Towards Interpretable Adversarial Examples via Sparse Adversarial Attack

Fudong Lin[1], Jiadong Lou[1], Hao Wang[2], Brian Jalaian[3], and Xu Yuan[1] (✉)

[1] University of Delaware, Newark, DE 19716, USA
{fudong,loujd,xyuan}@udel.edu
[2] Stevens Institute of Technology, Hoboken, NJ 07030, USA
hwang9@stevens.edu
[3] University of West Florida, Pensacola, FL 32514, USA bjalaian@uwf.edu

Abstract. Sparse attacks are to optimize the magnitude of adversarial perturbations for fooling deep neural networks (DNNs) involving only a few perturbed pixels (*i.e.*, under the l_0 constraint), suitable for interpreting the vulnerability of DNNs. However, existing solutions fail to yield interpretable adversarial examples due to their poor sparsity. Worse still, they often struggle with heavy computational overhead, poor transferability, and weak attack strength. In this paper, we aim to develop a sparse attack for understanding the vulnerability of DNNs by minimizing the magnitude of initial perturbations under the l_0 constraint, to overcome the existing drawbacks while achieving a fast, transferable, and strong attack to DNNs. In particular, a novel and *theoretical sound* parameterization technique is introduced to approximate the NP-hard l_0 optimization problem, making directly optimizing sparse perturbations computationally feasible. Besides, a novel loss function is designed to augment initial perturbations by maximizing the adversary property and minimizing the number of perturbed pixels simultaneously. Extensive experiments are conducted to demonstrate that our approach, with theoretical performance guarantees, outperforms state-of-the-art sparse attacks in terms of computational overhead, transferability, and attack strength, expecting to serve as a benchmark for evaluating the robustness of DNNs. In addition, theoretical and empirical results validate that our approach yields sparser adversarial examples, empowering us to discover two categories of noises, *i.e.*, "obscuring noise" and "leading noise", which will help interpret how adversarial perturbation misleads the classifiers into incorrect predictions. Our code is available at https://github.com/fudong03/SparseAttack.

Keywords: Sparse Attack · Adversarial Attack · Interpretability.

1 Introduction

Deep neural networks (DNNs) have demonstrated impressive performance on a range of challenging tasks, including image classification [21,42,16,12], natural language processing [48,8], and various other domains [18,17,26,39,28,31,1,24,55,5]. However, recent studies [45,15] have revealed a critical vulnerability: DNNs can be easily fooled by adversarial examples. These examples are generated by adding small, human-imperceptible perturbations to natural images, causing the models to make incorrect predictions with

high confidence. This vulnerability leads to severe security threats on DNNs, *e.g.*, a prior study [13] reported that adding human-imperceptible perturbation to a *Stop* sign made state-of-the-art classifiers misclassify it as a Speed Limit 45, thereby hindering DNNs' wide applicability to such security-critical domains as face recognition [22,46], autonomous driving [13,19], *etc*.

The mainstream attack strategy targeting DNNs is to optimize the magnitude of adversarial perturbations. In general, a perturbation is constrained by l_p norm, with $p = 0, 1, 2$, or ∞, and can be categorized into two clusters, *i.e.*, dense attack and sparse attack. The former needs to modify almost all pixels under the l_2 or l_∞ constriant [9], while the latter perturbs a few pixels under the l_0 (or sometimes l_1) constriant [57].

To date, Fast Gradient Sign Method (FGSM)-based approaches [15,22,10,53,11] are known to be prominent dense attacks, because they arrive at fast and highly transferable adversarial attacks by optimizing adversarial perturbations under the l_∞ constraint. However, they tend to perturb almost all pixels, making them hard to interpret adversarial attacks due to their overly perturbed adversarial examples. In sharp contrast, sparse attacks [43,35,14,9,57,38,6,51,34,49] minimize the l_0 distance between natural images and adversarial examples, for attacking DNNs with only a few perturbed pixels. Hence, sparse attacks usually provide additional insights into adversarial attacks, able to better interpret the vulnerability of DNNs [14]. However, optimizing the magnitude of perturbations under the l_0 constraint falls into the NP-hard problem, so previous solutions often get trapped in local optima [35,57], making the resulted attacks possess an insufficient adversary property. As such, existing sparse attacks suffer from the drawbacks of heavy computational overhead [9], poor transferability, and weak attack intensity. Worse still, their resultant adversarial examples suffer from poor sparsity, making them unsuitable for interpreting the vulnerability of DNNs.

In this work, we focus on the sparse attack, aiming to develop a new solution that yields interpretable adversarial examples, allowing us to have a deep understanding in the vulnerability of DNNs. To achieve our goal, we introduce a novel and *theoretically solid* reparameterization technique to effectively approximate the NP-hard l_0 optimization problem, making direct optimization of sparse perturbations computationally tractable. In addition, a novel loss function is proposed to augment initial perturbations through maximizing the adversary property and minimizing the number of perturbed pixels simultaneously. As such, our approach, underpinned by theoretical performance guarantees, can yield a fast, transferable, and powerful adversarial attack while unveiling the mystery underlying adversarial perturbations. Extensive experimental results demonstrate that our approach outperforms state-of-the-art sparse attacks in terms of computational complexity, transferability, and attack strength. Meanwhile, we theoretically and empirically validate that our approach yields much sparser adversarial examples, suitable for interpreting the vulnerability of DNNs. Through analyzing the minimal perturbed adversarial examples, we discover two categories of adversarial perturbations to help understand how adversarial perturbations mislead the classifiers, resulted directly from "obscuring noise" and "leading noise", where the former obscures the classifiers from identifying true classes, while the latter misleads the classifiers into targeted predictions.

2 Related Work

The state-of-the-art solutions for adversarial attacks can be grouped into two categories, *i.e.*, dense attack and sparse attack. We shall discuss how our work relates to, and differs from, prior solutions.

Dense attacks optimize the magnitude of adversarial perturbations under the l_2 or l_∞ constriant. Popularized by Fast Gradient Sign Method (FGSM) [15], FGSM-based methods are the most prominent dense attacks, where adversarial examples (under the l_∞ constraint) are effectively produced by adding the gradients of the classification loss to natural images. Subsequent work includes I-FGSM [22] which applies FGSM in multiple rounds, R-FGSM [47] which augments FGSM with random initialization, PGD [32] which extends I-FGSM with multiple random restarts, MI-FGSM [10] which boosts I-FGSM with momentum, and DI-FGSM [53] and TI-FGSM [11] which improve transferability respectively with random resizing and translation operations. Other dense attacks include [45,36,2,4,3,33,27], which perform effective dense attacks by minimizing the l_2 (or l_∞) distance between natural images and adversarial examples. However, this category of solutions requires modification of almost all pixels, infeasible to be used for interpreting the vulnerability of DNNs. Our solution, by contrast, perturbs only a few pixels, able to provide additional insights about adversarial vulnerability.

