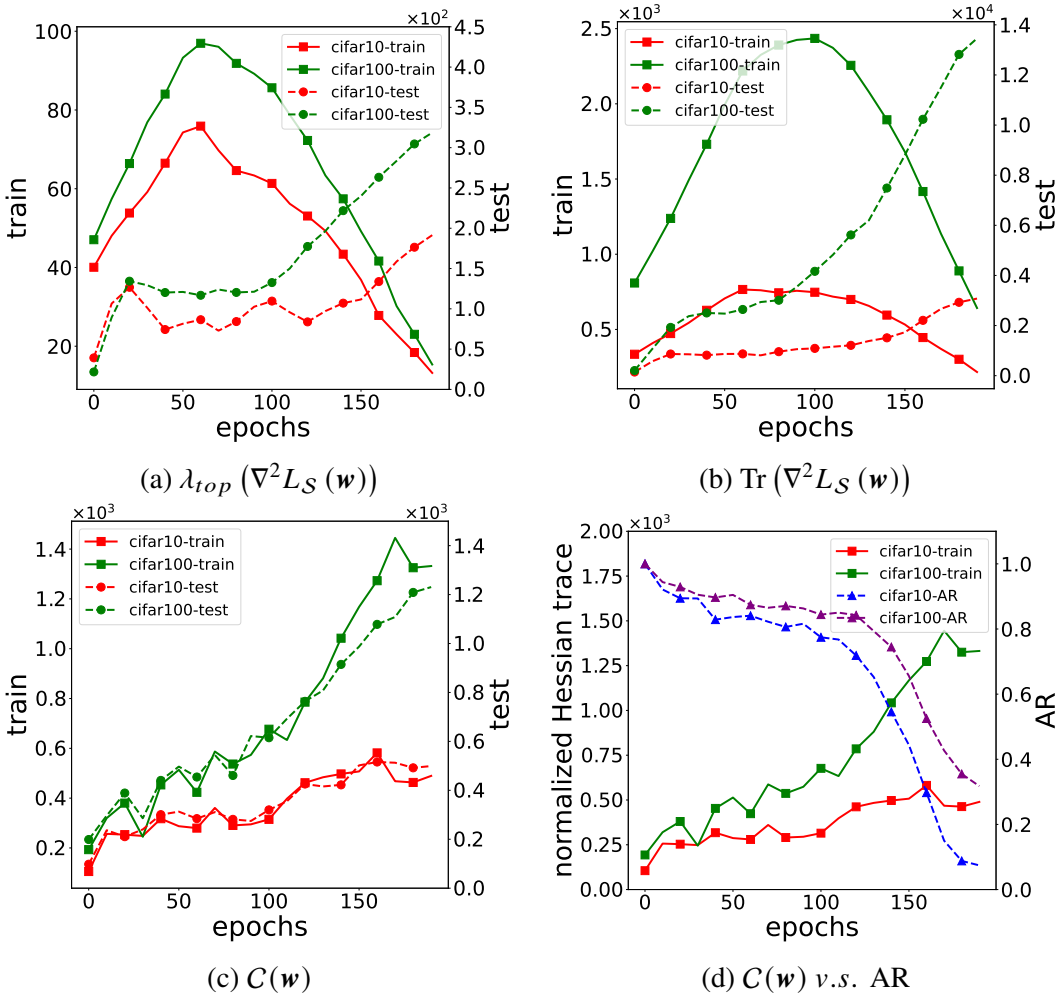


Figure 5.2. Evolution of the three curvature metrics during SAM training of ResNet-18 on CIFAR-10 and CIFAR-100.

This metric exhibits continual growth during SAM training, indicating increasing curvature. This behavior aligns well with the decreasing AR of one-step gradient ascent, as depicted in Figure 5.2. An additional advantage of the normalized Hessian trace is its consistent trends and values across both training and test sets. A similar phenomenon was observed in the domain of adversarial robustness [136]. In contrast, plain $\mathrm{Tr}\left(\nabla^2 L_{\mathcal{S}}(\boldsymbol{w})\right)$ display inconsistent behaviors between these sets, as evidenced in Figure 5.2 (a,b,c). This discrepancy questions the viability of solely utilizing Hessian trace or the top Hessian eigenvalue for DNN regularization based on training data. Subfigs. (a) (b) show that the top Hessian

eigenvalue and Hessian trace exhibit large discrepancy on train and test sets where values calculated on test set can be 50x more than those on training set. Subfig (c) shows that our proposed normalized Hessian trace shows consistent trends which implies that it well captures the true model geometry. Finally, subfig (d) illustrates that the normalized Hessian trace also reflects (inversely) the phenomenon of decreasing approximation ratio (AR) since they both indicate a growing curvature throughout training.

5.3.2. Curvature Regularized Sharpness-Aware Minimization (CR-SAM). For the sake of generalization, it is preferable to steer clear of excessive non-linearity in deep learning models, as it implies highly non-convex loss surfaces. On such models, the challenge of flattening minima (which improves generalization) becomes considerably harder, potentially exceeding the capabilities of gradient-based optimizers. In this context, our proposed normalized Hessian trace (5.5) can be employed to train deep models with more manageable loss landscapes. However, a direct minimization of $C(\boldsymbol{w})$ would lead to an elevation in the gradient norm $|\nabla L_{\mathcal{S}}(\boldsymbol{w})|_2$, which could adversely affect generalization [132]. Therefore, we propose to optimize $\mathrm{Tr}\left(\nabla^2 L_{\mathcal{S}}\left(\boldsymbol{w}\right)\right)$ and $\|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2$ separately. Specifically, we penalize both $\mathrm{Tr}\left(\nabla^2 L_{\mathcal{S}}\left(\boldsymbol{w}\right)\right)$ and $\|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2$ but with different extent such that they jointly lead to a smaller $C(\boldsymbol{w})$. Thus, we introduce our proposed curvature regularizer as:

$$R_c(\boldsymbol{w}) = \alpha \log \mathrm{Tr}\left(\nabla^2 L_{\mathcal{S}}\left(\boldsymbol{w}\right)\right) + \beta \log \|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2 \tag{5.6}$$

where $\alpha > \beta > 0$ such that the numerator of $C(\boldsymbol{w})$ is penalized more than the denominator. This regularizer is equivalent to $\alpha \log C(\boldsymbol{w}) + (\alpha+\beta) \log \|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2$, which is a combination of normalized Hessian trace with gradient norm penalty regularizer. Our regularization strategy can also be justified by analyzing the sharpness:

$$\begin{aligned} R^{\mathrm{True}}(\boldsymbol{w}) &= \max_{\|\boldsymbol{v}\|_2 \leq 1} L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) - L_{\mathcal{S}}(\boldsymbol{w}) \\ &= \max_{\|\boldsymbol{v}\|_2 \leq 1} \left(\rho \boldsymbol{v}^\top \nabla L_{\mathcal{S}}(\boldsymbol{w}) + \frac{\rho^2}{2} \boldsymbol{v}^\top \nabla^2 L_{\mathcal{S}}(\boldsymbol{w}) \boldsymbol{v} + O\left(\rho^3\right)\right) \end{aligned}$$

