big data and cognitive computing

MDPI

Article

Improving Flat Maxima with Natural Gradient for Better Adversarial Transferability

Yunfei Long and Huosheng Xu *

College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China; 2016064109@hrbeu.edu.cn

* Correspondence: xuhuosheng@hrbeu.edu.cn

Abstract

Deep neural networks are vulnerable and susceptible to adversarial examples, which can induce erroneous predictions by injecting imperceptible perturbations. Transferability is a crucial property of adversarial examples, enabling effective attacks under black-box settings. Adversarial examples at flat maxima-those around which the loss peaks and grows slowly-have been demonstrated to exhibit higher transferability. Existing methods to achieve flat maxima rely on the gradient of the worst-case loss within the small neighborhood around the adversarial point. However, the neighborhood structure is typically defined as a Euclidean space, which neglects the input space's information geometry, leading to suboptimal results. In this work, we build upon the idea of flat maxima but extend the neighborhood structure from Euclidean space to the manifold measured by the Fisher metric, which takes into account the information geometry of the data space. In the non-Euclidean case, we search for the worst-case point in the direction of the natural gradient with respect to adversarial examples. The natural gradient adjusts the original gradient using the Fisher information matrix, giving the steepest direction in the manifold. Furthermore, to reduce the computational cost of calculating the Fisher information matrix, we introduce a diagonal approximation of the matrix and propose an empirical Fisher method under the model ensemble setting. Experimental results demonstrate that our proposed manifold extensions significantly enhance attack success rates against both normally and adversarially trained models. In particular, compared to methods relying on the Euclidean metric, our approach demonstrates more efficient performance.

Keywords: transferable adversarial attacks; flat loss landscape; information geometry; natural gradient

Academic Editor: Hui Tian

Received: 2 December 2025
Revised: 30 December 2025
Accepted: 5 January 2026
Published: 9 January 2026

1. Introduction

Numerous works [1–3] have demonstrated the vulnerability of Deep Neural Networks (DNNs) to adversarial examples, which are crafted by imposing human-imperceptible perturbations to natural examples, but can cause DNNs to produce incorrect predictions. Transferability is a central property of adversarial examples [4,5], which refers to the ability of adversarial examples generated against surrogate models to mislead other models. This property allows attacks to be performed under black-box settings, which can be exploited to evaluate the robustness of different models. Therefore, methods for generating high-transferable adversarial examples, known as transfer-based attacks, have gained increasing attention.

From the perspectives of optimization and generalization, many works have been proposed to improve the transferability of adversarial examples. Gradient optimization attacks [4–13] improve transferability by designing advanced gradient calculation algorithms to avoid falling into poor local maxima. Input transformation attacks [5,14–19] apply data augmentation strategies to inputs to avoid overfitting the surrogate model. Model ensemble attacks [4,20–22] generate adversarial examples for multiple surrogate models to reduce the generalization error [23], which is similar to training more data to improve model generalization. These methods treat the process of generating adversarial examples on the surrogate model as analogous to standard neural network training and leverage model generalization principles to guide the improvement of the transferability.

Inspired by the observation that flat local minima tend to result in better model generalization than their sharp counterparts [24–29], several studies demonstrate that adversarial examples at flat maxima tend to exhibit better transferability [8,9,22]. The loss landscape is formally defined as the geometry of the loss function with respect to the input. A flat maximum is characterized as an entire neighborhood having both high loss and low curvature in input space. In the context of model generalization, Sharpness-Aware Minimization (SAM) [24] is a seminal method for finding flat minima by minimizing the worst-case point within the neighborhood around the model instead of minimizing the original loss, where the worst-case point refers to a local loss maximum. Penalizing Gradient Norm (PGN) [27] is another method that can help the optimizer find flat minima by simultaneously minimizing the loss function and the gradient norm of the loss function. Existing works [8,22,30] have demonstrated that the generated adversarial examples exhibit higher transferability when gradient-based attacks are combined with SAM and PGN (denoted as SAM and PGN without ambiguity in the following). In the context of adversarial attack, SAM finds flat maxima by maximizing the worst-case point within the neighborhood around the input, where the worst-case point refers to a local loss minimum. PGN maximizes the loss function and minimizes the gradient norm of the loss function. We analyze the optimization objectives of SAM and PGN, and the intrinsic relationships between them when applied in adversarial attacks, from which we draw the corollary that the gradient of PGN can be expressed as a linear combination of the gradient of SAM with the gradient of the original loss function. This also implies that both SAM and PGN rely on the worst-case point within the neighborhood.

Despite their empirical success, these approaches implicitly assume that the neighborhood structure of the input space is Euclidean and that the standard gradient provides the steepest direction within this space. However, this assumption overlooks an important fact that DNNSs map inputs to discrete probability distributions over labels, and the input can be interpreted as a pullback mapping from the output space. The input space is rendered as a manifold with a non-linear Riemannian metric [31–34]. Under this perspective, following the standard gradients may fail to accurately find the worst-case point within the neighborhood around the input. The final found flat maxima would be suboptimal results.

In this paper, we follow the idea of flat maxima, but address the issue of the existing flat maxima attack approaches by redefining the neighborhood around adversarial examples as a manifold. Specifically, we consider the neighborhood boundary to be constrained by the KL divergence instead of the Euclidean metric, which geometrically conforms to the intrinsic property of the statistical manifold. This change in metrics suggests searching in the direction of the natural gradient for the worst-case point. The natural gradient adjusts the original gradient using the Fisher information matrix, which gives the steepest descent direction in the manifold with the Fisher metric. Based on the above analysis, it can be reasonable to assume that using natural gradients will find a more accurate worst-case. Therefore, optimizing with the gradient of this worst-case drives adversarial examples

toward flatter and more stable high-loss regions. For the Fisher information matrix of adversarial examples, we approximate it as the diagonal matrix and use the average gradient of the model ensemble to estimate. The SAM with manifold neighborhood, termed NG-SAM is implemented under the model ensemble setting. Combining with original loss and adjusting the balancing coefficients, PGN with manifold neighborhood, termed NG-PGN can be flexibly implemented. We show in Section 4.8 that NG-PGN outperforms NG-SAM, and we provide a detailed analysis of this observation in Section 4.10. Concisely, the NG-SAM strategy primarily guides adversarial examples toward flat regions of the loss landscape, without significantly increasing the loss value. NG-PGN balances the loss value with the flatness of the loss surface, achieving better attack performance. Therefore, NG-PGN is adopted as the default method in subsequent comparisons with state-of-the-art approaches. Furthermore, similar to other ensemble attacks, both NG-SAM and NG-PGN can be integrated with gradient optimization and input transformation methods, further boosting the adversarial transferability.

The contributions of the proposed work can be summarized as follows:

- We reveal the intrinsic relationship between the attack objectives of SAM and PGN in the context of adversarial attacks, demonstrating that the gradient of PGN can be expressed as a linear combination of the SAM gradient and the original loss gradient. This insight enables the extension of the optimization of adversarial example regions in SAM to PGN.
- We address the issues of the existing flat maxima attack approaches (SAM and PGN) by considering the information geometry of the input space. Specifically, we redefine the neighborhood structure of adversarial examples from the Euclidean space to the manifold defined by the Fisher metric. To the best of our knowledge, this aspect has not been previously studied.
- An approximation of the Fisher information metric is proposed under the model ensemble setting, resulting in the same computational complexity of the proposd attack method as SAM and PGN, $O(2n)$, where n represents the image size. This approximation achieves consistent performance as full Fisher matrix without incurring additional computational cost.

Extensive experiments across ImageNet-compatible [35], CIFAR-10, and CIFAR-100 datasets demonstrate that NG-PGN significantly outperforms state-of-the-art attacks on diverse target models, including convolutional neural networks, vision transformers, adversarially trained models (e.g., Inc-v3$_{adv}$ [36]), and defense mechanisms. In particular, NG-PGN achieves the highest attack success rates of 74.8% on Swin-Base [37] and 87.3% on ResNet-152 [38], surpassing baseline methods by up to 30%. Furthermore, we quantitatively demonstrate that our manifold extension based on the natural gradient yields flatter loss regions, as evidenced by a 25% reduction in the Hessian trace compared to standard PGN, indicating that our method identifies a smoother and flatter loss region.

The rest of the paper is organized and detailed below. Section 2 details the related research work in transfer-based attacks and fisher information. Section 3 describes the intrinsic relationships between SAM and PGN within the context of adversarial attacks and implementation of manifold extension. Section 4 gives a detailed overview of the experimental results and analysis. In the last section, we conclude the paper and discuss future work.

2. Related Work

2.1. Transfer-Based Attack

(1) Gradient optimization attacks. These methods are predominantly based on utilizing the gradient direction of a surrogate model to optimize objective functions. The most typical

technique is the Fast Gradient Sign Method [2]. The current work improves adversarial transferability by preventing adversarial examples from converging to poor local optima. Dong et al. [4] propose a momentum iterative fast gradient sign method (MI-FGSM) to improve the vanilla iterative FGSM methods by integrating the momentum term in the input gradient. Lin et al. [5] propose NI-FGSM (Nesterov Iterative Fast Gradient Sign Method), which leverages the looking-ahead property of Nesterov accelerated gradient to help escape from poor local optima. Wang et al. [6] present Variance tuning iterative method (VMI-FGSM), which follows the momentum iterative concept and incorporates gradient variance computed in the neighborhood of the previous data point to adjust the gradient direction at each iteration. Wang et al. [7] propose Enhanced Momentum Iterative method (EMI-FGSM), which accumulates the gradient of several data points in the direction of the previous gradient to enhance the current gradient. While these methods guide adversarial examples toward more favorable local maxima, they do not provide a precise or formal definition of what constitutes an optimal local maxima in the context of adversarial attacks.

(2) Input transformation attacks. In addition to modifying the gradient, input transformation methods enhance the transferability of adversarial examples by applying image transformation techniques to the input image. Xie et al. [14] propose the Diverse Input Method (DIM), which is the first work to suggest applying differentiable transformations to the clean input. Dong et al. [15] present the Translation Invariance Method (TIM), which aims to generate adversarial perturbations that remain effective not only for the original image but also for its pixel-shifted versions. In a similar motivation to TIM, Liu et al. [39] propose the Scale Invariant Method (SIM), which generates adversarial perturbations that remain effective across various downscaled versions of the input. Wang et al. [16] introduce the Admix method, which mixes up the input with images from other categories while maintaining its original labels. Long et al. [40] propose the Spectrum simulation attack (SSA), which transforms the input image into the frequency domain and performs the attack in that space. Wang et al. [41] present Block Shuffle & Rotation method (BSR), which divides he input image into multiple blocks, then randomly shuffles and rotates these blocks to generate diverse transformed images for gradient computation. These methods can be combined with gradient optimization attacks to further improve the transferability. However, these methods rely on the gradient accumulation of multiple transformations, which significantly increases the computational cost.