Sparse attacks minimize the magnitude of perturbations under the l_0 (or sometimes l_1) constraint. Previous sparse attacks include C&W L_0 attack [2] which iteratively fixes less important pixels, OnePixel [43] which applies an evolutionary algorithm, SparseFool [35] which converts the l_0 optimization problem to the l_1 constraint one, GreedyFool [9] which uses a two-stage greedy strategy, Homotopy attack [57] which utilizes a homotopy algorithm to jointly optimize the sparsity and the perturbation bound, among many others [37,54,7,14,38,6,51,34,49]. Unfortunately, prior sparse attacks suffer from considerable computational overhead, poor transferability, and weak attack intensity. In contrast, we propose a theoretical sound reparameterization technique to approximate the NP-hard l_0 optimization problem and a novel loss function to augment initial perturbations. As such, our approach advances existing sparse attacks in terms of computational efficiency, transferability, and attack strength. In addition, we theoretically and empirically validate that our approach yields much sparser adversarial examples, empowering us to interpret the vulnerability of DNNs.

3 Preliminary

Given a well-trained classifier $f_{\boldsymbol{\theta}}$, the cross-entropy loss function J, and a natural image $\boldsymbol{x}$ with the ground-truth label y_{true}, the adversary aims to mislead the classifier $f_{\boldsymbol{\theta}}$ into an incorrect prediction via adding certain perturbation $\boldsymbol{\delta}$ (under the constraint ϵ) into the natural image, mathematically expressed as follows:

$$f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}) = y_{\text{adv}} \quad \text{s.t. } \|\boldsymbol{\delta}\|_p \leq \epsilon \text{ and } \boldsymbol{x}+\boldsymbol{\delta} \in [0,1]^{\text{d}}, \tag{1}$$

where y_{adv} represents the adversarial label and is different from y_{true}, $\|\cdot\|_p$ denotes the l_p norm (with $p = 0, 1, 2$, or ∞), and manipulating a natural image should yield one valid image. Note here the pixel values are normalized over $[0, 1]$ to simplify the calculation.

FGSM (Fast Gradient Sign Method) [15] aims to mislead classifiers to predict incorrectly through adding gradients to natural images. The sign function is leveraged to ensure the perturbation $\boldsymbol{\delta}$ under the l_∞ constraint of ϵ, *i.e.*,

$$\boldsymbol{\delta} = \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x}}\, J(\boldsymbol{x},\, y_{\text{true}})) \quad \text{s.t. } f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}) = y_{\text{adv}} \text{ and } \boldsymbol{x}+\boldsymbol{\delta} \in [0,1]^d. \tag{2}$$

FGSM is deemed as the fastest attack algorithm [10]. Subsequent solutions [22,32,53,11] have been proposed, yielding fast and highly transferable dense attacks.

I-FGSM (Iterative Fast Gradient Sign Method) [22] augments FGSM to have a stronger white-box attack by applying a small step length α to FGSM iteratively:

$$\boldsymbol{x}_0^{\text{adv}} = \boldsymbol{x}, \;\; \boldsymbol{x}_{N+1}^{\text{adv}} = \boldsymbol{x}_N^{\text{adv}} + \alpha \cdot \text{sign}(\nabla_{\boldsymbol{x}}\, J(\boldsymbol{x},\, y_{\text{true}})). \tag{3}$$

We can simply set $\alpha = \epsilon/T$ (T is the number of iterations) to satisfy the l_∞ constraint ϵ. Note that Eq. (3) performs a non-target attack by adding positive gradients to natural images $\boldsymbol{x}$ for maximizing the classification loss. To perform a targeted attack, we can maximize the logical probability of y_{adv} on natural image $\boldsymbol{x}$ (*i.e.*, $\log p(y_{\text{adv}}|\boldsymbol{x})$) by iteratively moving towards the direction of $\text{sign}\{\nabla_{\boldsymbol{x}}\ \log p(y_{\text{adv}}|\boldsymbol{x})\}$:

$$\boldsymbol{x}_0^{\text{adv}} = \boldsymbol{x}, \;\; \boldsymbol{x}_{N+1}^{\text{adv}} = \boldsymbol{x}_N^{\text{adv}} - \alpha \cdot \text{sign}(\nabla_{\boldsymbol{x}}\, J(\boldsymbol{x},\, y_{\text{adv}})). \tag{4}$$

The intuition behind Eq. (4) is that adding negative gradients to the natural image $\boldsymbol{x}$ can make the prediction of classifier $f_{\boldsymbol{\theta}}$ iteratively move towards the adversarial class y_{adv}. Notably, both Eq. (3) and Eq. (4) must also satisfy the constraint $\boldsymbol{x}_{N+1}^{\text{adv}} \in [0,1]^d$ to ensure that the resulting adversarial examples remain valid input images.

4 Our Approach

4.1 Problem Statement

Sparse attacks optimize the magnitude of perturbations under the l_0 constraint, aiming to achieve successful attacks with a small number of perturbed pixels, as formulated below:

$$\text{minimize } \|\boldsymbol{\delta}\|_0 \quad \text{s.t. } f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{\delta}) = y_{\text{adv}} \text{ and } \boldsymbol{x}+\boldsymbol{\delta} \in [0,1]^{\text{d}}. \tag{5}$$

Unfortunately, Eq. (5) is an NP-hard problem. When solving it, prior sparse attacks often got trapped in local optima [7,57], causing the resulting attacks to suffer from poor sparsity, unable to be used for interpreting the vulnerability of DNNs. To address this limitation, we develop a sparse attack that yields highly sparse adversarial examples, suitable for understanding the vulnerability of DNNs.

4.2 Challenges

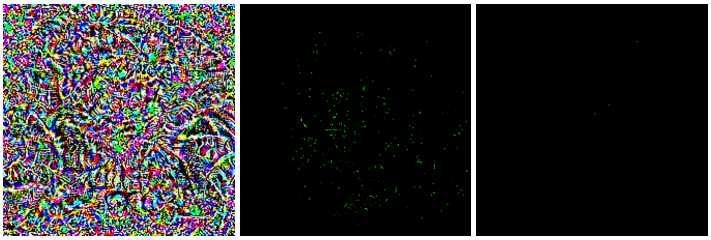
Several challenges are to be addressed, as elaborated below.