We can see that $\max_{\|\boldsymbol{v}\|_2 \leq 1} \rho \boldsymbol{v}^\top \nabla L_{\mathcal{S}}(\boldsymbol{w}) = \rho \|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2$ (cf. (5.2)). Under the condition that $\boldsymbol{v} \sim N(0, I)$, we have $\mathbb{E}_{\boldsymbol{v} \sim N(0,I)} \boldsymbol{v}^\top \nabla^2 L_{\mathcal{S}}(\boldsymbol{w}) \boldsymbol{v} = \operatorname{Tr}\left(\nabla^2 L_{\mathcal{S}}(\boldsymbol{w})\right)$ for the second term. However, the first-order term $\|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2$ vanishes at the local minimizers of the loss L, and thus the second-order term will become prominent and hence be penalized. Therefore, introducing our regularizer will have the effect of penalizing both the Hessian trace and the gradient norm and thereby reduce the sharpness of a loss landscape.

Informed by our heuristic and theoretical analysis above, our CR-SAM optimizes the following objective:

$$\begin{aligned} &\min_{\boldsymbol{w}} L^{\text{CR-SAM}}(\boldsymbol{w}) \\ &\text{where } L^{\text{CR-SAM}}(\boldsymbol{w}) = L^{\text{SAM}}(\boldsymbol{w}) + R_c(\boldsymbol{w}) \end{aligned} \tag{5.7}$$

5.3.3. Solving Computational Efficiency. Computing the Hessian trace as in $R_c(\boldsymbol{w})$ for very large matrices is computationally intensive, especially for modern over-parameterized DNNs with millions of parameters. To address this issue, we first propose a stochastic estimators for $R_c(\boldsymbol{w})$:

$$\begin{aligned} R_c(\boldsymbol{w}) &= \alpha \log \operatorname{Tr}\left(\nabla^2 L_{\mathcal{S}}(\boldsymbol{w})\right) + \beta \log \|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2 \\ &= \mathbb{E}_{\boldsymbol{v} \sim N(0,I)} \left[\alpha \log \boldsymbol{v}^\top \nabla^2 L_{\mathcal{S}}(\boldsymbol{w}) \boldsymbol{v} + \beta \log \boldsymbol{v}^\top \nabla L_{\mathcal{S}}(\boldsymbol{w})\right] \end{aligned}$$

which reduces Hessian trace computation to averages of Hessian-vector products. However, the complexity of computing the Hessian-vector products in the above estimator is still high for optimizers in large scale problems. Hence, we further propose an approximation based on finite difference (FD) which not only reduces the computational complexity, but also makes the computation *parallelizable*.

Theorem 1 *If $L_{\mathcal{S}}(\boldsymbol{w})$ is 2-times-differentiable at $\boldsymbol{w}$, with $\boldsymbol{v} \sim N(0, I)$, by finite difference we have*

$$\begin{cases} \boldsymbol{v}^\top \nabla L_{\mathcal{S}}(\boldsymbol{w}) = \frac{1}{2\rho}(L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) - L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v})) + o\left(\epsilon^2\right); \\ \boldsymbol{v}^\top \nabla^2 L_{\mathcal{S}}(\boldsymbol{w})\boldsymbol{v} = \dfrac{1}{\rho^2}(L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) + L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v}) \\ \qquad\qquad - 2L_{\mathcal{S}}(\boldsymbol{w})) + o\left(\epsilon^3\right). \end{cases}$$

Proof: Using Taylor polynomial expansion of $L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v})$ and $L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v})$ centered at $\boldsymbol{w}$. We have

$$\begin{cases} L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) = L_{\mathcal{S}}(\boldsymbol{w}) + \rho\boldsymbol{v}\nabla L_{\mathcal{S}}(\boldsymbol{w}) + O\left(\rho^2\right); \\ L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v}) = L_{\mathcal{S}}(\boldsymbol{w}) - \rho\boldsymbol{v}\nabla L_{\mathcal{S}}(\boldsymbol{w}) + O\left(\rho^2\right). \end{cases} \tag{5.8}$$

Thus rearranging the above two qwuation we can obtain $\boldsymbol{v}^\top \nabla L_{\mathcal{S}}(\boldsymbol{w}) = \frac{1}{2\rho}(L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) - L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v})) + O\left(\rho^2\right)$.

We rewrite $\boldsymbol{v}^\top \nabla^2 L_{\mathcal{S}}(\boldsymbol{w})\boldsymbol{v}$ as directional derivatives as $\nabla^2_{\boldsymbol{v}} L_{\mathcal{S}}(\boldsymbol{w})$. Reapply the above formulation gives

$$\begin{aligned} \nabla^2_{\boldsymbol{v}} L_{\mathcal{S}}(\boldsymbol{w}) &= \frac{1}{\rho}(\nabla_{\boldsymbol{v}} L_{\mathcal{S}}(\boldsymbol{w}+0.5\rho\boldsymbol{v}) - \nabla_{\boldsymbol{v}} L_{\mathcal{S}}(\boldsymbol{w}-0.5\rho\boldsymbol{v})) \\ &\quad + O\left(\rho^2\right) \\ &= \frac{1}{\rho^2}(L_{\mathcal{S}}(\boldsymbol{w}+0.5\rho\boldsymbol{v}+0.5\rho\boldsymbol{v}) - L_{\mathcal{S}}(\boldsymbol{w} \\ &\quad + 0.5\rho\boldsymbol{v} - 0.5\rho\boldsymbol{v}) - L_{\mathcal{S}}(\boldsymbol{w}-0.5\rho\boldsymbol{v}+0.5\rho\boldsymbol{v}) \\ &\quad + L_{\mathcal{S}}(\boldsymbol{w}-0.5\rho\boldsymbol{v}-0.5\rho\boldsymbol{v})) + O\left(\rho^2\right) \\ &= \frac{1}{\rho^2}(L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) + L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v}) - 2L_{\mathcal{S}}(\boldsymbol{w})) \\ &\quad + O\left(\rho^2\right) \end{aligned}$$

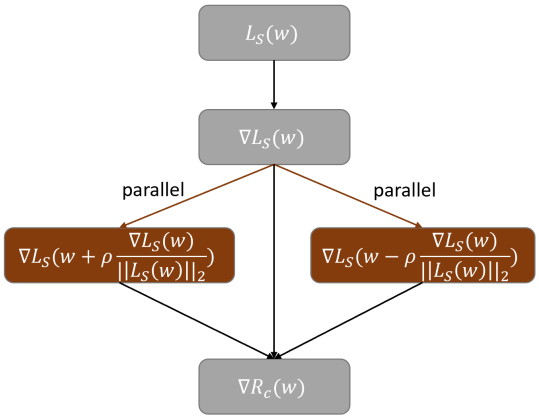


Figure 5.3. Computing the gradient of $R_c(\boldsymbol{w})$. The two gradient steps are independent of each other and can be perfectly parallelized.