(3) Model ensemble attacks. These approaches are based on the hypothesis that if an example remains adversarial for multiple models, then it is more likely to transfer to other models. The basic idea of ensemble attacks is to utilize the output of multiple models to obtain an average model loss [42] or logits [4] and then apply gradient optimization attacks. Xiong et al. [20] propose the Stochastic Variance Reduced Ensemble (SVRE), which aims to reduce the gradient variance across multiple models through an additional iterative procedure computing an unbiased estimate of the ensemble gradient. In practice, the number of surrogate models is often limited, necessitating multiple sampling from the model ensemble to approach the generalization upper bound, which leads to substantial computational costs. Furthermore, only focusing on the variance does not necessarily lead to better generalization [43]. Chen et al. [21] focus on capturing the intrinsic characteristics of transferability and propose the Adaptive Model Ensemble Attack (AdaEA). This method adaptively regulates the fusion of each model's output by monitoring the discrepancy ratio of their contributions to the adversarial objective, thereby enhancing attack effectiveness through dynamic ensemble weighting. Li et al. [44] advocate attacking a Bayesian surrogate model for achieving desirable transferability. However, such self-ensembled surrogate models typically exhibit lower performance compared to standard surrogate models, re-

sulting in limited performance. Tang et al. [45] propose SMER, which introduces ensemble reweighing to refine ensemble weights by maximizing attack loss based on reinforcement learning. In contrast to the gradient optimization attacks and input transformation attacks, the crucial characteristics of ensemble attacks have not been fully exploited.

2.2. Flat Loss Surface and Transferability

Recently, learning algorithms motivated by the sharpness of the loss surface as an effective measure of the generalization gap have demonstrated state-of-the-art performance [46–48]. In particular, Jiang et al. [49] conducted a comprehensive study of 40 complexity measures and found that sharpness-based measures exhibit the strongest correlation with generalization performance. The SAM method [24] is an effective algorithm for obtaining a flatter loss landscape. It is formulated as a min-max optimization problem, where the objective is to simultaneously minimize both the loss value and the sharpness of the loss landscape. The PGN method [27] demonstrated that adding a gradient norm of the loss function can help the optimizer find flat local minima. Similarly, empirical and theoretical studies [8,9,22,30] suggest that transferability is also closely correlated with the flatness of the loss landscape at the adversarial examples. Qin et al. [9] propose improving the transferability of adversarial examples within the SAM-like min-max bi-level optimization framework. They seek to find perturbations that can maximize the loss of the perturbed data point updated by adding a reverse perturbation in the inner-minimization process. Chen et al. [22] propose the Common Weakness Attack (CWA) method, which introduces a modified SAM algorithm that optimizes landscape flatness within the infinity-norm space, enhancing transferability in ensemble attacks. Ge et al. [8] extend the PGN strategy to the adversarial attacks by penalizing the gradient norm of adversarial examples, thereby guiding the adversarial examples toward flat maxima.

These attacks validate the observation that adversarial examples located in flat local regions of the loss surface tend to exhibit improved transferability. However, we find that these methods rely on the gradient of the worst-case loss within a small neighborhood around the adversarial point. A significant drawback of these approaches is their assumption that the neighborhood around adversarial examples is denoted as a Euclidean space; the worst-case is found in the direction of the gradient with respect to the input. Since the model outputs are defined over a discrete probability distribution (e.g., class probabilities), the input can be interpreted as a pullback mapping from the output space, and the input space is rendered as a manifold with a non-linear Riemannian metric. This neglect of the information geometry of the input space leads to suboptimal results.

3. Methodology

In this section, we first introduce preliminaries and then review the objectives of SAM and PGN in adversarial settings. We demonstrate that PGN's gradient is a linear combination of SAM's gradient and the original loss gradient. Building on this, we redefine the neighborhood structure of adversarial examples from the Euclidean space to the manifold and propose NG-PGN, which addresses the issues of the existing flat maxima attack approaches.

3.1. Preliminaries

We let $\mathcal{F}_\Theta := \{f_\theta\}$ denote a set of deep models, where each model f_θ with parameter θ maps input variables X to label variables Y and Θ denotes a discrete ensemble of model parameters. Given an input image $x \in X$ with its corresponding true label $y \in Y$, let $\mathcal{B}_\epsilon(x) = \{\hat{x} : \|\hat{x} - x\|_p \leq \epsilon\}$ be an ϵ-ball around x, where $\epsilon > 0$ is a predefined perturbation magnitude, and $\|\cdot\|_p$ denotes the L_p-norm (we study the L_∞-norm). By simultaneously

attacking multiple surrogate models, ensemble attacks reduce the upper bound on the generalization error in empirical risk minimization and generate adversarial examples with higher transferability [23]. The attack objective is to maximize the loss over the average of the logits ensemble, which can be formulated as the following constrained optimization problem:

$$\max_{x' \in \mathcal{B}_\epsilon(x)} \mathcal{L}(y, x'; \Theta) = \max_{x' \in \mathcal{B}_\epsilon(x)} \mathcal{L}_{ce}\left(\mathbb{E}_{f_\theta \in \mathcal{F}_\Theta}\left[f_\theta(x')\right], y\right) \tag{1}$$

where $\mathcal{L}(\cdot)$ represents the standard ensemble loss, $f_\theta(x')$ represents the logits output of model f_θ for the adversarial example x', and $\mathcal{L}_{ce}(\cdot)$ is the loss function (e.g., cross-entropy loss).

Previous works [4,20–22] achieve the attack objective by using advanced gradient optimization attacks. For example, the momentum iterative update [4] of adversarial examples is expressed as follows:

$$x'_{t+1} = \prod_{\mathcal{B}_\epsilon(x)} \left[x'_t + \alpha \cdot \text{sign}(\mathcal{M}_{t+1})\right], \mathcal{M}_{t+1} = \mu \mathcal{M}_t + \frac{\nabla_{x'_t} \mathcal{L}(y, x'_t; \Theta)}{\|\nabla_{x'_t} \mathcal{L}(y, x'_t; \Theta)\|_1} \tag{2}$$

where μ is the decay factor, $\mathcal{M}$ represents the accumulation of gradients, t is the current number of iteration, α is the step size, and $\prod_{\mathcal{B}_\epsilon(x)}$ constraints x' in the ϵ-ball around x.

3.2. Finding Flat Maxima Strategies

Recent empirical and theoretical studies [8,9,22,30,50,51] suggest that transferability is closely correlated with the flatness of the loss landscape at adversarial examples. In general, adversarial examples at flat maxima tend to exhibit better transferability. The SAM [22,24] and PGN methods [8,27] are two efficient approaches to achieve a flatter landscape. We begin by analyzing the attack objectives of standard SAM and PGN as follows:

SAM [24] modifies the loss function during each iteration of model training, where the new loss at the current iteration computes an approximate worst-case loss within its neighborhood. The optimization process of adversarial examples is analogous to the model training process [5,8,20]. Therefore, we can directly apply the SAM strategy [22] in ensemble attacks to improve transferability. The attack objective is modified to maximize the following worst-case objective instead of the standard ensemble loss. More formally, considering an r-ball neighborhood, the attack objective of SAM is defined as:

$$\max_{x' \in \mathcal{B}_\epsilon(x)} \mathcal{L}_{SAM}(y, x'; \Theta) = \max_{x' \in \mathcal{B}_\epsilon(x)} \min_{\|\gamma\|_2^2 \leq r} \mathcal{L}(y, x' - \gamma; \Theta) \tag{3}$$

where γ is the perturbation vector on adversarial examples. γ can be approximately solved using the closed-form of first-order Taylor expansion of the inner minimization. In the Euclidean space, the solution of γ becomes the gradient norm of the standard loss function,

$$\gamma = r \cdot \nabla_{x'_t} \mathcal{L}(y, x'_t; \Theta) \|\nabla_{x'_t} \mathcal{L}(y, x'_t; \Theta)\|_2^{-1}. \tag{4}$$

The objective of SAM can also be accomplished iteratively using gradient optimization methods (e.g., MI-FGSM). Computationally, SAM incurs twice the cost of gradient computation at each iteration: one for computing γ and the other for updating the adversarial examples by accumulating the gradient at $x' - \gamma$.

PGN [27] promotes the loss function to have a small Lipschitz constant locally, thereby flattening the loss landscape. Intuitively, the objective of PGN can be formulated as introducing a penalty term on the gradient norm into the standard loss function $\mathcal{L}$, as follows.

$$\max_{x' \in \mathcal{B}_\epsilon(x)} \mathcal{L}_{PGN}(y, x'; \Theta) = \mathcal{L}(y, x'; \Theta) - \|\nabla_{x'} \mathcal{L}(y, x'; \Theta)\|_2 \tag{5}$$

where λ is the penalty coefficient, s.t. $\lambda \geq 0$. However, the gradient of $\mathcal{L}_{PGN}$ involves the Hessian matrix, which is infeasible to compute directly. The finite difference method is used to approximate the second-order Hessian matrix using the first-order gradient. Through analysis of the approximate solution of Equation (5), we derive the following corollary:

Corollary 1. *The gradient of $\mathcal{L}_{PGN}$ can be expressed as a linear combination of the gradient of $\mathcal{L}_{SAM}$ with the gradient of the standard ensemble loss $\mathcal{L}$:*

$$\nabla\mathcal{L}_{PGN}(y, x'; \Theta) \approx (1-\delta)\nabla\mathcal{L}(y, x'; \Theta) + \delta\nabla\mathcal{L}_{SAM}(y, x'; \Theta) \tag{6}$$

where $0 \leq x' \leq 1$ is a balancing coefficient.