Box Constraint. "Box constraint" is fundamental to adversarial attacks. It ensures that manipulating natural images should yield valid images (*i.e.*, $\boldsymbol{x}+\boldsymbol{\delta} \in [0,1]^{\text{d}}$). Two strategies exist to meet the box constraint for sparse attacks: i) *clipping invalid pixels*

(a) I-FGSM
"mink"
98.6% confidence

(b) BruSLeAttack
"panda"
13.5% confidence

(c) Ours
"mink"
97.9% confidence



(d) perturbation
$\|\boldsymbol{\delta}\|_0 = 150523$
(99.99%)
0.52 seconds

(e) perturbation
$\|\boldsymbol{\delta}\|_0 = 425$
(0.282%)
36.9 seconds

(f) perturbation
$\|\boldsymbol{\delta}\|_0 = 4$
(0.003%)
3.86 seconds

Fig. 1: Illustration of adversarial examples (AEs) computed by different attack algorithms. (a), (b), and (c) show AEs computed by I-FGSM, BruSLeAttack, and our approach, respectively, dependent on the classifier VGG-16. (d), (e), and (f) exhibit the adversarial perturbation with respect to (a), (b), and (c). The classification results, *i.e.*, "mink", "panda", "mink" (below (a), (b), and (c), respectively) and classification confidence levels, are reported by VGG-19. The number of ($\|\boldsymbol{\delta}\|_0$) and the percentage of perturbed pixels, as well as the computation time, are all listed below (d), (e), and (f).

which exceed the valid range and ii) *changing the optimization direction* to make resulting adversarial examples valid. However, the former results in a severe reduction in the fooling rate, while the latter incurs a large computational burden. So far, how to effectively handle the "box constraint" with desired properties remains challenging.

Computational efficiency. Optimizing the magnitude of perturbations under the l_0 constraint is NP-hard. Prior sparse attacks attempt different approximation algorithms to reduce the computational burden, *e.g.*, BruSLeAttack [49]. Yet, considerable computational overhead still incurs (see Figure 1e, where 36.9 seconds are taken). Hence, how to efficiently minimize the number of perturbed pixels remains open.

Transferability and Attack Strength. Prior sparse attacks often suffer from poor transferability and weak attack intensity. For example, as shown in Figure 1b, BruSLeAttack fails to perform a black-box attack, *i.e.*, the adversarial example generated by VGG-16

fails to fool VGG-19. It is challenging to develop a transferable and powerful sparse attack with only a small amount of perturbed pixels.

4.3 Our Idea

To overcome the aforementioned challenges, we aim to develop a novel sparse attack by optimizing the magnitudes of initial perturbations generated by I-FGSM under the l_0 constraint. In particular, a novel loss function is introduced to augment initial perturbations through maximizing the adversary property and minimizing the number of perturbed pixels simultaneously. Two observations motivate this idea. *First*, a prior study [29] reported that different models tend to learn similar decision boundaries. Hence, starting from initial perturbations generated by dense attacks accelerates convergence and makes our approach more likely to reach the global optima [44]. *Second*, I-FGSM results in overly perturbed adversarial examples, and thereby reducing the number of perturbed pixels may not negatively affect its adversary property. For example, as depicted in Figure 1a (I-FGSM) and Figure 1c (Our Approach), both I-FGSM and our approach mislead VGG-19 into the same incorrect prediction (*i.e.*, "mink"). Therefore, it is feasible to optimize the magnitude of perturbations produced by I-FGSM under the l_0 constraint without sacrificing its transferability. Furthermore, a new box constraint strategy is designed to make our solution always yield valid adversarial examples.

4.4 Our Proposed Approach

Objective. To achieve our goal, we propose a novel sparse attack by optimizing the magnitude of initial perturbations under the l_0 constraint. Our problem can be formulated as follows:

$$\text{minimize } \|\boldsymbol{w}\|_0 \quad \text{s.t. } f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{w}\odot\boldsymbol{\delta}) = y_{\text{adv}} \text{ and } \boldsymbol{x}+\boldsymbol{w}\odot\boldsymbol{\delta} \in [0,1]^{\text{d}}, \tag{6}$$

where $\boldsymbol{\delta}$ indicates the initial perturbation, $\boldsymbol{w}$ denotes the weight with the same dimension as perturbation $\boldsymbol{\delta}$, and $\odot$ represents an element-wise product. Motivated by the "pre-train then tune" paradigm [56], instead of randomly initializing the perturbation $\boldsymbol{\delta}$, we pre-compute it by using *Iterative Fast Gradient Sign Method* (I-FGSM) [22], which in turn accelerates convergence and makes the resulting perturbation more likely to reach the global optima [44]. Note that solving the problem formulated by Eq. (6) yields near-optimal solutions for Eq. (5), *i.e.*, with the fewest perturbed pixels.

To augment initial perturbations, we follow prior work [2,29] to reformulate our problem of Eq. (6) via the Lagrangian relaxation formulation for concurrently maximizing the classification loss and minimizing the number of perturbed pixels, yielding:

$$J_{\text{adv}} = J(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\boldsymbol{w}\odot\boldsymbol{\delta}),\ y_{\text{adv}}) + \lambda\,\|\boldsymbol{w}\|_0, \tag{7}$$

where λ is a hyperparameter to balance the classification loss and the degree of perturbation. As negative elements in $\boldsymbol{w}$ indicate that the corresponding perturbations in $\boldsymbol{\delta}$ negatively affect the adversary property, a simple but effective way to augment Eq. (7) is via applying the ReLU function to drop negative values in $\boldsymbol{w}$, *i.e.*,

$$J_{\text{adv}} = J(f_{\boldsymbol{\theta}}(\boldsymbol{x}+\pi(\boldsymbol{w})\odot\boldsymbol{\delta}),\ y_{\text{adv}}) + \lambda\,\|\pi(\boldsymbol{w})\|_0, \tag{8}$$

where $\pi(\cdot)$ represents the ReLU function.