By Theorem 1, we can instantiate $R_c(\boldsymbol{w})$ as:

$$R_c(\boldsymbol{w}) = \mathbb{E}_{\boldsymbol{v}\sim N(0,I)}\Big[\alpha \log\big(L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) + L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v}) - 2L_{\mathcal{S}}(\boldsymbol{w})\big) + \beta \log\big(L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) - L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v})\big)\Big] + const. \tag{5.9}$$

The above formulation involves an expectation over $\boldsymbol{v}$, which uniformly penalizes expected curvature across all directions. Previous studies [137, 138] highlight that gradient directions represent high-curvature directions. Hence, we choose to optimize over perturbations solely along gradient directions, approximating $R_c(\boldsymbol{w})$ by considering $\boldsymbol{v} = \nabla L_{\mathcal{S}}(\boldsymbol{w})$. Additionally, the terms $L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v})$ and $L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v})$ can be computed in parallel as shown in Figure 5.3.

We offer a meaningful interpretation of the finite difference regularizer (5.9): The second term within $R_c(\boldsymbol{w})$, i.e., $[L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) - L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v})]$, resembles the surrogate gap $[L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) - L_{\mathcal{S}}(\boldsymbol{w})]$ as introduced in [139]. However, unlike solely focusing on optimizing the ridge (locally worst-case perturbation) within the ρ-bounded neighborhood

Algorithm 4 Training with CR-SAM

Input: Training set $\mathcal{S}$; DNN model $f(x; \boldsymbol{w})$; Loss funtion $\ell\left(f\left(x_i; \boldsymbol{w}\right), y_i\right)$; Batch size B; Learning rate η; Perturbation size ρ; regularizer coefficients α and β
Output: model trained by CR-SAM

1: Parameter initialization $\boldsymbol{w}_0$.
2: **while** *not converged* **do**
3: Sample batch $\mathcal{B} = \{(x_i, y_i)\}_{i=0}^{B}$ from $\mathcal{S}$;
4: Compute $\boldsymbol{v} = \frac{\nabla L_{\mathcal{S}}(\boldsymbol{w})}{\|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2}$;
5: Compute $L_{\mathcal{S}}(\boldsymbol{w} + \rho \boldsymbol{v})$ and $L_{\mathcal{S}}(\boldsymbol{w} - \rho \boldsymbol{v})$;
6: Compute $\nabla R_c(\boldsymbol{w})$ per equation 5.9;
7: $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta(\nabla \mathcal{L}\left(\boldsymbol{w}_t\right) + R_c\left(\boldsymbol{w}_t\right))$;
8: **end while**
9: **return** $\boldsymbol{w}_t$

around the current parameter vector, our proposed regularizer also delves into the valley (locally best-case perturbation) of the DNN loss landscape, with their loss discrepancies similarly constrained by $R_c(\boldsymbol{w})$. Additionally, by expressing the first term within $R_c(\boldsymbol{w})$ as $[L_{\mathcal{S}}(\boldsymbol{w}+\rho \boldsymbol{v}) - L_{\mathcal{S}}(\boldsymbol{w})] - [L_{\mathcal{S}}(\boldsymbol{w}) - L_{\mathcal{S}}(\boldsymbol{w}-\rho \boldsymbol{v})]$, our approach encourages minimizing the disparity between the worst-case perturbed sharpness and the best-case perturbed sharpness. In essence, our strategy jointly optimizes the worst-case and best-case perturbations within the parameter space neighborhood, promoting a smoother, flatter loss landscape with fewer excessive wavy ridges and valleys.

The full pseudo-code of our CR-SAM training is given in Algorithm 4.

5.4. EXPERIMENTS

To assess CR-SAM, we conduct thorough experiments on prominent image classification benchmark datasets: CIFAR-10/CIFAR-100 and ImageNet-1k/-C/-R. Our evaluation encompasses a wide array of network architectures, including ResNet, WideResNet, Pyra-

midNet, and Vision Transformer (ViT), in conjunction with diverse data augmentation techniques. These experiments are implemented using PyTorch and executed on Nvidia A100 and V100 GPUs.

5.5. DETAILS OF EXPERIMENTAL SETUP

5.5.1. Training from Scratch on CIFAR-10 / CIFAR-100. In this section, we evaluate CR-SAM using the CIFAR-10/100 datasets [100]. Our evaluation encompasses a diverse selection of widely-used DNN architectures with varying depths and widths. Specifically, we employ ResNet-18 [6], ResNet-50 [6], Wide ResNet-28-10 (WRN-28-10) [140], and PyramidNet-110 [141], along with a range of data augmentation techniques, including basic augmentations (horizontal flip, padding by four pixels, and random crop) [87], Cutout [125], and AutoAugment [142], to ensure a comprehensive assessment. Following the setup in [117, 119], we train all models from scratch for 200 epochs, using batch size 128 and employing a cosine learning rate schedule. We conduct grid search to determine the optimal learning rate, weight decay, perturbation magnitude (ρ), coefficient (α and β) values that yield the highest test accuracy. To ensure a fair comparison, we run each experiment three times with different random seeds. Our experimental setup for training CIFAR10/100 from scratch is detailed in Table 5.1.

Results. Refer to Table 5.2 for a comprehensive overview. CR-SAM consistently outperforms both vanilla SAM and SGD across all configurations on both CIFAR-10 and CIFAR-100 datasets. Notable improvements are observed, such as a 1.11% enhancement on CIFAR-100 with ResNet-18 employing cutout augmentation and a 1.30% boost on CIFAR-100 with WRN-28-10 using basic augmentation. Furthermore, we empirically observe that CR-SAM exhibits a faster convergence rate in comparison to vanilla SAM. This accelerated convergence could be attributed to CR-SAM's ability to mitigate excessive curvature, ultimately reducing optimization complexity and facilitating swifter arrival at local minima.

Table 5.1. Hyperparameters of models ResNet-18, ResNet-101, Wide-28-10 and PyramidNet-110 for training from scratch on CIFAR10 and CIFAR100.