Proof. The gradient of Equation (6) can be expressed as follows:

$$\begin{aligned}\nabla\mathcal{L}_{PGN}(y, x'; \Theta) &\approx \nabla\mathcal{L}(y, x'; \Theta) - \lambda\nabla\left\|\nabla\mathcal{L}(y, x'; \Theta)\right\|_2 \\ &\approx \nabla\mathcal{L}(y, x'; \Theta) - \lambda\nabla^2\mathcal{L}(y, x'; \Theta)\frac{\nabla\mathcal{L}(y, x'; \Theta)}{\|\nabla\mathcal{L}(y, x'; \Theta)\|_2}\end{aligned} \tag{7}$$

Using the Finite Difference Method to approximate the Hessian matrix $\nabla^2\mathcal{L}(x', y; \Theta)$, we have

$$-\nabla^2\mathcal{L}(y, x'; \Theta)\frac{\nabla\mathcal{L}(y, x'; \Theta)}{\|\nabla\mathcal{L}(y, x'; \Theta)\|_2} \approx \frac{\nabla\mathcal{L}(y, x'-\gamma; \Theta) - \nabla\mathcal{L}(y, x'; \Theta)}{r} \tag{8}$$

Based on Equation (3), we obtain

$$-\nabla^2\mathcal{L}(y, x'; \Theta)\frac{\nabla\mathcal{L}(y, x'; \Theta)}{\|\nabla\mathcal{L}(y, x'; \Theta)\|_2} \approx \frac{\nabla\mathcal{L}_{SAM}(y, x'; \Theta) - \nabla\mathcal{L}(y, x'; \Theta)}{r} \tag{9}$$

where $\gamma = r \cdot \frac{\nabla\mathcal{L}(y,x';\Theta)}{\|\nabla\mathcal{L}(y,x';\Theta)\|_2}$. Combining Equations (7) and (8),

$$\nabla\mathcal{L}_{PGN}(y, x'; \Theta) \approx \left(1 - \frac{\lambda}{r}\right)\nabla\mathcal{L}(y, x'; \Theta) + \frac{\lambda}{r}\nabla\mathcal{L}_{SAM}(y, x'; \Theta) \tag{10}$$

Setting $\delta = \frac{\lambda}{r}$ as the balancing coefficient, we thereby complete the Corollary 1's proof. Figure 1 illustrates the iterative update process of the three aforementioned attack objectives. □

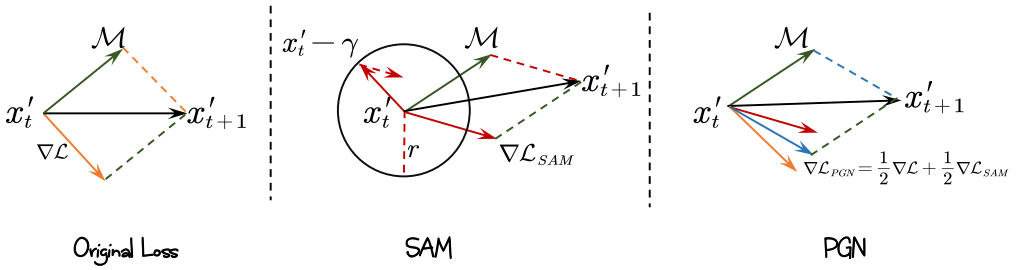


Figure 1. The process illustrates the momentum iterative optimization for achieving the three attack objectives: original loss, SAM, and PGN. The symbols are introduced in Equations (3)–(5).

3.3. Manifold Extensions Using Natural Gradient

The above theoretical analysis shows that existing flat maxima strategies use the gradient of the standard loss function to find the worst-case loss within neighborhoods around adversarial examples, which neglect the intrinsic geometry of the input space. Deep models inherently map inputs to discrete probability distributions, rendering the input space a statistical manifold endowed with nonlinear Riemannian geometry [31–34].

The Euclidean metric fails to capture the local curvature characteristics of such manifolds, causing gradient directions to deviate from the true steepest descent paths and limiting adversarial examples' generalization.

To address this problem, we redefine the neighborhood of input as a manifold equipped with a non-linear Riemannian metric. In this case, the natural gradient gives the steepest-changing direction. We can compute the Kullback–Leibler (KL) divergence between the model probability outputs $p(y|x;\theta)$ and $p(y|x';\theta)$ to quantify the distance between x and any x'. For notational distinction, we redefine γ under the non-Euclidean metric as γ_m. Therefore, the objective of the inner minimization in Equation (3) is optimized as:

$$\begin{aligned} &\min \mathcal{L}\left(y, x' - \gamma_m; \Theta\right), \\ &\text{s.t.} \mathrm{KL}\left(p\left(y|x' - \gamma_m; \Theta\right)||p\left(y|x'; \Theta\right)\right) \leq r \end{aligned} \tag{11}$$

where $p(y|\cdot;\Theta)$ represent $\mathbb{E}_{f_\theta \in \mathcal{F}_\Theta}[p(y|\cdot;\theta)]$ in the model ensemble setting.

Fisher Metric. Since the first-order gradient of the KL divergence with respect to γ vanishes, the second derivative corresponds to the Fisher information matrix, leading to:

$$\mathrm{KL}\left(p\left(y|x' - \gamma_m; \Theta\right)||p\left(y|x'; \Theta\right)\right) \approx \frac{1}{2}\gamma_m^T F\left(x'\right)\gamma_m \tag{12}$$

where $F(x')$ is the Fisher Information Matrix associated with x' which contains all the information about the curvature in the standard loss function. Therefore, Equation (11) is modified to:

$$\begin{aligned} &\min \mathcal{L}\left(y, x' - \gamma_m; \Theta\right), \\ &\text{s.t.} \frac{1}{2}\gamma_m^T F\left(x'\right)\gamma_m \leq r \end{aligned} \tag{13}$$

Next, this objective is transformed into a unconstrained optimization problem by the Lagrange multiplier method:

$$\begin{aligned} \gamma_m &= \arg\min_{\gamma_m} \mathcal{L}\left(y, x' - \gamma_m; \Theta\right) + \lambda_m \left(\frac{1}{2}\gamma_m^T F\left(x'\right)\gamma_m - r\right) \\ &\approx \arg\min_{\gamma_m} \mathcal{L}\left(y, x'; \Theta\right) + \nabla_{x'}\mathcal{L}\left(y, x'; \Theta\right)^T \gamma_m + \frac{\lambda_m}{2}\gamma_m^T F\left(x'\right)\gamma_m - \lambda_m r \end{aligned} \tag{14}$$

where λ_m is the Lagrangian coefficient. We set the partial derivative concerning γ_m to 0 and have:

$$\begin{aligned} &\nabla_{x'}\mathcal{L}\left(y, x'; \Theta\right)^T + \frac{\lambda_m}{2} F\left(x'\right)\gamma_m = 0 \\ \Rightarrow &\gamma_m = -\frac{2}{\lambda_m} F\left(x'\right)^{-1} \nabla_{x'}\mathcal{L}\left(y, x'; \Theta\right) \end{aligned} \tag{15}$$

where $F(x')^{-1}\nabla_{x'}\mathcal{L}(y, x';\Theta)$ represents the Natural Gradient, adjusting the gradient $\nabla_{x'}\mathcal{L}(y, x';\Theta)$ according to the manifold curvature. To meet the neighborhood size r (As $r \to 0$, KL divergence is approximately symmetric) while maintaining suitability for adversarial attacks, γ_m is defined as the norm of the natural gradient:

$$\gamma_m = r \cdot \frac{F\left(x'\right)^{-1}\nabla_{x'}\mathcal{L}\left(y, x'; \Theta\right)}{\|F\left(x'\right)^{-1}\nabla_{x'}\mathcal{L}\left(y, x'; \Theta\right)\|_2^{-1}} \tag{16}$$

3.4. Approximating Fisher Information Matrix

Following the above analysis, the key to calculating the natural gradient is obtaining the Fisher Information Matrix of the adversarial examples. To review, for model parameters θ, the Fisher information matrix is defined by:

$$F(\theta) = \mathbb{E}_x \mathbb{E}_y \left[\nabla_\theta \log p(y, x; \theta) \nabla_\theta \log p(y, x; \theta)^T \right] \tag{17}$$

In the concept of adversarial attacks, adversarial examples can be viewed as parameters to be optimized and the surrogate models as inputs. Therefore, in the model ensemble setting, the Fisher information matrix $F(x')$ can be approximated by:

$$F(x') = \mathbb{E}_\theta \mathbb{E}_y \left[\nabla_{x'} \log p(y, x'; \theta) \nabla_{x'} \log p(y, x; \theta)^T \right] \tag{18}$$

However, estimating such a Fisher information matrix by averaging the gradients multiple surrogate models for all classes is computationally unacceptable, especially when dealing with large-scale images (*i.e.*, $F(x') \in \mathbb{R}^{|x'| \times |x'|}$ with $|x'|$ denoting the size of input image). To save the computational effort, following previous works [31,32], we adopt a diagonal approximation, *i.e.*, $F(x') \in \mathbb{R}^{|x'|}$:

$$F(x') = \mathbb{E}_\theta \mathbb{E}_y \left[\nabla_{x'} \log p(y, x'; \theta)^2 \right] \tag{19}$$

Furthermore, since the objective of adversarial attacks is similar to that in supervised learning, the expectation $\mathbb{E}_y$ over label variables can be achieved by using only the true label. Consequently, we introduce the Empirical Fisher method in the model ensemble setting:

$$F(x') = \mathbb{E}_\theta \left[\nabla_{x'} \log p(y, x'; \theta)^2 \right] \tag{20}$$

Taking the element-wise reciprocal, we can obtain the inverse of $F(x')$. Figure 2 illustrates the calculation process of the Empirical Fisher. The details of calculating the natural gradient are summarized in Algorithm 1. Our empirical results (Section 4.6) demonstrate that employing a diagonal approximation yields performance comparable to the standard full-matrix version while significantly reducing computational overhead.

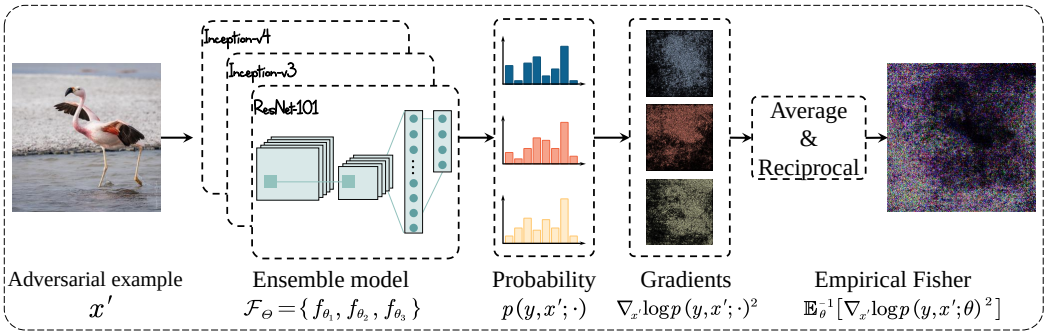


Figure 2. The process illustrates the calculation of the Fisher Information Matrix. To save computational effort, we adopt a diagonal approximation and introduce the Empirical Fisher method in the model ensemble setting.