Reparameterization Technique. However, as indicated by [30], the l_0 norm of $\pi(\boldsymbol{w})$ is non-differentiable, so directly optimizing Eq. (8) is computationally intractable. To address this intractability, let $H(\cdot)$ be the Heaviside step function and consider a simple re-parametrization technique:

$$\|\pi(\boldsymbol{w})\|_0 = \sum_{j=1}^{d} H(\pi(w_j)), \text{ with } H(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}. \tag{9}$$

As such, we can reformulate Eq. (8) as follows,

$$J_{\text{adv}} = J(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \pi(\boldsymbol{w}) \odot \boldsymbol{\delta}),\ y_{\text{adv}}) + \lambda \sum_{j=1}^{d} H\left(\pi(w_j)\right). \tag{10}$$

Obviously, penalizing the second term of Eq. (10) is equivalent to penalize the l_0 norm of $\pi(\boldsymbol{w})$. However, it is impractical to directly optimize Eq. (10) because the distributional derivative of the Heaviside step function, *i.e.*, the Dirac delta function, equals to zero almost everywhere [50]. The Dirac delta function, by definition, can be simply regarded as a function that is zero everywhere except at the origin, where it is infinite, *i.e.*,

$$p(x) \simeq \begin{cases} \infty, & x = 0 \\ 0, & x \neq 0 \end{cases}. \tag{11}$$

To make the Heaviside step function differentiable, we devise a novel reparametrization technique to approximate the Dirac delta function, resorting to the zero-centered normal distribution presented as follows:

$$q_a(x) = \frac{1}{|a|\sqrt{\pi}} \exp^{-(x/a)^2}. \tag{12}$$

Here, the variance of $q_a(x)$ is determined by the hyperparameter a. When the hyperparameter a approaches zero, the function $q_a(x)$ converges to the Dirac delta function $p(x)$, as stated next.

Theorem 1. *(Convergence) Let $p(x)$ denote the Dirac delta function. Consider $q_a(x)$ defined as $q_a(x) = \frac{1}{|a|\sqrt{\pi}} \exp^{-(x/a)^2}$, which represents a zero-centered normal distribution with variance dependent on the hyperparameter a. Then, in the distributional sense, we have:*

$$\lim_{a \to 0} q_a(x) = p(x). \tag{13}$$

Proof. First, consider the case when $x = 0$, we always have:

$$\lim_{a \to 0} q_a(x) = \lim_{a \to 0} \frac{1}{|a|\sqrt{\pi}} \exp^{-(0/a)^2} = \lim_{a \to 0} \frac{1}{|a|\sqrt{\pi}} = \infty. \tag{14}$$

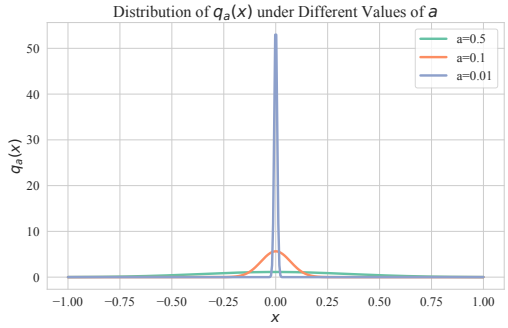


Fig. 2: Distribution of $q_a(x)$ under different values of a. As a approaches to 0, $q_a(x)$ increasingly resembles the Dirac delta function.

Second, when $x \neq 0$, we need to consider both the positive and negative directions of a, *i.e.*, $\lim_{a\to 0^+}$ and $\lim_{a\to 0^-}$. Starting with the positive direction, let $t = \frac{1}{a}$, we have

$$\begin{aligned}\lim_{a\to 0^+} q_a(x) &= \lim_{a\to 0} \frac{1}{a\sqrt{\pi}} \exp^{-(x/a)^2} = \lim_{t\to\infty} \sqrt{\pi} t \cdot \exp^{-x^2t^2} = \lim_{t\to\infty} \frac{\sqrt{\pi}t}{\exp^{x^2t^2}} \\ &= \lim_{t\to\infty} \frac{\sqrt{\pi}}{2x^2t \cdot \exp^{x^2t^2}} \text{ (L'Hopital's rule)} = \frac{\sqrt{\pi}}{\infty} = 0.\end{aligned} \tag{15}$$

Similarly, for the negative direction of a, we have:

$$\begin{aligned}\lim_{a\to 0^-} q_a(x) &= \lim_{a\to 0} -\frac{1}{a\sqrt{\pi}} \exp^{-(x/a)^2} = \lim_{t\to\infty} -\sqrt{\pi} t \cdot \exp^{-x^2t^2} \\ &= \lim_{t\to\infty} -\frac{\sqrt{\pi}t}{\exp^{x^2t^2}} = \lim_{t\to\infty} -\frac{\sqrt{\pi}}{2x^2t \cdot \exp^{x^2t^2}} = -\frac{\sqrt{\pi}}{\infty} = 0.\end{aligned} \tag{16}$$

Based on the above discussion, we have:

$$\lim_{a\to 0} q_a(x) = \begin{cases} \infty, & x = 0 \\ 0, & x \neq 0 \end{cases} = p(x). \tag{17}$$

Figure 2 presents the distribution of $q_a(x)$ for various values of a. It is clear that as a approaches zero, the shape of $q_a(x)$ more closely resembles the Dirac delta function, a crucial aspect in estimating the derivative of the Heaviside step function. This estimation is significant as it allows for the differentiability of Eq. (10), demonstrated by:

$$\frac{dH}{dx} \approx \frac{1}{|a|\sqrt{\pi}} \exp^{-(x/a)^2}. \tag{18}$$

Here, the hyperparameter a modulates the balance between optimization smoothness and approximation accuracy.

Performance Guarantees. A simple trick to increase the sparsity for convergence acceleration is to tailor the ReLU function as follows:

$$\pi'(\boldsymbol{w}) = \pi(\boldsymbol{w} - \frac{\tau}{\epsilon}), \tag{19}$$

where ϵ is the l_∞ constraint for I-FGSM and τ is used to simplify hyperparameter tuning. By using our tailored ReLU function $\pi'(\cdot)$, we can re-write the loss function given below:

$$J_{\text{adv}} = J(f_{\boldsymbol{\theta}}(\boldsymbol{x} + \pi(\boldsymbol{w} - \frac{\tau}{\epsilon}) \odot \boldsymbol{\delta}),\ y_{\text{adv}}) + \ \lambda \sum_{j=1}^{d} H\left(\pi(w_j - \frac{\tau}{\epsilon})\right). \tag{20}$$

By employing Eq. (20), our approach yields much sparser adversarial examples, as stated below.