	CIFAR-10			CIFAR-100		
ResNet-18	SGD	SAM	CR-SAM	SGD	SAM	CR-SAM
Epoch		200			200	
Batch size		128			128	
Data augmentation		Basic			Basic	
Peak learning rate		0.05			0.05	
Learning rate decay		Cosine			Cosine	
Weight decay		5×10^{-3}			5×10^{-3}	
ρ	-	0.05	0.10	-	0.10	0.15
α	-	-	0.1	-	-	0.5
β	-	-	0.01	-	-	0.01
ResNet-101	SGD	SAM	CR-SAM	SGD	SAM	CR-SAM
Epoch		200			200	
Batch size		128			128	
Data augmentation		Basic			Basic	
Peak learning rate		0.05			0.05	
Learning rate decay		Cosine			Cosine	
Weight decay		5×10^{-3}			5×10^{-3}	
ρ	-	0.05	0.10	-	0.10	0.15
α	-	-	0.2	-	-	0.5
β	-	-	0.05	-	-	0.05
Wide-28-10	SGD	SAM	CR-SAM	SGD	SAM	CR-SAM
Epoch		200			200	
Batch size		128			128	
Data augmentation		Basic			Basic	
Peak learning rate		0.05			0.05	
Learning rate decay		Cosine			Cosine	
Weight decay		1×10^{-3}			1×10^{-3}	
ρ	-	0.10	0.10	-	0.10	0.15
α	-	-	0.5	-	-	0.5
β	-	-	0.1	-	-	0.1
PyramidNet-110	SGD	SAM	CR-SAM	SGD	SAM	CR-SAM
Epoch		200			200	
Batch size		128			128	
Data augmentation		Basic			Basic	
Peak learning rate		0.05			0.05	
Learning rate decay		Cosine			Cosine	
Weight decay		5×10^{-3}			5×10^{-3}	
ρ	-	0.15	0.20	-	0.15	0.20
α	-	-	0.5	-	-	0.5
β	-	-	0.1	-	-	0.1

Table 5.2. Results on CIFAR-10 and CIFAR-100. The base optimizer for SAM and CR-SAM is SGD with Momentum (SGD+M).

		CIFAR-10			CIFAR-100		
Model	Aug	SGD	SAM	CR-SAM	SGD	SAM	CR-SAM
ResNet-18	Basic	$95.29_{\pm 0.16}$	$96.46_{\pm 0.18}$	$\mathbf{96.95}_{\pm 0.13}$	$78.34_{\pm 0.22}$	$79.81_{\pm 0.18}$	$\mathbf{80.76}_{\pm 0.21}$
	Cutout	$95.96_{\pm 0.13}$	$96.55_{\pm 0.15}$	$\mathbf{97.01}_{\pm 0.21}$	$79.23_{\pm 0.13}$	$80.15_{\pm 0.17}$	$\mathbf{81.26}_{\pm 0.19}$
	AA	$96.33_{\pm 0.15}$	$96.75_{\pm 0.18}$	$\mathbf{97.27}_{\pm 0.12}$	$79.05_{\pm 0.17}$	$81.26_{\pm 0.21}$	$\mathbf{82.11}_{\pm 0.22}$
ResNet-101	Basic	$96.35_{\pm 0.12}$	$96.51_{\pm 0.16}$	$\mathbf{97.14}_{\pm 0.11}$	$80.54_{\pm 0.13}$	$82.11_{\pm 0.12}$	$\mathbf{83.03}_{\pm 0.17}$
	Cutout	$96.56_{\pm 0.18}$	$96.95_{\pm 0.13}$	$\mathbf{97.51}_{\pm 0.24}$	$81.26_{\pm 0.21}$	$82.39_{\pm 0.27}$	$\mathbf{83.46}_{\pm 0.16}$
	AA	$96.78_{\pm 0.14}$	$97.11_{\pm 0.16}$	$\mathbf{97.76}_{\pm 0.16}$	$81.83_{\pm 0.37}$	$83.25_{\pm 0.47}$	$\mathbf{84.19}_{\pm 0.23}$
WRN-28-10	Basic	$95.89_{\pm 0.21}$	$96.81_{\pm 0.26}$	$\mathbf{97.36}_{\pm 0.15}$	$81.84_{\pm 0.13}$	$83.15_{\pm 0.14}$	$\mathbf{84.45}_{\pm 0.09}$
	Cutout	$96.89_{\pm 0.07}$	$97.55_{\pm 0.16}$	$\mathbf{97.98}_{\pm 0.21}$	$81.96_{\pm 0.40}$	$83.47_{\pm 0.15}$	$\mathbf{84.48}_{\pm 0.13}$
	AA	$96.93_{\pm 0.12}$	$97.59_{\pm 0.06}$	$\mathbf{97.94}_{\pm 0.08}$	$82.16_{\pm 0.11}$	$83.69_{\pm 0.26}$	$\mathbf{84.74}_{\pm 0.21}$
PyramidNet-110	Basic	$96.27_{\pm 0.13}$	$97.34_{\pm 0.13}$	$\mathbf{97.89}_{\pm 0.08}$	$83.27_{\pm 0.12}$	$84.89_{\pm 0.09}$	$\mathbf{85.68}_{\pm 0.14}$
	Cutout	$96.79_{\pm 0.13}$	$97.61_{\pm 0.21}$	$\mathbf{98.08}_{\pm 0.11}$	$83.43_{\pm 0.21}$	$84.97_{\pm 0.17}$	$\mathbf{85.86}_{\pm 0.21}$
	AA	$96.97_{\pm 0.08}$	$97.81_{\pm 0.13}$	$\mathbf{98.26}_{\pm 0.11}$	$84.59_{\pm 0.08}$	$85.76_{\pm 0.23}$	$\mathbf{86.58}_{\pm 0.14}$

5.5.2. Training from Scratch on ImageNet-1k/-C/-R. This section details our evaluation on the ImageNet dataset [143], containing 1.28 million images across 1000 classes. We assess the performance of ResNet [6] and Vision Transformer (ViT) [144] architectures. Evaluation is extended to out-of-distribution data, namely ImageNet-C [145] and ImageNet-R [146]. ResNet50, ResNet101, ViT-S/32, and ViT-B/32 are evaluated with Inception-style preprocessing. For ResNet models, SGD serves as the base optimizer. We follow the setup in [116], training ResNet50 and ResNet101 with batch size 512 for 90 epochs. The initial learning rate is set to 0.1, progressively decayed using a cosine schedule. For ViT models, we adopt AdamW [147] as the base optimizer with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. ViTs are trained with batch size 512 for 300 epochs. Our experimental setup for training ImageNet from scratch is detailed in Table 5.3.