In the model ensemble setting, we finally approximate the Fisher information matrix and use the direction of natural gradient γ_m to find the worst-case point for adversarial examples. Therefore, the manifold extension of SAM, called NG-SAM, can be expressed as follows,

$$\begin{aligned} &\mathcal{L}_{NG-SAM}(y, x'; \Theta) \approx \mathcal{L}(y, x' - \gamma_m; \Theta) \\ &\gamma_m = r \cdot \frac{\nabla_{x'} \mathcal{L}(y, x'; \Theta) / \mathbb{E}_\theta \left[\nabla_{x'} \log p(y, x'; \theta)^2 \right]}{\left\| \nabla_{x'} \mathcal{L}(y, x'; \Theta) / \mathbb{E}_\theta \left[\nabla_{x'} \log p(y, x'; \theta)^2 \right] \right\|_2^{-1}} \end{aligned} \tag{21}$$

Furthermore, by specifying the balancing coefficients $0 < \delta < 1$, we propose the manifold extension of PGN, termed NG-PGN. The calculation gradient of NG-PGN can be expressed as follows,

$$\nabla \mathcal{L}_{NG-PGN}(y, x'; \Theta) \approx (1-\delta)\nabla \mathcal{L}(y, x'; \Theta) + \delta \mathcal{L}_{NG-SAM}(y, x'; \Theta) \tag{22}$$

where the gradient is computed analogously to Equation (6) with $x' = x' - \gamma_m$. Notably, our approach considers only the local neighborhood around the adversarial example as the manifold, rather than the entire image space. Additionally, We randomly sample multiple examples and average the gradients of these examples to obtain a more stable gradient. We empirically show in Section 4.8 that NG-SAM performs worse than NG-PGN, and we provide a detailed analysis of this observation in Section 4.10. Therefore, NG-PGN is adopted as the default method in subsequent comparisons with state-of-the-art approaches. The detailed procedure of the NG-PGN attack is presented in Algorithm 2.

Algorithm 1: Calculating Natural Gradient

Input: Surrogate networks $\mathcal{F}_\Theta$ with parameters ensemble Θ, loss function $\mathcal{L}$; adversarial example x';

Output: The direction under the Fisher metric γ_m.

Calculate the gradient of standard ensemble loss using Equation (1);

Estimate Fisher Information Matrix using Equation (20);

Calculate the norm of natural gradient using Equation (16);

return: γ_m.

Algorithm 2: NG-PGN attack method (based on MI-FGSM)

Input: Surrogate networks $\mathcal{F}_\Theta$ with parameters ensemble Θ, loss function $\mathcal{L}$; a natural example x with ground-truth label y;

Output: The adversarial example x';

Parameters: The magnitude of perturbation ϵ; the number of iteration T; the decay factor μ; the balancing coefficient δ; the neighborhood size r;

Initialize: $\alpha = \epsilon/T$; $\mathcal{M}_0 = 0$; $x'_0 = x$;

for $t = 1$ *to* T **do**

  Calculate $g = \nabla_{x'_t} \mathcal{L}(y, x'_t; \Theta)$;

  Calculate γ_m using Algorithm 1;

  Calculate $g' = \nabla_{x'_t} \mathcal{L}(y, x'_t - \gamma^m; \Theta)$;

  Update $g = x' \cdot g + (1 - x') \cdot g'$;

  Update $x'_{t+1} = \prod_{\mathcal{B}_\epsilon(x)} [x'_t + \alpha \cdot \text{sign}(\mathcal{M}_{t+1})]$, $\mathcal{M}_{t+1} = \mu \mathcal{M}_t + \frac{g}{\|g\|_1}$;

end

return: x';

4. Experiment

4.1. Experimental Settings

(1) Dataset: Following the protocol established in previous studies [6,7,9,27], we conduct our experiments on 3 widely used datasets, ImageNet-compatible [35], CIFAR-10 and CIFAR-100. For the ImageNet-compatible dataset, it is comprised of 1000 images with the size of $299 \times 299 \times 3$. For both CIFAR-10 and CIFAR-100, we select 1000 images along with their corresponding true labels from the test set. The image size is $32 \times 32 \times 3$.

(2) Target models: To validate the effectiveness of our methods, we test attack performance in comprehensive models. For the black-box attacks task on the ImageNet-compatible dataset, we select five normally trained models from both Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs): ResNet-152 (Res-152) [38], DenseNet-121 (Dense-121) [52], ViT-Base (ViT-B) [53], Swin-Base (Swin-B) [37], and Deit-Base (Deit-B) [54], all are pretrained and available in Timm [55]. We also consider adversarially trained models, including Inc-v3$_{adv}$, Inc-v3$_{ens3}$, Inc-v3$_{ens4}$, and Inc-Res-v2$_{ens}$ [36]. Additionally, we evaluate various defense methods, such as HGD [56], RS [57], NRP [58], and Diffpure [59], which have demonstrated robustness against black-box attacks. For adversarial attacks on the CIFAR-10 and CIFAR-100, we select normally trained models, including Inc-v3 [60], MobileNet (Mobile) [61], and Densenet [52] as black-box target models.

(3) Baselines: Since the Empirical Fisher approximation is implemented in the model ensemble setting, we compare the proposed NG-PGN with state-of-the-art gradient optimization attacks under the model ensemble setting, including MI [4], NI [5], PI [62], VMI [6], VNI [6], EMI [7], RAP [9], and recent model ensemble attacks like SVRE [20], AdaEA [21], CWA [22] and SMER [45]. Additionally, we conduct ablation experiments comparing our method to the original PGN and SAM (both implemented under the Euclidean metric) to verify the effectiveness of our manifold.

(4) Hyper-parameters: Following the conventional settings in previous works [4,6,8–11], we set the maximum perturbation ϵ to 16/255, the number of iterations $T = 10$, the step size $\alpha = \epsilon/T$, and the decay factor for MI-FGSM to 1.0. The neighborhood size r is set to $3 \cdot \alpha \approx 0.019$. For the NG-PGN attack, we set the balancing coefficient $\delta = 1/2$. For the compared methods, we use the optimal hyper-parameters as reported in their respective papers. All the experiments were implemented using Pytorch 2.1.2 on an Intel(R) Core(TM) i9-14900K and a NVIDIA GeForce RTX 4090 GPU with 24 GB graph memory.

(5) Metrics: We use the black-box attack success rates (ASR) as the evaluation metric, which is calculated as the percentage of attacks that successfully cause a model to misclassify or generate incorrect outputs. This metric is pivotal in evaluating the robustness of models to adversarial perturbations.

4.2. Attack Results

We conduct a series of black-box attacks on target models and report the attack success rates (ASR). For the ImageNet-compatible dataset, we first consider surrogate models with a similar architecture, including four normally trained networks: ResNet-101 (Res-101) [38], Inception-v3 (Inc-v3), Inception-v4 (Inc-v4) [60], and Inception-ResNet-v2 (IncRes-v2) [63]. The results are presented in Table 1. Notably, for DiffPure, we set the maximum perturbation of all attack methods as $\epsilon = 4/255$ to align with its default parameter settings. Gradient optimization methods (e.g., MI, PI, VMI, EMI) typically achieve attack success rates ranging from 70% to 90% on normally trained CNNs (e.g., ResNet-152, DenseNet-121) and 30–50% on ViTs (e.g., ViT-B, Swin-B, DeiT-B). In general, methods that more effectively optimize gradient information tend to yield higher transferability, thereby achieving greater attack success rates in black-box settings. AdaEA achieves higher 61.5% ASR on Swin-B, demonstrating that employing a gradient adaptation strategy within the ensemble attack framework can indeed enhance performance. Across all target models, the NG-PGN attack outperforms other methods; for instance, MI, SVRE and AdaEA achieve 33.9%, 61.5% and 51.2% ASRs respectively on Swin-B, while our NG-PGN method achieves an impressive success rate of 74.8%. Moreover, when targeting adversarially trained models, the NG-PGN attack method consistently surpasses other model ensemble attacks, achieving

an overall improvement of nearly 30% in attack success rates compared to the state-of-the-art methods. These results confirm that our proposed method effectively enhances the transferability of adversarial examples across various target models. We further evaluate black-box attacks against defense methods, with results presented in Table 1 (right). We observe that NG-PGN continues to outperform other model ensemble attacks. These findings further validate the efficacy of incorporating the information geometry of the data space to significantly enhance the transferability of adversarial attacks.

Table 1. The untargeted black-box attack success rates (%) of various attack methods under the model ensemble setting. The surrogate models include Res-101, Inc-v3, Inc-v4, and IncRes-v2. The target models include normally and adversarially trained models and defense methods. The dataset is ImageNet-compatible. Here, the best results are bold.

Attack	Normally Trained Models					Adversarially Trained Models				Defense Methods			
	Res	Dense	ViT-B	Swin-B	Deit-B	Inc-v3$_{adv}$	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	Inc-v2$_{ens}$	HGD	RS	NRP	Diffpure
MI	69.5	75.6	27.1	31.7	31.7	35.5	31.9	31.7	20.4	28.4	31.3	37.4	21.6
NI	76.6	82.9	28.4	31.0	31.9	37.7	31.5	29.1	20.8	24.6	30.6	39.2	26.5
PI	74.9	82.4	44.8	32.0	37.9	48.6	39.5	36.4	24.6	26.2	34.5	40.7	14.1
VMI	81.5	86.9	49.3	53.6	53.3	61.3	65.1	61.3	53.7	55.3	37.1	46.5	31.7
VNI	90.5	93.4	45.9	54.5	52.7	62.4	62.3	58.9	52.7	60.8	39.4	47.9	37.2
EMI	**93.9**	**96.2**	49.5	56.6	54.0	62.5	58.3	52.4	40.5	65.2	40.1	48.8	42.9
RAP	91.2	94.6	45.4	53.1	51.7	51.2	35.0	31.3	19.1	70.4	53.4	50.2	41.2
PGN	89.3	93.0	75.5	72.5	76.1	85.5	86.5	85.4	80.7	80.2	**64.2**	54.9	44.6
SVRE	79.8	85.2	28.5	33.9	30.6	51.1	47.7	55.6	46.3	47.6	40.2	30.5	22.5
AdaEA	78.2	80.6	53.6	61.5	60.3	48.3	55.2	56.2	40.3	45.8	41.6	32.6	21.4
CWA	80.5	80.2	54.7	51.2	52.9	55.0	63.5	58.8	51.7	49.8	46.2	35.9	34.9
SMER	83.2	85.0	84.0	64.7	65.1	60.9	70.2	62.0	44.5	56.3	47.5	42.1	38.7
NG-PGN	90.6	93.6	**76.5**	**74.8**	**76.2**	**87.6**	**88.0**	**86.9**	**81.8**	**85.4**	47.1	**56.2**	**48.3**

We then compare the attack success rates of adversarial examples generated by attacking a heterogeneous ensemble, including Res-101, Inc-v3, and ViT-B, with image size set to 224×224. The results are presented in Table 2. EMI and VMI also perform well, with average rates above 70% for normally trained models and 65% for adversarially trained models. We observe that NG-PGN consistently achieves the highest success rates across all models, significantly outperforming other attacks. For example, NG-PGN achieves the highest 87.3% on Res-152 and 89.7% on Inc-v3$_{adv}$. Its average success rates reach 88.2% for normal trained models and 87.0% for adversarially trained models, which are notably higher than other methods. These results also demonstrate the applicability of our Fisher Information Matrix approximation method to various model structures.