Proposition 1 *(Sparsity) Given an initial perturbation $\boldsymbol{\delta} \in [-\epsilon, \epsilon]^d$ under some l_∞ constraint ϵ ($\epsilon > 0$), let $\boldsymbol{w} \in \mathbb{R}^d$ be the weight matrix used in our study, and $\pi(\cdot)$ be the ReLU function, in terms of the l_0 norm, we have:*

$$\|\pi(\boldsymbol{w} - \frac{\tau}{\epsilon}) \odot \boldsymbol{\delta}\|_0 \ \leq \ \|\pi(\boldsymbol{w}) \odot \boldsymbol{\delta}\|_0 \ \leq \ \|\boldsymbol{w} \odot \boldsymbol{\delta}\|_0. \tag{21}$$

The proof of Proposition 1 is deferred to Appendix A.1 for conserving sparse.

Although Eq. (20) can yield more sparse adversarial examples, it cannot guarantee valid pixel values, *i.e.*, $\exists\ w_j \in \boldsymbol{w}, x_j + \pi(w_j - \frac{\tau}{\epsilon}) \odot \delta_j \notin [0, 1]$. To remedy this, we devise a new strategy to satisfy the box constraint:

$$\boldsymbol{w}' = \boldsymbol{\Omega} \odot H\left(\pi(\boldsymbol{w} - \frac{\tau}{\epsilon})\right), \tag{22}$$

where $\boldsymbol{\Omega}$ serves to impose a tight bound on the resulting perturbation. Following this step, we are able to obtain valid adversarial examples, as stated next.

Proposition 2 *(Box constraint) Let $\boldsymbol{x} \in [0, 1]^d$ be a natural image, $\boldsymbol{\delta} \in [-\epsilon, \epsilon]^d$ be the initial perturbation under a l_∞ constraint ϵ ($\epsilon > 0$), and $\boldsymbol{w}'$ be the output of Eq. (22). If $\boldsymbol{\Omega} = \min(\frac{\boldsymbol{x}}{\epsilon}, \frac{1-\boldsymbol{x}}{\epsilon})$, the resulting adversarial example is valid, i.e.,*

$$\boldsymbol{x} + \boldsymbol{w}' \odot \boldsymbol{\delta} \in [0, 1]^d. \tag{23}$$

The proof of Proposition 2 is deferred to Appendix A.2. Note that such a box constraint strategy can automatically yield valid pixels, making adversarial attacks escape from local optima, as stated in the prior study [2]. Details of our algorithm flow are deferred to Appendix B.

5 Experiments and Results

5.1 Experimental Settings

Datasets. We experimentally benchmark our approach on three widely used image datasets: i) $70,000$ greyscale examples of **MNIST** [25]; ii) $60,000$ RGB examples of

CIFAR-10 [20]; iii) $10,000$ RGB examples randomly selected from the **ImageNet** [40] validation set.

Compared Methods. We compare our approach to ten sparse attack counterparts, *i.e.*, **C&W L_0** [2], **OnePixel** [43], **SparseFool** [35], **GreedyFool** [9], **FMN** [38], **Homotopy** [57], **Sparse-RS** [6], **SA-MOO** [51], **EGS-TSSA** [34] and **BruSLeAttack** [49]. Hyperparameters for compared methods, if not specified, are set as mentioned in their respective articles. All comparative results represent the average of 5 trials.

Parameter Settings. Four pre-trained models, VGG-16 [42], VGG-19, ResNet-101 [16], and ResNet-152, are exploited to evaluate the adversarial perturbation under ImageNet. The hyperparameters for I-FGSM, unless specified otherwise, are set as $\epsilon = 4/255$, $\alpha = 1/255$, and the number of iterations equal to 10. We optimize the weight in our approach by using SGD with a momentum of 0.9 and a learning rate of $1e-2$. A mini-batch of 256 is used for MNIST and CIFAR-10, and of 64 for ImageNet. We set $\lambda = 1e-2$ for MNIST and CIFAR-10, and $\lambda = 1e-3$ for ImageNet. We grid-search a (and τ) and empirically set them to 0.1 (and 0.30) for all three datasets. The number of iterations for our approach is set to 100, 100, and 200 for MNIST, CIFAR-10, and ImageNet, respectively. For the targeted attack, we follow prior studies [22,23] by setting the least-likely class as the targeted label. All experiments were conducted on a workstation equipped with an RTX 4090 GPU.

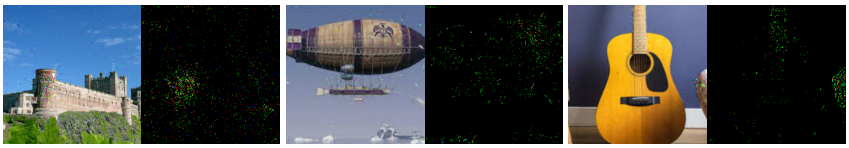
5.2 Evaluation on Sparsity

We compare our approach to sparse attacks listed in Section 5.1 in terms of sparsity (*i.e.*, l_0 norm of perturbation) under the non-targeted and targeted attack scenarios. ResNet-18 and VGG-16 are used to generate adversarial examples and report the classification results for CIFAR-10 and ImageNet, respectively. For all methods (except for OnePixel), we report the averagely required amount of perturbed pixels for achieving a fooling rate of 100% in both scenarios. Note that OnePixel cannot achieve such a fooling rate because its perturbations are limited to an extreme case. Table 1 presents the experimental results. Under the non-targeted attack scenario, we observe that except for OnePixel, our approach achieves the minimal magnitude of perturbations, with the averaged number of 44 (1.45% pixels) and of 57 (0.04% pixels) for CIFAR-10 and ImageNet, respectively. Although OnePixel outperforms our approach in terms of sparsity, it suffers from an extremely poor fooling rate, substantially inferior to our approach. Similarly, under the targeted attack scenario, our approach achieves the best sparsity of 69 (2.27% pixels) and of 136 (0.09% pixels) for CIFAR-10 and ImageNet, respectively, significantly outperforming all its counterparts. Notably, OnePixel and SpareFool cannot perform effective targeted attacks, so their results are unavailable.

Next, we conduct qualitative experiments to evaluate the sparsity under the targeted attack scenario. Figure 3 shows the visualized results, where our approach modifies only $123, 107$, and 218 pixels, respectively, with its adversarial examples able to mislead the classifier into targeted mispredictions. In contrast, C&W L_0, GreedyFool, and Homotopy have to perturb 3555, 1070, and 1731 pixels, respectively, to perform effective targeted attacks. These qualitative results demonstrate the effectiveness of our approach on sparsity. We also conduct qualitative comparison under the non-targeted attack scenario, with the results deferred to Appendix C.1 of supplementary materials.