Results. Summarized in Table 5.4, our results indicate substantial accuracy improvements across various DNN models, including ResNet and ViT, on the ImageNet dataset. Notably, CR-SAM's performance surpasses that of SAM by 1.16% for ResNet-50 and by 1.77% for ViT-B/32. These findings underscore the efficacy of our CR-SAM approach.

Table 5.3. Hyperparameters of models ResNet-50, ResNet-101, ViT-S/32 and ViT-B/32 for training on ImageNet from scratch.

	ResNet-50			**ResNet-101**		
ImageNet	SGD	SAM	CR-SAM	SGD	SAM	CR-SAM
Epoch		90			90	
Batch size		512			512	
Data augmentation		Inception-style			Inception-style	
Peak learning rate		1.3			1.3	
Learning rate decay		Cosine			Cosine	
Weight decay		3×10^{-5}			3×10^{-5}	
ρ	-	0.10	0.15	-	0.10	0.15
α	-	-	0.1	-	-	0.2
β	-	-	0.01	-	-	0.01
	ViT-S/32			**ViT-B/32**		
ImageNet	SGD	SAM	CR-SAM	SGD	SAM	CR-SAM
Epoch		300			300	
Batch size		512			512	
Data augmentation		Inception-style			Inception-style	
Peak learning rate		3×10^{-3}			3×10^{-3}	
Learning rate decay		Cosine			Cosine	
Weight decay		0.3			0.3	
ρ	-	0.05	0.10	-	0.05	0.10
α	-	-	0.05	-	-	0.05
β	-	-	0.01	-	-	0.01

5.5.3. Model Geometry Analysis. CR-SAM aims to reduce the normalized trace of the Hessian to promote flatter minima. Empirical validation of CR-SAM's ability to locate optima with lower curvature is presented through model geometry comparisons among models trained by SGD, SAM, and CR-SAM (see Table 5.5). Our analysis is based on ResNet-18 trained on CIFAR-100 for 200 epochs using the three optimization methods. Hutchinson's method [135, 148] is utilized to compute the Hessian trace, with values obtained from the test set across three independent runs. Notably, the results reveal that CR-SAM significantly reduces both gradient norms and Hessian traces throughout training in contrast to SGD and SAM. This reduction contributes to a smaller normalized Hessian trace, affirming the effectiveness of our proposed regularization strategy.

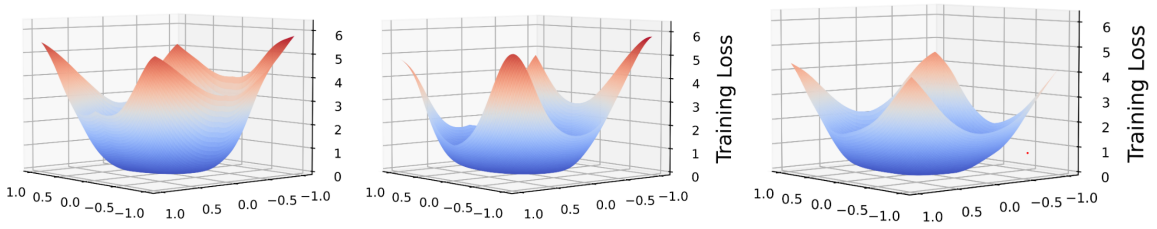
Table 5.4. Results on ImageNet, the base optimizer for ResNets and ViTs are SGD+M and AdamW, respectively.

Model	Datasets	Vanilla	SAM	R-SAM	CR-SAM
ResNet-50	ImageNet-1k	75.94	76.48	76.89	**77.64**
	ImageNet-C	43.64	46.03	46.19	**46.94**
	ImageNet-R	21.93	23.13	22.89	**23.48**
ResNet-101	ImageNet-1k	77.81	78.64	78.71	**79.12**
	ImageNet-C	48.56	51.27	51.35	**51.87**
	ImageNet-R	24.38	25.89	25.91	**26.37**
ViT-S/32	ImageNet-1k	68.40	70.23	70.39	**71.68**
	ImageNet-C	43.21	45.78	45.92	**46.46**
	ImageNet-R	19.04	21.12	21.35	**21.98**
ViT-B/32	ImageNet-1k	71.25	73.51	74.06	**75.28**
	ImageNet-C	44.37	46.98	47.28	**48.12**
	ImageNet-R	23.12	24.31	24.53	**25.04**

Table 5.5. Model geometry of ResNet-18 models trained with SGD, SAM and CR-SAM, values are computed on test set.

Optimizer	$\|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2$	$\mathrm{Tr}\left(\nabla^2 L_{\mathcal{S}}(\boldsymbol{w})\right)$	$C(\boldsymbol{w})$	Accuracy (%)
SGD	19.97 ± 0.52	32673.88 ± 1497.56	1674.89 ± 78.69	78.34 ± 0.22
SAM	11.51 ± 0.31	14176.52 ± 327.69	1193.87 ± 59.18	79.81 ± 0.18
CR-SAM	**8.26** ± 0.19	**7968.19** ± 145.73	**884.95** ± 23.59	**80.76** ± 0.21

5.5.4. Visualization of Landscapes. We visualize the flatness of minima obtained using CR-SAM by plotting loss landscapes of PyramidNet110 trained with SGD, SAM, and CR-SAM on CIFAR-100 for 200 epochs. Employing the visualization techniques from [149], we depict loss values along two randomly sampled orthogonal Gaussian perturbations around local minima. As depicted in 5.4, the visualization illustrates that CR-SAM yields flatter minima compared to SGD and SAM.



(a) SGD (b) SAM (c) CR-SAM

Figure 5.4. CR-SAM yields flatter loss landscape which is consistent with better generalization as verified in experiments.

5.5.5. Faster and Smoother Convergence. The convergence in a single run of SAM vs. CR-SAM is presented in 5.5. From the figure, we observe that CR-SAM achieves much faster and stabler convergence, which can be explained by the fact that CR-SAM discourages excessive curvature and thus reduces optimization complexity, making the local minimum easier to reach.