To further validate the effectiveness of our NG-PGN method across different datasets, we conduct experiments on the CIFAR-10 and CIFAR-100 datasets. Following conventional settings from prior work [4,6,8,21], we set the maximum perturbation ϵ to $8/255$, the number of iterations to $T = 10$, and the step size α to $2/255$. The adversarial examples are generated by attacking a model ensemble that includes VGG-16 (VGG) [64], ResNet-18 (Res-18), and ResNet-50 (Res-50) [38]. Results in Table 3 clearly show that our method improves attack transferability on the CIFAR-10 dataset. These results demonstrate broad applicability of our NG-PGN across various datasets and supports the notion that adversarial examples situated in flat local regions tend to exhibit improved transferability across diverse models.

Table 2. The untargeted black-box attack success rates (%) of various attack methods under the model ensemble setting. The surrogate models include Res-101, Inc-v3, and ViT-B. The target models include normally and adversarially trained models. The dataset is ImageNet-compatible. To accommodate the requirements of the ViT model, we resize the input images to 224 × 224 × 3. Here, the best results are bold.

Attack	Normally Trained Models						Adversarially Trained Models				
	Res-152	Dense-121	Inc-v4	Swin-B	Deit-B	Avg.	Inc-v3$_{adv}$	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	Inc-v2$_{ens}$	Avg.
MI	57.3	65.0	60.6	41.2	61.2	57.1	54.3	48.4	48.7	38.7	47.5
NI	64.1	72.9	69.1	41.2	59.9	61.4	62.5	54.9	52.6	41.5	52.9
PI	62.5	76.1	72.8	34.7	58.5	60.9	68.3	64.9	60.5	53.5	61.8
VMI	73.3	78.4	79.1	58.2	75.4	72.9	74.6	71.2	69.6	61.5	69.2
VNI	80.8	85.3	83.9	61.6	77.9	77.9	78.6	75.6	73.4	65.8	73.4
EMI	86.0	88.0	86.8	64.6	82.4	81.6	80.9	74.9	73.3	65.5	73.7
RAP	86.0	91.0	89.2	64.0	82.0	82.4	80.1	69.3	66.4	51.8	66.9
PGN	85.5	90.3	91.1	74.3	89.7	86.1	86.1	87.2	86.5	83.1	85.7
SVRE	62.0	68.7	63.9	40.7	54.4	57.9	63.4	55.0	53.8	42.4	53.6
AdaEA	57.9	63.6	59.2	66.7	65.5	58.6	60.6	58.4	56.3	45.8	59.3
CWA	73.9	78.5	66.8	64.9	61.5	61.1	70.5	62.2	63.8	50.8	65.6
SMER	74.3	79.8	72.8	58.1	74.9	71.9	72.6	72.7	72.3	63.8	70.4
NG-PGN	**87.3**	**91.7**	**92.6**	**76.8**	**92.7**	**88.2**	**89.7**	**88.1**	**87.1**	**83.3**	**87.0**

Table 3. The white-box and black-box attack success rates (%) of various attack methods under the model ensemble setting. The surrogate models include VGG, Res-18, and Res-50. The target models include white-box models and black-box models (Inc-v3, Mobile, and Densenet). The datasets are CIFAR10 and CIFAR100. Here, the best results are bold.

Attack	CIFAR-10							CIFAR-100						
	VGG	Res-18	Res-50	Inc-v3	Mobile	Densenet	Avg.	VGG	Res-18	Res-50	Inc-v3	Mobile	Densenet	Avg.
MI	68.7	80.1	76.0	68.5	70.7	68.2	72.0	77.1	75.0	73.9	53.3	60.7	56.6	66.1
NI	84.2	92.3	88.6	75.8	80.4	77.1	82.9	79.4	78.1	76.6	53.5	63.9	57.4	68.1
PI	83.0	51.9	48.9	28.1	29.1	34.8	38.5	61.8	66.4	65.9	28.4	36.0	35.1	48.9
VMI	75.0	83.9	81.8	74.6	76.7	74.3	77.7	88.6	85.6	84.5	68.3	75.3	69.6	78.6
VNI	86.0	92.7	89.5	81.7	84.8	82.5	86.2	91.6	88.6	87.5	71.3	78.3	72.6	81.6
EMI	92.8	92.3	89.5	90.7	90.2	85.4	90.2	89.7	87.4	86.4	73.9	79.4	74.1	81.8
RAP	93.0	90.2	88.2	89.4	91.0	83.7	89.3	92.1	86.5	86.2	74.0	78.7	71.6	81.5
PGN	93.8	90.4	89.8	90.8	92.3	87.0	90.7	93.7	87.0	84.9	75.8	81.4	72.0	82.5
SVRE	91.8	90.9	88.6	81.2	84.3	81.1	86.3	88.2	86.6	86.6	63.6	75.0	67.6	77.9
AdaEA	86.1	80.5	83.8	72.6	74.2	73.6	78.4	85.9	86.4	84.7	60.8	70.2	67.1	75.9
CWA	88.9	90.5	87.5	80.8	84.2	77.0	84.8	90.0	87.6	89.0	67.2	77.6	69.1	80.1
NG-PGN	**95.1**	**94.5**	**91.0**	**91.8**	**92.7**	**88.4**	**92.3**	**97.5**	**88.2**	**88.7**	**78.9**	**85.6**	**75.2**	**85.7**

4.3. Combined with Input Transformation Attacks

With its high flexibility and generality, our proposed method can be integrated with input transformation methods to further enhance adversarial example transferability. To demonstrate their effectiveness, we combine NG-PGN with DIM [14], Admix [16], SSA [40] and BSR [41]. We craft adversarial examples using a model ensemble comprising Res-101, Inc-v3, Inc-v4, and IncRes-v2, and evaluate their transferability on both normally and adversarially trained models. Figure 3 shows the attack success rates of adversarial examples generated by these input transformation methods, both individually and combined with NG-PGN. Our method demonstrates significant improvements when combined with these approaches. For instance, SSA achieves only a 65.1% average success rate across adversarially trained models; however, when integrated with our NG-PGN method, this rate increases to 87.3%, representing a 22.2% enhancement. Notably, after incorporating these

input transformation-based methods, our NG-PGN achieves significantly better results on all target models compared to those shown in Table 1. This highlights the scalability of our method and its effectiveness in improving the success rates of transfer-based black-box attacks when combined with existing techniques.

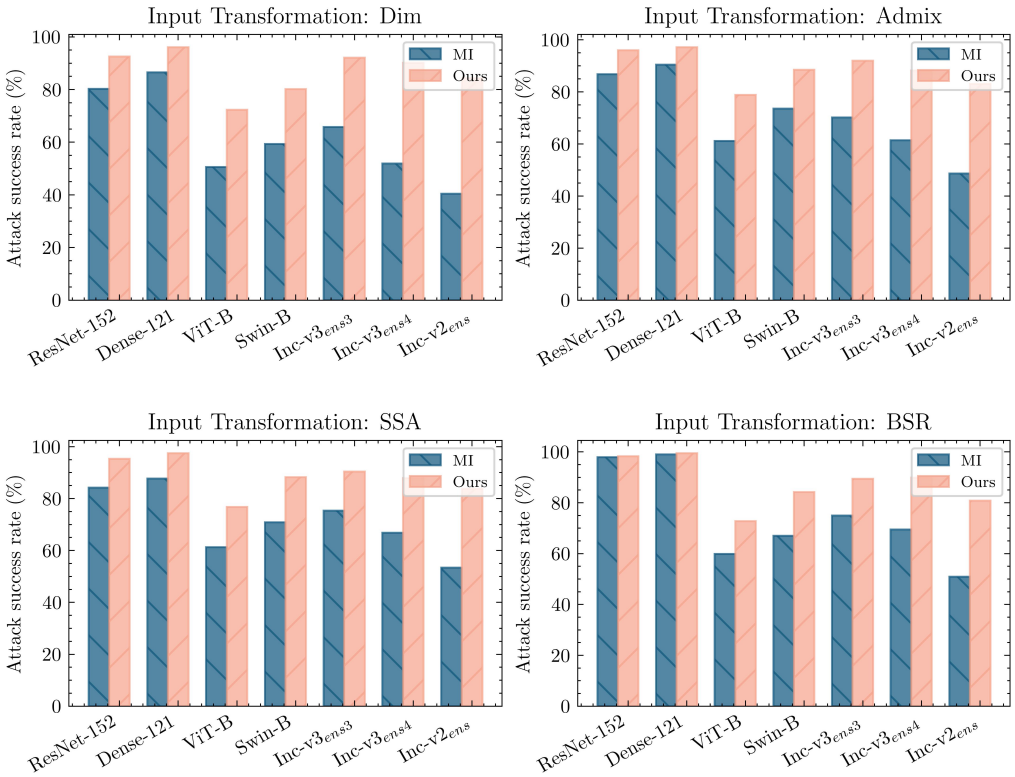


Figure 3. The black-box attack success rates (%) of our NG-PGN method, when it is integrated with DIM, Admix, SSA, and BSR, respectively. We generate adversarial examples using a surrogate model ensemble comprising ResNet-101, Inception-v3, Inception-v4, and Inception-ResNet-v2, and evaluate their transferability on seven black-box models. After combining these input transformation-based methods, our method tends to achieve much better results.