Table 1: Comparative results of sparsity under CIFAR-10 and ImageNet, with the minimum amounts of perturbed pixels required to achieve the fooling rate (FR) of 100% for non-targeted and targeted attacks reported

Method	Non-targeted Attack				Targeted Attack			
	CIFAR-10		ImageNet		CIFAR-10		ImageNet	
	FR	Sparsity	FR	Sparsity	FR	Sparsity	FR	Sparsity
C&W L_0	100%	52	100%	424	100%	102	100%	5132
OnePixel	35.2%	1	20.5%	3	-	-	-	-
SparseFool	100%	76	100%	234	-	-	-	-
FMN	100%	106	100%	632	100%	183	100%	1087
Sparse-RS	100%	167	100%	758	100%	244	100%	1293
SA-MOO	100%	173	100%	1392	100%	305	100%	2896
EGS-TSSA	100%	105	100%	549	100%	218	100%	1054
BruSLeAttack	100%	93	100%	164	100%	148	100%	678
Ours	100%	44	100%	57	100%	69	100%	136



(a) C&W L_0 "wombat", $\|\boldsymbol{\delta}\|_0 = 3555$

(b) GreedyFool "garter snake", $\|\boldsymbol{\delta}\|_0 = 1070$

(c) Homotopy "tailed frog", $\|\boldsymbol{\delta}\|_0 = 1731$

(d) Ours "wombat", $\|\boldsymbol{\delta}\|_0 = 123$

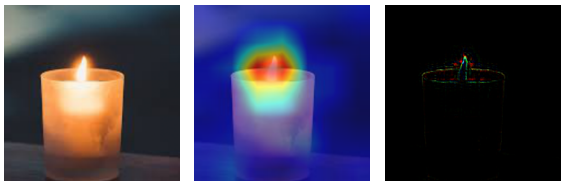
(e) Ours "garter snake", $\|\boldsymbol{\delta}\|_0 = 107$

(f) Ours "tailed frog", $\|\boldsymbol{\delta}\|_0 = 218$

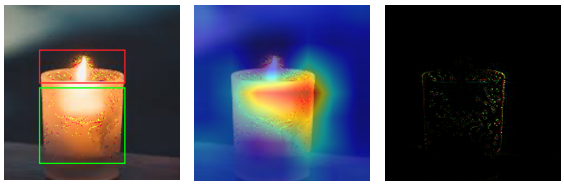
Fig. 3: Illustration of targeted attacks by sparse attack counterparts (Top) and our approach (Bottom). From left to right, the ground-truth classes are "castle", "airship", and "guitar", respectively. The incorrect predictions and the sparsity (*i.e.*, $\|\boldsymbol{\delta}\|_0$) are listed under each image.

5.3 Interpreting the Adversarial Perturbation

Despite extensive attention to sparse attacks, limited work has ever clearly explained how adversarial perturbations mislead DNNs into incorrect predictions. This is due to the mediocre performance results of prior sparse attacks on sparsity, as discussed in Section 5.2. We aim to fill this gap by unveiling the mystery underlying the proposed adversarial perturbation, resorting to Grad-CAM and Guided Grad-CAM visualizations [41]. Specifically, we let the prediction made by a well-trained VGG-16 as the decision of interest for Grad-CAM and Guided Grad-CAM visualizations. As such, we can inter-



(a) Clean image "candle" (b) Grad-CAM "candle" (c) Guided Grad-CAM "candle"



(d) Adversarial example "toilet tissue" (e) Grad-CAM "toilet tissue" (f) Guided Grad-CAM "toilet tissue"

Fig. 4: Illustration of how adversarial perturbations computed by our approach mislead VGG-16 to predict the label of "candle" to "toilet tissue". **Left:** clean image and adversarial example with their predicted labels. **Middle:** Grad-CAM visualization. **Right:** Guided Grad-CAM visualization.

pret adversarial perturbations by comparing visualizations on clean images with those on their corresponding adversarial examples.

Figure 4 exhibits the experimental results, where our generated perturbations mislead VGG-16 to mispredict "toilet tissue" on an image as "candle". In particular, Figures 4a, 4b and 4c show visualizations on the clean image. We observed that the "candle wick" is the critical region for making a correct prediction on the "candle". By contrast, Figures 4d, 4e and 4f depict visualizations of the adversarial example generated by our approach. From Figure 4d, we observe that the adversarial perturbation can be divided into two categories, namely, "obscuring noise" (*i.e.*, perturbations within the red box) and "leading noise" (*i.e.*, perturbations within the green box). The former prevents the classifier from identifying the true label by covering essential features of the true class, *i.e.*, obscuring the feature of "candle wick", whereas the latter leads the classifier to mispredict the adversarial example by adding the essential features of the targeted class, *i.e.*, adding noise within the shape of "toilet tissue"; see Figures 4b versus 4e and Figures 4c versus 4f.

We also conduct experiments to show how the two types of perturbations mislead ResNet-50 to make an incorrect prediction as "wing" on an image of "canoe", with the results deferred to Appendix C.2 of the supplementary materials. To the best of our knowledge, we are the first to interpret how adversarial perturbations mislead DNNs into incorrect predictions, by discovering the "obscuring noise" and "leading noise".

Table 2: The fooling rates (%) on different robust models trained adversarially by PGD-AT and Fast-AT, respectively, with the best results are shown in bold

Method	PGD-AT		Fast-AT		
	MNIST	CIFAR-10	MNIST	CIFAR-10	ImageNet
C&W L_0	11.8	52.8	12.1	62.3	67.8
GreedyFool	3.6	26.8	4.9	23.5	24.9
FMN	7.9	55.0	9.8	52.9	54.1
Homotopy	8.5	51.9	7.2	53.2	58.7
Sparse-RS	10.3	54.6	10.9	57.1	56.3
SA-MOO	11.5	56.9	7.8	58.9	62.8
EGS-TSSA	10.7	57.8	8.4	61.6	63.8
BruSLeAttack	11.2	61.2	9.6	58.9	65.1
Ours	**13.8**	**62.5**	**14.6**	**66.4**	**72.1**

5.4 Attacking Robust Models

We next compare our approach to sparse attacks in terms of attack strength, *i.e.*, the fooling rate on robust models trained adversarially by PGD Adversarial Training (PGD-AT) [32] or Fast Adversarial Training (Fast-AT) [52]. We set $\epsilon = 0.3$, $\epsilon = 8/255$, $\epsilon = 2/255$ for MNIST, CIFAR-10, and ImageNet, respectively. Table 2 presents the experimental results.