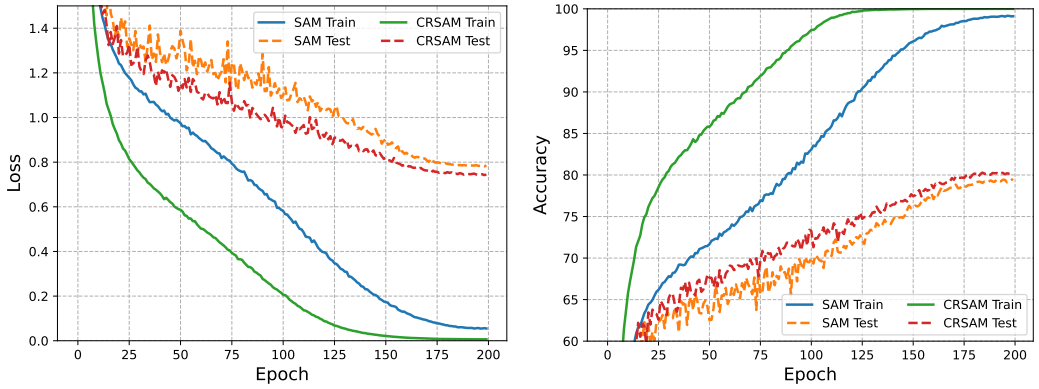


Figure 5.5. Evolution of training and testing loss/accuracy on CIFAR100 trained with ResNet18 by SAM and our proposed CR-SAM.

5.6. SUMMARY

In this section, we identify the limitations of the one-step gradient ascent in SAM's inner maximization during training due to the excessive non-linearity of the loss landscape. In addition, existing curvature metrics lack the ability to precisely capture the loss function geometry. To address these issues, we introduce normalized Hessian trace, which offers consistent and accurate characterization of loss function curvature on both training and test data. Building upon this metric, we present CR-SAM, a novel training approach for enhancing neural network generalizability by regularizing our proposed curvature metric. Additionally, to mitigate the overhead of computing the Hessian trace, we incorporate a parallelizable finite difference method. Our comprehensive experiments that span a wide variety of model architectures across popular image classification datasets including CIFAR10/100 and ImageNet-1k/-C/-R, affirm the effectiveness of our proposed CR-SAM training strategy.

6. CONCLUSION

This research delves deeply into investigating the transferability of adversarial examples, approaching the subject from data, optimization, and model perspectives. Through an intricate exploration, we aim to unravel the underlying mechanisms that govern the phenomenon of adversarial transferability, shedding light on its intricacies and implications across various dimensions of machine learning research and practice.

In section 2, from the data perspective, we propose a new method of crafting transferable AE which consists of two techniques: *elastic momentum* (EM) and *random erasure* (RE). EM generalizes the conventional momentum and the Nesterov's momentum methods by computing gradients over a flexible look-ahead horizon, and RE erase part of image with random noise which increases the diversity of adversarial perturbations and helps stabilize gradient fluctuations. Through extensive evaluation with 5 recent baseline methods, 7 target deep learning models, and 9 advanced defense mechanisms, we demonstrate the superior transferability of our proposed approach.

In section 3, we explore from optimization perspective by penalizing the input gradient norm, aim to identify AE within flat regions of the loss landscape. Our approach, known as the input *gradient norm penalty (GNP)*, has been demonstrated to substantially enhance adversarial transferability across a diverse array of deep networks. Furthermore, we illustrate that GNP can seamlessly integrate with existing transfer-based attacks, yielding even more impressive performance, thus showcasing its highly desirable flexibility.

In section 4, unlike previous approaches that primarily focus on the AE generation process itself, we investigate from the model perspective and propose a novel strategy centered on transforming surrogate models. By modifying these surrogate models to possess specific properties conducive to adversarial transferability, existing transfer-based black-box AE generation methods can operate on our transformed surrogate models without any modifications. To our knowledge, this is the first work to establish a connection

between the inner properties of surrogate models and AE transferability. We identify three such properties that enhance adversarial transferability: smaller local Lipschitz constant, smoother loss landscape, and stronger adversarial robustness. This approach offers valuable insights into understanding the factors influencing adversarial transferability.

In section 5, we introduce the normalized Hessian trace, a metric capable of accurately and consistently characterizing the curvature of loss landscapes. Leveraging this metric, we propose CR-SAM, a novel optimization technique that integrates curvature regularization into the Sharpness-Aware Minimization (SAM) optimizer. By doing so, we enhance the generalizability of deep neural networks, leading to improved performance across various tasks and datasets.

Overall, this research contributes to a deeper understanding of adversarial transferability and generalization, offering novel insights and techniques that advance the field of machine learning and contribute to the development of more robust and reliable AI systems.

REFERENCES

[1] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

[2] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *International Conference on Learning Representations*, 2020.

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

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[5] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[6] 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*, pages 770–778, 2016.

[7] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[9] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[10] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[11] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015.

[12] Joel Janai, Fatma Güney, Aseem Behl, Andreas Geiger, et al. Computer vision for autonomous vehicles: Problems, datasets and state of the art. *Foundations and Trends® in Computer Graphics and Vision*, 12(1–3):1–308, 2020.

[13] Filippo Pesapane, Marina Codari, and Francesco Sardanelli. Artificial intelligence in medical imaging: threat or opportunity? radiologists again at the forefront of innovation in medicine. *European radiology experimental*, 2:1–10, 2018.

[14] Guosheng Hu, Yongxin Yang, Dong Yi, Josef Kittler, William Christmas, Stan Z Li, and Timothy Hospedales. When face recognition meets with deep learning: an evaluation of convolutional neural networks for face recognition. In *Proceedings of the IEEE international conference on computer vision workshops*, pages 142–150, 2015.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[16] Bonan Min, Hayley Ross, Elior Sulem, Amir Pouran Ben Veyseh, Thien Huu Nguyen, Oscar Sainz, Eneko Agirre, Ilana Heintz, and Dan Roth. Recent advances in natural language processing via large pre-trained language models: A survey. *ACM Computing Surveys*, 56(2):1–40, 2023.

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[18] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[19] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

[20] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

[21] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[22] Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. Ieee, 2013.

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

[24] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations*, 2015.

[25] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *International Conference on Learning Representations, Workshop Track Proceedings*, 2017.

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

[27] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[28] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. *International Conference on Machine Learning*, 2018.

[29] Yandong Li, Lijun Li, Liqiang Wang, Tong Zhang, and Boqing Gong. Nattack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks. *International Conference on Machine Learning*, 2019.

[30] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *International Conference on Learning Representations*, 2018.

[31] Weilun Chen, Zhaoxiang Zhang, Xiaolin Hu, and Baoyuan Wu. Boosting decision-based black-box adversarial attacks with random sign flip. In *Proceedings of the European Conference on Computer Vision*, 2020.

[32] Yurii Nesterov. A method for unconstrained convex minimization problem with the rate of convergence o(1/kˆ2). *Doklady AN USSR*, 269:543–547, 1983.

[33] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[34] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204*, 2017.

[35] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *International Conference on Learning Representations*, 2017.

[36] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. *International Conference on Learning Representations*, 2018.