4.4. Visualization of Attack Influence

(1) Loss landscape: To validate that our proposed NG-PGN method helps adversarial examples find flatter maxima regions, we compare the loss surface maps of adversarial examples generated by various attack methods on a surrogate model ensemble (Inc-v3, Res-101, Inc-v4, and IncRes-v2). The adversarial example is positioned at the center of the loss landscape. We randomly select one image from the dataset and visualize loss surfaces of corresponding adversarial examples generated by different attack methods in Figure 4. The comparison reveals that the adversarial examples generated by our approach reside in larger and smoother local maxima regions.

(2) Heatmap: Furthermore, to intuitively illustrate the attack performance, we visualize the attention heatmaps of a clean image and adversarial examples generated by different methods under the black-box ResNet-152 model in Figure 5. As shown in Figure 5a, ResNet-152 focuses on the primary object in the clean image. Figure 5b–k demonstrate that although the attention shifts slightly under adversarial perturbations from other attack methods, it still largely aligns with the focus of the clean image. In contrast, as depicted in Figure 5l, the attention induced by adversarial examples generated by NG-PGN deviates significantly, no longer concentrating on semantically relevant regions.

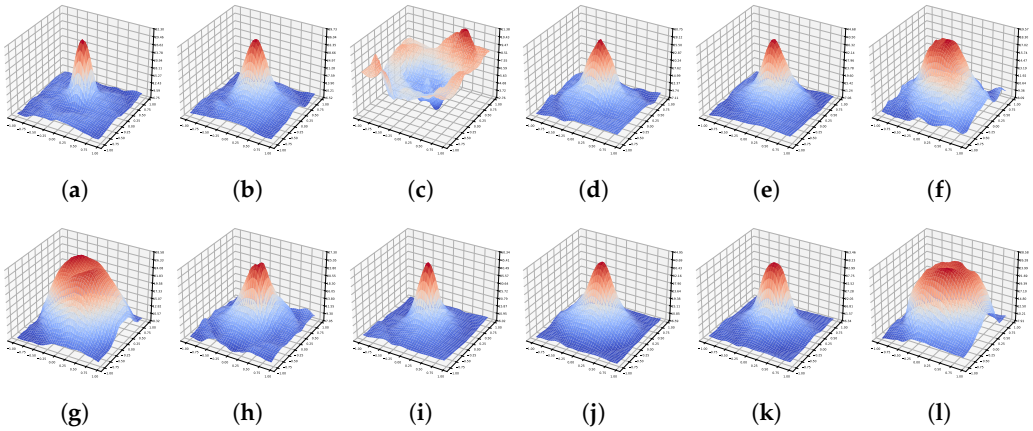


Figure 4. Visualization of loss surfaces along two random directions for one randomly sampled adversarial examples. The center of each 2D graph corresponds to the adversarial example generated by different attack methods in the model ensemble setting. (**a**) MI (**b**) NI (**c**) PI (**d**) VMI (**e**) EMI (**f**) RAP (**g**) PGN (**h**) SVRE (**i**) AdaEa (**j**) CWA (**k**) SMER (**l**) Ours.

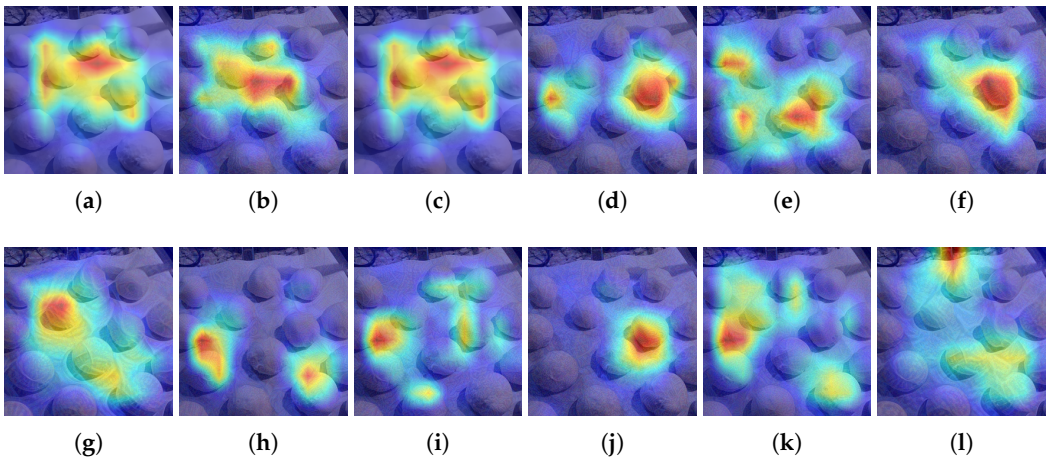


Figure 5. Heatmaps of different inputs in a black-box model. (**a**) clean image. (**b**–**e**) are the corresponding adversarial examples generated by different attack methods in the model ensemble setting. (**a**) Clean (**b**) MI (**c**) PI (**d**) VMI (**e**) EMI (**f**) RAP (**g**) PGN (**h**) SVRE (**i**) AdaEa (**j**) CWA (**k**) SMER (**l**) Ours.

4.5. Quantitative Comparison on Loss Landscape Flatness

We note that flatness is commonly characterized by second-order curvature measures. We quantitatively evaluate the loss landscape flatness of the generated adversarial examples by MI, RAP, PGN, CWA, and our NG-PGN based on the Hessian trace, which is calculated using the method introduced in the work [65]. To ensure a fair comparison, we calculate the average of the Hessian trace of all adversarial examples generated on the ensemble including Res-101, Inc-v3, Inc-v4 and IncRes-v2. A lower Hessian trace directly indicates a flatter loss region. As reported in Table 4, NG-PGN can significantly reduce the Hessian trace compared to the other four methods, which indicates that the generated adversarial examples converge to smoother, flatter regions.

Table 4. The comparison of the Hessian trace of MI, RAP, PGN, CWA and NG-PGN.

Method	MI	RAP	PGN	CWA	NG-PGN
Hessian Trace	16.2	14.5	12.8	11.6	10.3

Table 5. A comparison of Attack success rate, time complexity, and computational cost between Empirical Fisher and Full Fisher Matrix.

Method	Empirical Fisher	Full Fisher Matrix
Attack success rate (%) ↑	92.3	93.5
Time cost (s) ↓	4.45	77.6
Memory usage (GB) ↓	0.3	12.5

4.6. Validation of the Fisher Information Matrix Approximation

To validate the effectiveness of the empirical Fisher information matrix approximation proposed in Section III-E, we conducted comparative experiments on the CIFAR-10 dataset between the full Fisher matrix (Equation (18)) and its diagonalized counterpart (Equation (20)). To comprehensively demonstrate the performance of the Empirical Fisher, we further provide the following three in-depth comparisons against the full Fisher information matrix. Results in Table 5 demonstrate that the diagonal approximation preserves competitive attack capability, achieving an average success rate of 92.3% compared to 93.5% for the full matrix method, with statistically insignificant transferability degradation. Crucially, the computational efficiency improved substantially: For a $32 \times 32 \times 3$ image and a surrogate model ensemble including VGG, Res-18 and Res-50, the full matrix method required 77.6 s per iteration and 12.5 GB memory due to covariance computations across all gradient dimensions, while the diagonal approximation reduced runtime to 4.45 s (94% reduction) and memory consumption to 0.3 GB (97.6% reduction) by retaining only the expectation of squared gradient components. These findings confirm that the empirical approximation maintains near-equivalent attack potency while resolving the prohibitive computational overhead of the full Fisher matrix, enabling practical deployment in resource-constrained scenarios.

4.7. Computational Cost and Efficiency Analysis

To evaluate the practical efficiency of the proposed NG-PGN method, we conduct a comparison of its computational cost against representative gradient-based attacks. We measure both the average time per iteration and the total attack time for different attack methods. The compared methods include MI, VMI, EMI, RAP, PGN, and the proposed NG-PGN. Table 6 reports the computational cost of different methods attacking a 10-image batch ($10 \times 299 \times 299 \times 3$). We can observe that while NG-PGN is more expensive than simple first-order methods such as MI and EMI, this overhead is expected since PGN and NG-PGN both require additional gradient evaluations to optimize flat maxima. Importantly, NG-PGN maintains the same asymptotic computational complexity as PGN, as the proposed diagonal and empirical Fisher approximation avoids explicit matrix construction and inversion.

Table 6. The comparison of the computational cost of different methods.

Method	MI	VMI	EMI	RAP	PGN	NG-PGN
Time/Iter (s)	0.8	2.9	2.1	26.2	11.1	11.3
Total Attack Time (s)	8.9	30.5	20.4	269.4	110.5	112.4

4.8. Ablation Study on Metrics

Standard SAM defines the neighborhood around adversarial examples as a Euclidean ball. We extend this by incorporating input space geometry and adapting it to a manifold. Tables 1–3 show that NG-PGN outperforms PGN, supporting the effectiveness of manifold metrics. To further validate this, we compare SAM under Euclidean and manifold metrics

in Table 7. In black-box attacks on normally trained models, the manifold consistently outperforms the Euclidean version, confirming its advantage in enhancing transferability.

Table 7. The untargeted black-box attack success rates (%) of SAM under the model ensemble setting, using Euclidean and non-Euclidean neighborhoods, respectively.

Attack	Res-152	Dense-121	ViT-B	Swin-B	Avg.
SAM	71.6	78.5	43.7	50.4	61.1
NG-SAM	75.2	83.0	47.8	51.3	64.3

To quantify the impact of manifold extension on loss flatness, we compute the Hessian trace of Inc-v3 at PGN and NG-PGN adversarial examples. The trace, as the sum of eigenvalues, reflects curvature—lower values indicate flatter maxima. As shown in Table 8 (row 1), NG-PGN reduces the trace by 25% over PGN, indicating smoother loss regions. Additionally, our Fisher matrix approximation reduces NG-SAM's complexity to $O(2n)$—matching SAM—with n as image size. We compare NG-PGN and PGN runtime on a single NVIDIA 4090 GPU using a 10-image batch ($299 \times 299 \times 3$), shown in Table 8 (row 3). NG-PGN improves adversarial transferability while maintaining low computational cost.

Table 8. A comparison of the Hessian trace at generated adversarial examples, time complexity, and computational cost between PGN and NG-PGN.