From Table 2, we observe that our approach achieves the most powerful attacks under all scenarios, with the best fooling rate of 14.6%, 66.4%, and 72.1% on MNIST, CIFAR-10, and ImageNet, respectively. The reason is that i) our novel loss function (*i.e.*, Eq. (20)) significantly augments the adversary property; and ii) the proposed theoretically sound box constraint strategy yields valid adversarial examples without the need of clipping invalid pixels, empowering our approach to generate more optimized solutions. Meanwhile, all sparse attacks, including our approach, achieve smaller fooling rates on MNIST than those on CIFAR-10 and ImageNet. This is because adversarial training can significantly improve the model's robustness on the simple dataset. But, for CIFAR-10 (or ImageNet), our approach can achieve the averaged fooling rate of 62.5% and of 66.4% (or 72.1%), respectively, on PGD-AT and Fast-AT.

5.5 Comparison on Transferability

We take two models, *i.e.*, VGG-19 and ResNet-101, to generate adversarial examples, which are then employed to attack four different classifiers, *i.e.*, VGG-16, VGG-19, ResNet-101, and ResNet-152. Table 3 presents the comparative results of our approach and sparse attack counterparts in terms of transferability. Notably, the adversarial examples generated via VGG-19 (or ResNet-101) to attack VGG-19 (or ResNet-101) are considered as white-box attacks, while all others are black-box attacks. We observe that compared to other sparse attacks, our approach achieves the best performance under all scenarios, with the largest fooling rate of 99.1% (or 99.9%) under the black-box (or white-box) attack. This demonstrates that modifying a few pixels is sufficient to make our approach enjoy high transferability.

Table 3: Comparison of transferability (*i.e.*, Mean Fooling Rate (%)) under ImageNet. The first column denotes the models where adversarial samples are generated while the first row indicates the target models for attacking, with the best results shown in bold and $*$ indicating white-box attacks

Model	Method	**VGG-16**	**VGG-19**	**ResNet-101**	**ResNet-152**
VGG-19	C&W L_0	84.6	96.2*	55.6	47.5
	GreedyFool	10.6	19.2*	8.2	9.0
	FMN	83.4	91.8*	43.0	46.8
	Homotopy	65.8	87.6*	24.0	9.8
	Sparse-RS	85.2	96.0*	46.8	48.2
	SA-MOO	86.2	92.8*	44.5	49.2
	EGS-TSSA	87.7	93.2*	51.3	51.2
	BruSLeAttack	89.4	94.4*	56.7	51.8
	Ours	**99.1**	**99.9***	**63.3**	**55.1**
ResNet-101	C&W L_0	82.6	81.9	91.2*	50.5
	GreedyFool	11.4	13.4	9.4*	8.4
	FMN	82.6	81.8	60.1*	41.6
	Homotopy	63.6	63.6	72.0*	46.2
	Sparse-RS	85.2	76.8	89.4*	65.6
	SA-MOO	86.6	84.3	91.1*	61.3
	EGS-TSSA	86.1	83.8	91.5*	62.6
	BruSLeAttack	86.8	84.2	92.6*	67.8
	Ours	**87.6**	**86.3**	**99.9***	**89.9**

Table 4: Comparing various sparse attacks in terms of computational complexity under different models, with the best results shown in bold

Method	Time Cost (s)			
	VGG-16	VGG-19	ResNet-101	ResNet-152
C&W L_0	20.8	23.4	28.7	43.6
Homotopy	519.4	531.5	1011.9	1462.2
Sparse-RS	69.0	76.4	71.6	99.7
SA-MOO	537.4	628.1	1123.8	1379.1
EGS-TSSA	71.8	79.5	120.9	175.9
BruSLeAttack	87.2	93.7	142.2	212.3
Ours	**4.6**	**5.3**	**18.7**	**27.5**

5.6 Comparison on Computational Complexity

We conduct the white-box attacks under ImageNet to show the superior computational efficiency of our approach for high-dimensional data. Four classifiers, *i.e.*, VGG-16, VGG-19, ResNet-101, and ResNet-152, are taken into account. Table 4 lists the experimental results. Clearly, our approach runs much faster than all sparse attack counterparts (*i.e.*, C&W L_0, Homotopy, Sparse-RS, SA-MOO, EGS-TSSA, and BruSLeAttack) for all examined classifiers, enjoying 3.0x, 62.8x, 4.6x, 64.0x, 8.0x, and 13.0x, and computational speedups on average. Specifically, our approach takes only $4.6s$, $5.3s$, $18.7s$, and $27.5s$, on VGG-16, VGG-19, ResNet-101, and ResNet-152, respectively. This is because our reparameterization technique (*i.e.*, Eq. (18)) can efficiently approximate the NP-hard l_0 optimization problem and hence substantially accelerate the convergence.

We also conducted ablation studies to exhibit the hyperparameter sensitivity of a, λ, and τ respectively in Eq. (18), Eq. (19), and Eq. (20), with their details deferred to Section C.3 of supplementary materials.

6 Conclusion

This paper has addressed a sparse attack that yields interpretable adversarial examples, thanks to their superb sparsity. Meanwhile, our approach enjoys fast convergence, high transferability, and powerful attack strength. The key idea is to approximate the NP-hard l_0 optimization problem via a theoretical sound reparameterization technique, making direct optimization of sparse perturbations computationally tractable. Besides, a novel loss function and a theoretically sound box constraint strategy have been proposed to make our solution generate superior adversarial examples, yielding fast, transferable, and powerful sparse adversarial attacks. Experimental results demonstrate that our approach clearly outperforms state-of-the-art sparse attacks in terms of computational efficiency, transferability, and attack intensity. In addition, theoretical and empirical results verify that our approach yields sparser adversarial examples, empowering us to experimentally interpret how adversarial examples mislead state-of-the-art DNNs into incorrect predictions. Our work is expected to i) serve as the benchmark for evaluating the robustness of DNNs, and ii) shed light on future work about interpreting the vulnerability of DNNs.

Acknowledgments

This work was supported in part by NSF under Grants 2019511, 2315613, 2325564, 2438898, 2523997, 2534286, 2315612, and 2332638. Any opinions and findings expressed in the paper are those of the authors and do not necessarily reflect the views of funding agencies.