[37] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M Roy. A study of the effect of jpg compression on adversarial images. *arXiv preprint arXiv:1608.00853*, 2016.

[38] Weilin Xu, David Evans, and Yanjun Qi. Feature squeezing: Detecting adversarial examples in deep neural networks. *Network and Distributed System Security Symposium*, 2018.

[39] Chuan Guo, Mayank Rana, Moustapha Cisse, and Laurens Van Der Maaten. Countering adversarial images using input transformations. *International Conference on Learning Representations*, 2018.

[40] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1787, 2018.

[41] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML*, 2018.

[42] Jeremy M Cohen, Elan Rosenfeld, and J Zico Kolter. Certified adversarial robustness via randomized smoothing. *International Conference on Machine Learning*, 2019.

[43] Yuchen Zhang and Percy Liang. Defending against whitebox adversarial attacks via randomized discretization. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 684–693. PMLR, 2019.

[44] Xuanqing Liu, Minhao Cheng, Huan Zhang, and Cho-Jui Hsieh. Towards robust neural networks via random self-ensemble. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 369–385, 2018.

[45] Tianyu Pang, Kun Xu, Chao Du, Ning Chen, and Jun Zhu. Improving adversarial robustness via promoting ensemble diversity. In *International Conference on Machine Learning*, pages 4970–4979. PMLR, 2019.

[46] Zeyu Qin, Yanbo Fan, Yi Liu, Li Shen, Yong Zhang, Jue Wang, and Baoyuan Wu. Boosting the transferability of adversarial attacks with reverse adversarial perturbation. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL `https://openreview.net/forum?id=k5uFiFLWv3X`.

[47] 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*, pages 2730–2739, 2019.

[48] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019.

[49] Lei Wu and Zhanxing Zhu. Towards understanding and improving the transferability of adversarial examples in deep neural networks. In *Asian Conference on Machine Learning*, pages 837–850. PMLR, 2020.

[50] Xiaosen Wang, Xuanran He, Jingdong Wang, and Kun He. Admix: Enhancing the transferability of adversarial attacks. *arXiv preprint arXiv:2102.00436*, 2021.

[51] Weibin Wu, Yuxin Su, Xixian Chen, Shenglin Zhao, Irwin King, Michael R Lyu, and Yu-Wing Tai. Boosting the transferability of adversarial samples via attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1161–1170, 2020.

[52] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[53] Sizhe Chen, Zhengbao He, Chengjin Sun, Jie Yang, and Xiaolin Huang. Universal adversarial attack on attention and the resulting dataset damagenet. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):2188–2197, 2020.

[54] Nathan Inkawhich, Wei Wen, Hai Helen Li, and Yiran Chen. Feature space perturbations yield more transferable adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7066–7074, 2019.

[55] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *International Conference on Learning Representations*, 2017.

[56] Martin Gubri, Maxime Cordy, Mike Papadakis, Yves Le Traon, and Koushik Sen. Lgv: Boosting adversarial example transferability from large geometric vicinity. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part IV*, pages 603–618. Springer, 2022.

[57] 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.

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

[59] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJgIPJBFvH.

[60] Tengyuan Liang, Tomaso Poggio, Alexander Rakhlin, and James Stokes. Fisher-rao metric, geometry, and complexity of neural networks. In *The 22nd international conference on artificial intelligence and statistics*, pages 888–896. PMLR, 2019.

[61] Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey Hinton. Regularizing neural networks by penalizing confident output distributions, 2017. URL https://openreview.net/forum?id=HkCjNI5ex.

[62] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.

[63] Lei Wu, Mingze Wang, and Weijie Su. The alignment property of sgd noise and how it helps select flat minima: A stability analysis. *Advances in Neural Information Processing Systems*, 35:4680–4693, 2022.

[64] Lei Wu, Mingze Wang, and Weijie Su. When does sgd favor flat minima? a quantitative characterization via linear stability. *arXiv preprint arXiv:2207.02628*, 2022.

[65] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pages 1019–1028. PMLR, 2017.

[66] Simran Kaur, Jeremy Cohen, and Zachary Chase Lipton. On the maximum hessian eigenvalue and generalization. In *Proceedings on*, pages 51–65. PMLR, 2023.

[67] Wesley J Maddox, Gregory Benton, and Andrew Gordon Wilson. Rethinking parameter counting in deep models: Effective dimensionality revisited. *arXiv preprint arXiv:2003.02139*, 2020.

[68] Tao Wu, Tie Luo, and Donald C Wunsch. Black-box attack using adversarial examples: A new method of improving transferability. *World Scientific Annual Review of Artificial Intelligence*, 1:2250005, 2023.

[69] Tao Wu, Tie Luo, and Donald C Wunsch. Gnp attack: Transferable adversarial examples via gradient norm penalty. In *2023 IEEE International Conference on Image Processing (ICIP)*, pages 3110–3114. IEEE, 2023.

[70] Tao Wu, Tie Luo, and Donald C Wunsch II. Lrs: Enhancing adversarial transferability through lipschitz regularized surrogate. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 6135–6143, 2024.

[71] Tao Wu, Tie Luo, and Donald C Wunsch II. Cr-sam: Curvature regularized sharpness-aware minimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 6144–6152, 2024.

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

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

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

[75] Junhua Zou, Zhisong Pan, Junyang Qiu, Xin Liu, Ting Rui, and Wei Li. Improving the transferability of adversarial examples with resized-diverse-inputs, diversity-ensemble and region fitting. In *European Conference on Computer Vision*, pages 563–579. Springer, 2020.

[76] 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*, 2019.

[77] Yiwen Guo, Qizhang Li, and Hao Chen. Backpropagating linearly improves transferability of adversarial examples. In *NeurIPS*, 2020.

[78] Goran Nakerst, John Brennan, and Masudul Haque. Gradient descent with momentum — to accelerate or to super-accelerate? *arXiv preprint arXiv:2001.06472*, 2020.

[79] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13001–13008, 2020.

[80] Xiaolong Wang, Abhinav Shrivastava, and Abhinav Gupta. A-fast-rcnn: Hard positive generation via adversary for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2606–2615, 2017.

[81] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. *AAAI Conference on Artificial Intelligence*, 2017.

[82] 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*, pages 770–778, 2016.

[83] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. *International Conference on Learning Representations*, 2018.

[84] Zihao Liu, Qi Liu, Tao Liu, Nuo Xu, Xue Lin, Yanzhi Wang, and Wujie Wen. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 860–868. IEEE, 2019.

[85] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. Comdefend: An efficient image compression model to defend adversarial examples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6084–6092, 2019.

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

[87] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=6Tm1mposlrM`.

[88] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[89] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

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

[91] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[92] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

[93] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, 2018.