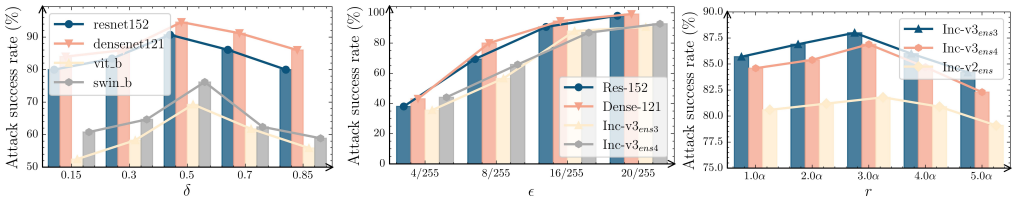
Method	PGN	NG-PGN
Hessian trace ↓	12.8	10.3
Time complexity	$O(n)$	$O(n)$
Computational cost (s) ↓	110.5	112.4

4.9. Ablation Study on Hyper-Parameters

(1) The Balancing Coefficient δ: As $\mathbf{0} < \delta < \mathbf{1}$, the proposed attack is termed NG-PGN. According to the Proof in Section 3.2, when $\boldsymbol{r}$ is fixed, increasing δ is equivalent to increasing the weight $\boldsymbol{\lambda}$ of the gradient norm penalty term. Conversely, for fixed $\boldsymbol{\lambda}$, increasing δ is equivalent to reducing the size $\boldsymbol{r}$ of the neighborhood. As shown in the following Figure 6a, the attack success rates achieve the peak for these black-box models when setting δ to **0.5**.

(2) The maximum perturbation ϵ: The impact of perturbation magnitude $\boldsymbol{\epsilon}$ on the attack success rates of NG-PGN is illustrated in Figure 6b. We observe that a larger perturbation results in higher attack success rates. To balance the attacks success rates and the imperceptibility of adversarial examples, we set the perturbation size to **16/255** in our experiments.

(3) The size of neighborhood $\boldsymbol{r}$: In SAM, the size of neighborhood $\boldsymbol{r}$ constrains the distance between the worst-case and the current point, which also applies to manifolds with non-Euclidean measures. As shown in Figure 6c, we study the influence of the $\boldsymbol{r}$ in the NG-PGN and NG-SAM. As we increase $\boldsymbol{r}$ from $\boldsymbol{\alpha}$ to $\mathbf{5} \cdot \boldsymbol{\alpha}$, the transferability increases and achieves the peak at $\boldsymbol{r} = \mathbf{3} \cdot \boldsymbol{\alpha}$.



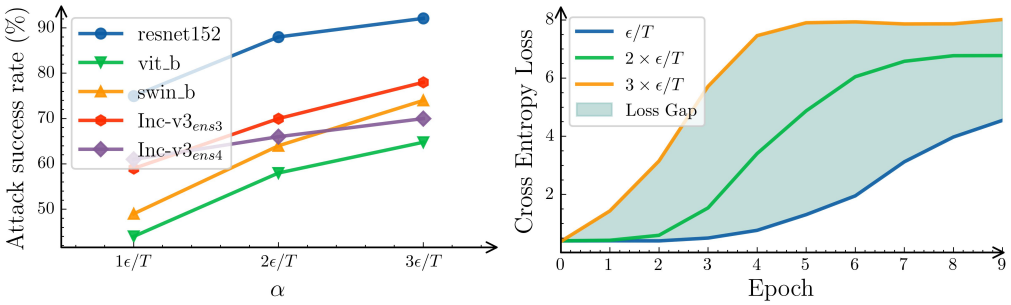
(**a**) (**b**) (**c**)

Figure 6. Impact of hyperparameters on the performance of the proposed Method. (**a**) shows the attack success rates of NG-PGN vary with different values of δ. The attack success rates peak at $\delta = 0.5$. (**b**) shows the attack success rates of NG-SAM increases with the increase of values of α. (**c**) shows the loss gap between NG-SAM using different values of α. The blue line indicates the step size $\alpha = \epsilon/T$, while the green line represents $\alpha = 2 \cdot \epsilon/T$.

4.10. Further Analysis

(1) Whydoes NG-SAM underperform NG-PGN? Under small iteration step sizes, NG-SAM demonstrates poor performance, particularly exhibiting lower transferability on normally trained models. We argue that the SAM strategy primarily guides adversarial examples toward flat regions of the loss landscape, without significantly increasing the loss value. To support this claim, we conducted a comparison experiment with NG-SAM at different step sizes. Figure 7a shows the attack success rates of NG-SAM vary with different values of α, and Figure 7b shows the loss gap between NG-SAM using different step sizes. The attack success rates increase as the loss value rises, further confirming our argument.

(2) Attack large visual language models: Furthermore, to illustrate the effectiveness of adversarial examples generated by NG-PGN, we conducted experiments on three large visual language models: GPT-4o (https://chat.openai.com/chat, accessed on 29 December 2025), ChatGLM-4.1v (https://chatglm.cn/main/alltoolsdetail?lang=en, accessed on 29 December 2025), and Google Gemini 2.0 (https://gemini.google.com/app, accessed on 29 December 2025). The surrogate models for generating adversarial examples include ViT-B and Swin-B. As shown in Figure 8, these models are fooled to varying degrees when describing the adversarial images. For instance, ChatGPT-4o and ChatGLM incorrectly identified the species, while Gemini miscounted the number of animals in the adversarial example.



(**a**) (**b**)

Figure 7. (**a**) shows the attack success rates of NG-SAM increases with the increase of values of α. (**b**) shows the loss gap between NG-SAM using different values of α. The blue line indicates the step size $\alpha = \epsilon/T$, the green line represents $\alpha = 2 \cdot \epsilon/T$, and the yellow line represents $\alpha = 3 \cdot \epsilon/T$.

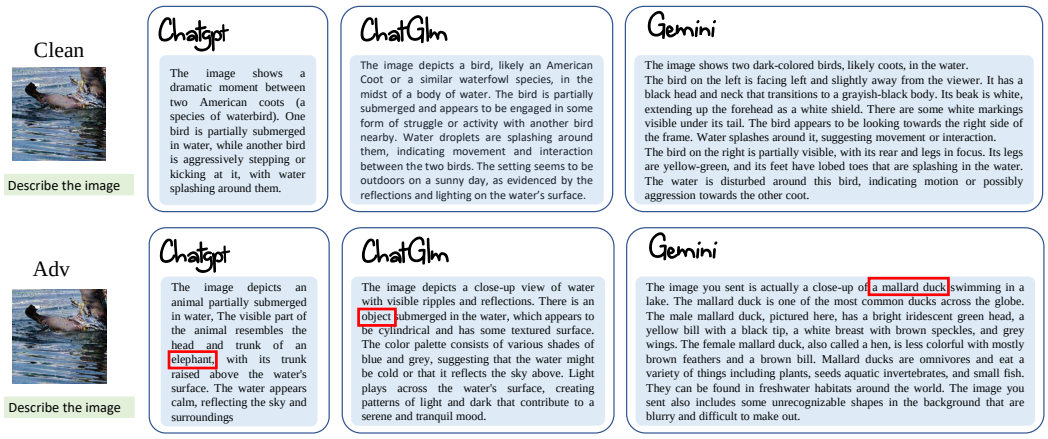


Figure 8. We evaluate the ability of NG-PGN-generated adversarial examples to mislead ChatGPT, ChatGLM, and Google Gemini. The top and second rows depict the model outputs for clean inputs and their corresponding adversarial examples, respectively. Results in red frames demonstrate that the adversarial inputs effectively induce misinterpretation of the visual subject by these large models.

5. Conclusions

Transferability allows adversarial examples to effectively attack a target model without any prior knowledge of it. It is widely recognized that adversarial examples at flat maxima of the loss landscape can exhibit higher transferability. SAM and PGN are efficient methods for finding flat maxima and have been shown to enhance adversarial transferability in transfer-based attacks. In this paper, we build upon the concept of flat maxima but incorporate the information geometry of the image space when defining the neighborhood around the input. To reduce computational effort, we approximate the Fisher information matrix for adversarial examples empirically as a diagonal matrix and estimate it within a model ensemble setting. Extensive experiments on the ImageNet-compatible, CIFAR-10, and CIFAR-100 datasets demonstrate that combining the proposed manifold extensions with a model ensemble attack generates adversarial examples with higher transferability compared to SAM, PGN, and other transfer-based attacks. Finally, although existing works have verified that flat maxima can improve transferability from a generalization perspective, a theoretical analysis is still not enough. In addition, the current implementation adopts fixed hyperparameters, such as step size and neighborhood radius, following standard settings in prior work. Future research could explore adaptive step-size strategies or dynamically adjusted neighborhood sizes to further improve efficiency and robustness. We hope our work further elucidates the mechanism of flat maxima in generating highly transferable adversarial examples.