References

1. Ahsan Ali, Xiaolong Ma, Syed Zawad, Paarijaat Aditya, Istemi Ekin Akkus, Ruichuan Chen, Lei Yang, and Feng Yan. Enabling scalable and adaptive machine learning training via serverless computing on public cloud. *Performance Evaluation*, 167:102451, 2025.
2. Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy*, 2017.
3. Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *IEEE Symposium on Security and Privacy, 2020*, 2020.
4. Lin Chen, Yifei Min, Mingrui Zhang, and Amin Karbasi. More data can expand the generalization gap between adversarially robust and standard models. In *International Conference on Machine Learning (ICML)*, 2020.
5. Tiankuo Chu, Fudong Lin, Shubo Wang, Jason Jiang, Wiley Jia-Wei Gong, Xu Yuan, and Liyun Wang. Bonemet: An open large-scale multi-modal murine dataset for breast cancer bone metastasis diagnosis and prognosis. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

6. Francesco Croce, Maksym Andriushchenko, Naman D. Singh, Nicolas Flammarion, and Matthias Hein. Sparse-rs: A versatile framework for query-efficient sparse black-box adversarial attacks. In *AAAI*, 2022.
7. Francesco Croce and Matthias Hein. Sparse and imperceivable adversarial attacks. In *ICCV*, 2019.
8. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, 2019.
9. Xiaoyi Dong, Dongdong Chen, Jianmin Bao, Chuan Qin, Lu Yuan, Weiming Zhang, Nenghai Yu, and Dong Chen. Greedyfool: Distortion-aware sparse adversarial attack. In *NeurIPS*, 2020.
10. Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.
11. Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
12. Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.
13. Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *CVPR*, 2018.
14. Yanbo Fan, Baoyuan Wu, Tuanhui Li, Yong Zhang, Mingyang Li, Zhifeng Li, and Yujiu Yang. Sparse adversarial attack via perturbation factorization. In *ECCV*, 2020.
15. Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2015.
16. Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
17. Yi He, Fudong Lin, Xu Yuan, and Nian-Feng Tzeng. Interpretable minority synthesis for imbalanced classification. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2542–2548, 2021.
18. John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.
19. Zelun Kong, Junfeng Guo, Ang Li, and Cong Liu. Physgan: Generating physical-world-resilient adversarial examples for autonomous driving. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
20. Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Technical Report, University of Toronto*, 2009.
21. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2012.
22. Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *International Conference on Learning Representations (ICLR)*, 2017.
23. Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial machine learning at scale. In *5th International Conference on Learning Representations (ICLR)*, 2017.

24. Guangchen Lan, Huseyin A. Inan, Sahar Abdelnabi, Janardhan Kulkarni, Lukas Wutschitz, Reza Shokri, Christopher G. Brinton, and Robert Sim. Contextual integrity in LLMs via reasoning and reinforcement learning. *arXiv preprint arXiv:2506.04245*, 2025.
25. Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.
26. Fudong Lin, Summer Crawford, Kaleb Guillot, Yihe Zhang, Yan Chen, Xu Yuan, et al. Mmst-vit: Climate change-aware crop yield prediction via multi-modal spatial-temporal vision transformer. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5751–5761, 2023.
27. Fudong Lin, Jiadong Lou, Xu Yuan, and Nian-Feng Tzeng. Towards robust vision transformer via masked adaptive ensemble. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1389–1399, 2024.
28. Fudong Lin, Xu Yuan, Yihe Zhang, Purushottam Sigdel, Li Chen, Lu Peng, and Nian-Feng Tzeng. Comprehensive transformer-based model architecture for real-world storm prediction. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD)*, pages 54–71, 2023.
29. Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *ICLR*, 2017.
30. Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through l_0 regularization. In *ICLR*, 2018.
31. Xiaolong Ma, Feng Yan, Lei Yang, Ian Foster, Michael E Papka, Zhengchun Liu, and Rajkumar Kettimuthu. Malletrain: Deep neural networks training on unfillable supercomputer nodes. In *Proceedings of the 15th ACM/SPEC International Conference on Performance Engineering*, pages 190–200, 2024.
32. Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.
33. Yifei Min, Lin Chen, and Amin Karbasi. The curious case of adversarially robust models: More data can help, double descend, or hurt generalization. In *Uncertainty in Artificial Intelligence*, 2021.
34. Di Ming, Peng Ren, Yunlong Wang, and Xin Feng. Transferable structural sparse adversarial attack via exact group sparsity training. In *Computer Vision and Pattern Recognition (CVPR)*, pages 24696–24705, 2024.
35. Apostolos Modas, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Sparsefool: A few pixels make a big difference. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
36. Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
37. Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *European Symposium on Security and Privacy (EuroS&P)*, 2016.
38. Maura Pintor, Fabio Roli, Wieland Brendel, and Battista Biggio. Fast minimum-norm adversarial attacks through adaptive norm constraints. In *Neural Information Processing Systems (NeurIPS)*, pages 20052–20062, 2021.
39. Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15), 2021.
40. Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and

Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 2015.
41. Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017.
42. Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015.
43. Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Trans. Evol. Comput.*, 2019.
44. Fnu Suya, Jianfeng Chi, David Evans, and Yuan Tian. Hybrid batch attacks: Finding black-box adversarial examples with limited queries. In *USENIX*, 2020.
45. Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.
46. Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: Adversarial patches to attack person detection. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.
47. Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations (ICLR)*, 2018.
48. Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NIPS)*, 2017.
49. Viet Quoc Vo, Ehsan Abbasnejad, and Damith Ranasinghe. Brusleattack: a query-efficient score- based black-box sparse adversarial attack. In *International Conference on Learning Representations(ICLR)*, 2024.
50. Wikipedia. Heaviside step function, 2024. [Online; accessed 01-January-2024].
51. Phoenix Neale Williams and Ke Li. Black-box sparse adversarial attack via multi-objective optimisation. In *Computer Vision and Pattern Recognition (CVPR)*, 2023.
52. Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. In *International Conference on Learning Representations (ICLR)*, 2020.
53. 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 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.
54. Kaidi Xu, Sijia Liu, Pu Zhao, Pin-Yu Chen, Huan Zhang, Quanfu Fan, Deniz Erdogmus, Yanzhi Wang, and Xue Lin. Structured adversarial attack: Towards general implementation and better interpretability. In *ICLR*, 2019.
55. Syed Zawad, Xiaolong Ma, Jun Yi, Cheng Li, Minjia Zhang, Lei Yang, Feng Yan, and Yuxiong He. Fedcust: Offloading hyperparameter customization for federated learning. *Performance Evaluation*, 167:102450, 2025.
56. Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, 2014.
57. Mingkang Zhu, Tianlong Chen, and Zhangyang Wang. Sparse and imperceptible adversarial attack via a homotopy algorithm. In *ICML*, 2021.