[94] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, 2017.

[95] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.

[96] Chenxi Liu, Barret Zoph, Maxim Neumann, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *ECCV*, 2018.

[97] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *CVPR*, 2019.

[98] Qizhang Li, Yiwen Guo, and Hao Chen. Yet another intermediate-leve attack. In *ECCV*, 2020.

[99] Yifeng Xiong, Jiadong Lin, Min Zhang, John E Hopcroft, and Kun He. Stochastic variance reduced ensemble adversarial attack for boosting the adversarial transferability. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14983–14992, 2022.

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

[101] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *CVPR*, pages 5927–5935, 2017.

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

[103] Yiwen Guo, Qizhang Li, Wangmeng Zuo, and Hao Chen. An intermediate-level attack framework on the basis of linear regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[104] Yao-Yuan Yang, Cyrus Rashtchian, Hongyang Zhang, Russ R Salakhutdinov, and Kamalika Chaudhuri. A closer look at accuracy vs. robustness. *Advances in neural information processing systems*, 33:8588–8601, 2020.

[105] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=H1oyRlYgg`.

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

[107] Chris Finlay, Adam M Oberman, and Bilal Abbasi. Improved robustness to adversarial examples using lipschitz regularization of the loss. 2018.

[108] Bohang Zhang, Du Jiang, Di He, and Liwei Wang. Rethinking lipschitz neural networks and certified robustness: A boolean function perspective. In *Advances in Neural Information Processing Systems*, 2022.

[109] Jacob Springer, Melanie Mitchell, and Garrett Kenyon. A little robustness goes a long way: Leveraging robust features for targeted transfer attacks. *Advances in Neural Information Processing Systems*, 34, 2021.

[110] Aaron Oord, Yazhe Li, Igor Babuschkin, Karen Simonyan, Oriol Vinyals, Koray Kavukcuoglu, George Driessche, Edward Lockhart, Luis Cobo, Florian Stimberg, et al. Parallel wavenet: Fast high-fidelity speech synthesis. In *International conference on machine learning*, pages 3918–3926. PMLR, 2018.

[111] Jacob Devlin Ming-Wei Chang Kenton and Lee Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019.

[112] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural computation*, 9(1): 1–42, 1997.

[113] Gintare Karolina Dziugaite and Daniel M Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. *arXiv preprint arXiv:1703.11008*, 2017.

[114] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30, 2017.

[115] Yong Liu, Siqi Mai, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Towards efficient and scalable sharpness-aware minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12360–12370, 2022.

[116] Jiawei Du, Hanshu Yan, Jiashi Feng, Joey Tianyi Zhou, Liangli Zhen, Rick Siow Mong Goh, and Vincent Tan. Efficient sharpness-aware minimization for improved training of neural networks. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=n0OeTdNRG0Q.

[117] Jiawei Du, Daquan Zhou, Jiashi Feng, Vincent Tan, and Joey Tianyi Zhou. Sharpness-aware training for free. *Advances in Neural Information Processing Systems*, 35: 23439–23451, 2022.

[118] Weisen Jiang, Hansi Yang, Yu Zhang, and James Kwok. An adaptive policy to employ sharpness-aware minimization. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=6Wl7-M2BC-.

[119] Yong Liu, Siqi Mai, Minhao Cheng, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Random sharpness-aware minimization. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=htUvh7xPoa.

[120] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision transformers outperform resnets without pre-training or strong data augmentations. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=LtKcMgGOeLt.

[121] Dara Bahri, Hossein Mobahi, and Yi Tay. Sharpness-aware minimization improves language model generalization. *arXiv preprint arXiv:2110.08529*, 2021.

[122] Peng Mi, Li Shen, Tianhe Ren, Yiyi Zhou, Xiaoshuai Sun, Rongrong Ji, and Dacheng Tao. Make sharpness-aware minimization stronger: A sparsified perturbation approach. *Advances in Neural Information Processing Systems*, 35:30950–30962, 2022.

[123] Andrew G Wilson and Pavel Izmailov. Bayesian deep learning and a probabilistic perspective of generalization. *Advances in neural information processing systems*, 33:4697–4708, 2020.

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

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

[126] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[127] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[128] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.

[129] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.

[130] Anders Krogh and John Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4, 1991.

[131] Harris Drucker and Yann Le Cun. Improving generalization performance using double backpropagation. *IEEE transactions on neural networks*, 3(6):991–997, 1992.

[132] Yang Zhao, Hao Zhang, and Xiuyuan Hu. Penalizing gradient norm for efficiently improving generalization in deep learning. In *International Conference on Machine Learning*, pages 26982–26992. PMLR, 2022.

[133] Jure Sokolić, Raja Giryes, Guillermo Sapiro, and Miguel RD Rodrigues. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65 (16):4265–4280, 2017.

[134] Yucong Liu, Shixing Yu, and Tong Lin. Hessian regularization of deep neural networks: A novel approach based on stochastic estimators of hessian trace. *Neurocomputing*, 536:13–20, 2023. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2023.03.017. URL https://www.sciencedirect.com/science/article/pii/S0925231223002515.

[135] Haim Avron and Sivan Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM (JACM)*, 58(2):1–34, 2011.

[136] Suraj Srinivas, Kyle Matoba, Himabindu Lakkaraju, and François Fleuret. Efficient training of low-curvature neural networks. *Advances in Neural Information Processing Systems*, 35:25951–25964, 2022.

[137] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Stefano Soatto. Empirical study of the topology and geometry of deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3762–3770, 2018.

[138] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9078–9086, 2019.

[139] Juntang Zhuang, Boqing Gong, Liangzhe Yuan, Yin Cui, Hartwig Adam, Nicha Dvornek, Sekhar Tatikonda, James Duncan, and Ting Liu. Surrogate gap minimization improves sharpness-aware training. *arXiv preprint arXiv:2203.08065*, 2022.

[140] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

[141] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5927–5935, 2017.

[142] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[143] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[144] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[145] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=HJz6tiCqYm`.

[146] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, et al. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8340–8349, 2021.

[147] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=Bkg6RiCqY7`.

[148] Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE international conference on big data (Big data)*, pages 581–590. IEEE, 2020.

[149] Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31, 2018.

VITA

Tao Wu was born in China and earned his Bachelor's degree in June 2018 from Huazhong University of Science and Technology, Wuhan, China. After that, he embarked on his academic journey at the Computer Science department at Missouri University of Science and Technology under the co-supervision of Dr. Donald C. Wunsch II and Dr. Tie Luo. He received his Ph.D. degree in Computer Science in July 2024 from Missouri University of Science and Technology. His major research interests included computer vision, optimization and adversarial machine learning.