Author Contributions: Conceptualization, H.X.; methodology, Y.L.; data curation, Y.L.; writing—original draft preparation, Y.L.; writing—review and editing, H.X.; visualization, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: ImageNet-compatible dataset presented in the study is openly available in https://github.com/cleverhans-lab/cleverhans/tree/master/cleverhans_v3.1.0/examples/nips17_adversarial_competition/dataset, accessed on 29 December 2025.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199.
2. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572.
3. Guo, R.; Chen, Q.; Liu, H.; Wang, W. Adversarial robustness enhancement for deep learning-based soft sensors: An adversarial training strategy using historical gradients and domain adaptation. *Sensors* **2024**, *24*, 3909. [CrossRef] [PubMed]
4. Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; Li, J. Boosting adversarial attacks with momentum. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 9185–9193.
5. Lin, J.; Song, C.; He, K.; Wang, L.; Hopcroft, J.E. Nesterov Accelerated Gradient and Scale Invariance for Adversarial Attacks. In *Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020*; OpenReview.net: Alameda, CA, USA, 2020.
6. Wang, X.; He, K. Enhancing the transferability of adversarial attacks through variance tuning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 1924–1933.
7. Wang, X.; Lin, J.; Hu, H.; Wang, J.; He, K. Boosting adversarial transferability through enhanced momentum. *arXiv* **2021**, arXiv:2103.10609. [CrossRef]
8. Ge, Z.; Liu, H.; Xiaosen, W.; Shang, F.; Liu, Y. Boosting adversarial transferability by achieving flat local maxima. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 70141–70161.
9. Qin, Z.; Fan, Y.; Liu, Y.; Shen, L.; Zhang, Y.; Wang, J.; Wu, B. Boosting the transferability of adversarial attacks with reverse adversarial perturbation. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 29845–29858.
10. Yang, X.; Lin, J.; Zhang, H.; Yang, X.; Zhao, P. Improving the transferability of adversarial examples via direction tuning. *arXiv* **2023**, arXiv:2303.15109. [CrossRef]
11. Zhu, H.; Ren, Y.; Sui, X.; Yang, L.; Jiang, W. Boosting adversarial transferability via gradient relevance attack. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 4741–4750.
12. Liang, L.; Hu, X.; Deng, L.; Wu, Y.; Li, G.; Ding, Y.; Li, P.; Xie, Y. Exploring Adversarial Attack in Spiking Neural Networks with Spike-Compatible Gradient. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 2569–2583. [CrossRef]
13. Chen, J.; Feng, Z.; Zeng, R.; Pu, Y.; Zhou, C.; Jiang, Y.; Gan, Y.; Li, J.; Ji, S. Enhancing Adversarial Transferability with Adversarial Weight Tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence, 2025*; AAAI Press: Washington, DC, USA, 2025; pp. 2061–2069.
14. Xie, C.; Zhang, Z.; Zhou, Y.; Bai, S.; Wang, J.; Ren, Z.; Yuille, A.L. Improving transferability of adversarial examples with input diversity. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 2730–2739.
15. Dong, Y.; Pang, T.; Su, H.; Zhu, J. Evading defenses to transferable adversarial examples by translation-invariant attacks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 16–20 June 2019; pp. 4312–4321.
16. Wang, X.; He, X.; Wang, J.; He, K. Admix: Enhancing the transferability of adversarial attacks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Virtual, 11–17 October 2021; pp. 16158–16167.
17. Wang, X.; Zhang, Z.; Zhang, J. Structure invariant transformation for better adversarial transferability. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 4607–4619.
18. Wei, X.; Zhao, S. Boosting adversarial transferability with learnable patch-wise masks. *IEEE Trans. Multimed.* **2023**, *26*, 3778–3787. [CrossRef]
19. Zhang, J.; Huang, J.t.; Wang, W.; Li, Y.; Wu, W.; Wang, X.; Su, Y.; Lyu, M.R. Improving the transferability of adversarial samples by path-augmented method. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 8173–8182.
20. Xiong, Y.; Lin, J.; Zhang, M.; Hopcroft, J.E.; He, K. Stochastic variance reduced ensemble adversarial attack for boosting the adversarial transferability. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–24 June 2022; pp. 14983–14992.
21. Chen, B.; Yin, J.; Chen, S.; Chen, B.; Liu, X. An adaptive model ensemble adversarial attack for boosting adversarial transferability. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 2–6 October 2023; pp. 4489–4498.
22. Chen, H.; Zhang, Y.; Dong, Y.; Yang, X.; Su, H.; Zhu, J. Rethinking model ensemble in transfer-based adversarial attacks. *arXiv* **2023**, arXiv:2303.09105.
23. Huang, H.; Chen, Z.; Chen, H.; Wang, Y.; Zhang, K. T-sea: Transfer-based self-ensemble attack on object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 20514–20523.
24. Foret, P.; Kleiner, A.; Mobahi, H.; Neyshabur, B. Sharpness-aware minimization for efficiently improving generalization. *arXiv* **2020**, arXiv:2010.01412.

25. Si, D.; Yun, C. Practical sharpness-aware minimization cannot converge all the way to optima. *Adv. Neural Inf. Process. Syst.* **2024**, *36*, 26190–26228.
26. Liu, Y.; Mai, S.; Cheng, M.; Chen, X.; Hsieh, C.J.; You, Y. Random sharpness-aware minimization. *Adv. Neural Inf. Process. Syst.* **2022**, *35*, 24543–24556.
27. Zhao, Y.; Zhang, H.; Hu, X. Penalizing gradient norm for efficiently improving generalization in deep learning. In *Proceedings of the International Conference on Machine Learning, Baltimore, MD, USA, 17–23 July 2022*; PMLR: Cambridge MA, USA, 2022; pp. 26982–26992.
28. Liang, H.; Zheng, H.; Wang, H.; He, L.; Lin, H.; Liang, Y. Exploring Flatter Loss Landscape Surface via Sharpness-Aware Minimization with Linear Mode Connectivity. *Mathematics* **2025**, *13*, 1259. [CrossRef]
29. Su, D.; Jin, L.; Wang, J. Noise-resistant sharpness-aware minimization in deep learning. *Neural Netw.* **2025**, *181*, 106829. [CrossRef]
30. Zhang, Y.; Hu, S.; Zhang, L.; Shi, J.; Li, M.; Liu, X.; Wan, W.; Jin, H. Why Does Little Robustness Help? A Further Step Towards Understanding Adversarial Transferability. In Proceedings of the 2024 IEEE Symposium on Security and Privacy (SP), Los Alamitos, CA, USA, 19–23 May 2024; p. 10. [CrossRef]
31. Zhao, C.; Fletcher, P.T.; Yu, M.; Peng, Y.; Zhang, G.; Shen, C. The adversarial attack and detection under the fisher information metric. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 5869–5876.
32. Kim, M.; Li, D.; Hu, S.X.; Hospedales, T. Fisher sam: Information geometry and sharpness aware minimisation. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge MA, USA, 2022; pp. 11148–11161.
33. Zhong, Q.; Ding, L.; Shen, L.; Mi, P.; Liu, J.; Du, B.; Tao, D. Improving sharpness-aware minimization with fisher mask for better generalization on language models. *arXiv* **2022**, arXiv:2210.05497. [CrossRef]
34. Chen, Y.; Liu, W. A theory of transfer-based black-box attacks: Explanation and implications. *Adv. Neural Inf. Process. Syst.* **2023**, *36*, 13887–13907.
35. K, A.; Hamner, B.; Goodfellow, I. NIPS 2017: Non-Targeted Adversarial Attack. Kaggle. 2017. Available online: https://kaggle.com/competitions/nips-2017-non-targeted-adversarial-attack (accessed on 29 December 2025).
36. Tramèr, F.; Kurakin, A.; Papernot, N.; Goodfellow, I.; Boneh, D.; McDaniel, P. Ensemble adversarial training: Attacks and defenses. *arXiv* **2017**, arXiv:1705.07204.
37. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
38. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
39. Liu, C.; Zhu, L.; Belkin, M. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Appl. Comput. Harmon. Anal.* **2022**, *59*, 85–116.
40. Long, Y.; Zhang, Q.; Zeng, B.; Gao, L.; Liu, X.; Zhang, J.; Song, J. Frequency domain model augmentation for adversarial attack. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 549–566.
41. Wang, K.; He, X.; Wang, W.; Wang, X. Boosting adversarial transferability by block shuffle and rotation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 16–22 June 2024; pp. 24336–24346.
42. Liu, Y.; Chen, X.; Liu, C.; Song, D. Delving into transferable adversarial examples and black-box attacks. *arXiv* **2016**, arXiv:1611.02770.
43. Yang, Z.; Li, L.; Xu, X.; Zuo, S.; Chen, Q.; Zhou, P.; Rubinstein, B.; Zhang, C.; Li, B. Trs: Transferability reduced ensemble via promoting gradient diversity and model smoothness. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 17642–17655.
44. Li, Q.; Guo, Y.; Zuo, W.; Chen, H. Making substitute models more bayesian can enhance transferability of adversarial examples. *arXiv* **2023**, arXiv:2302.05086. [CrossRef]
45. Tang, B.; Wang, Z.; Bin, Y.; Dou, Q.; Yang, Y.; Shen, H.T. Ensemble diversity facilitates adversarial transferability. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–21 June 2024; pp. 24377–24386.
46. Keskar, N.S.; Mudigere, D.; Nocedal, J.; Smelyanskiy, M.; Tang, P.T.P. On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv* **2016**, arXiv:1609.04836.
47. Neyshabur, B.; Bhojanapalli, S.; McAllester, D.; Srebro, N. Exploring generalization in deep learning. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. [CrossRef]
48. Izmailov, P.; Podoprikhin, D.; Garipov, T.; Vetrov, D.; Wilson, A.G. Averaging weights leads to wider optima and better generalization. *arXiv* **2018**, arXiv:1803.05407.
49. Jiang, Y.; Neyshabur, B.; Mobahi, H.; Krishnan, D.; Bengio, S. Fantastic generalization measures and where to find them. *arXiv* **2019**, arXiv:1912.02178. [CrossRef]
50. Qiu, C.; Duan, Y.; Zhao, L.; Wang, Q. Enhancing Adversarial Transferability Through Neighborhood Conditional Sampling. *arXiv* **2024**, arXiv:2405.16181. [CrossRef]

51. Wu, T.; Luo, T.; Wunsch, D.C. Gnp attack: Transferable adversarial examples via gradient norm penalty. In *Proceedings of the 2023 IEEE International Conference on Image Processing (ICIP)*; IEEE: Piscataway, NJ, USA, 2023; pp. 3110–3114.
52. Huang, G.; Liu, Z.; Van Der Maaten, L.; Weinberger, K.Q. Densely connected convolutional networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 4700–4708.
53. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
54. Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; Jégou, H. Training data-efficient image transformers & distillation through attention. In *Proceedings of the International Conference on Machine Learning*; PMLR: Cambridge MA, USA, 2021; pp. 10347–10357.
55. Wightman, R. PyTorch Image Models. 2019. Available online: https://github.com/rwightman/pytorch-image-models (accessed on 29 December 2025).
56. Liao, F.; Liang, M.; Dong, Y.; Pang, T.; Hu, X.; Zhu, J. Defense against adversarial attacks using high-level representation guided denoiser. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1778–1787.
57. Jia, J.; Cao, X.; Wang, B.; Gong, N.Z. Certified robustness for top-k predictions against adversarial perturbations via randomized smoothing. *arXiv* **2019**, arXiv:1912.09899. [CrossRef]
58. Naseer, M.; Khan, S.; Hayat, M.; Khan, F.S.; Porikli, F. A self-supervised approach for adversarial robustness. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 262–271.
59. Nie, W.; Guo, B.; Huang, Y.; Xiao, C.; Vahdat, A.; Anandkumar, A. Diffusion models for adversarial purification. *arXiv* **2022**, arXiv:2205.07460.
60. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
61. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520.
62. Gao, L.; Zhang, Q.; Song, J.; Liu, X.; Shen, H.T. Patch-wise attack for fooling deep neural network. In *Proceedings of the Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28. 2020, Proceedings, Part XXVIII 16*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 307–322.
63. Szegedy, C.; Ioffe, S.; Vanhoucke, V.; Alemi, A. Inception-v4, inception-resnet and the impact of residual connections on learning. In Proceedings of the AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017.
64. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
65. Hutchinson, M.F. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Commun. Stat. Simul. Comput.* **1989**, *18*, 1059–1076. [CrossRef]