

Boosting Adversarial Transferability via Negative Hessian Trace Regularization

Yunfei Long¹, Zilin Tian¹, Liguozhang^{1*}, Huosheng Xu¹
¹Harbin Engineering University

{2016064109, tzl, zhangliguo, xuhuosheng}@hrbeu.edu.cn

Abstract

*Transferability makes the black-box attacks to be practical. Recent studies demonstrate that adversarial examples situated at the flat maxima on the loss landscape tend to exhibit higher transferability and propose effective strategies to optimize adversarial examples to converge toward that region. However, these works primarily consider the first-order gradient regularization and have yet to explore higher-order geometry properties of the flat loss landscape, which may lead to suboptimal results. In this work, we propose leveraging the trace of the Hessian matrix of loss function with respect to the adversarial example as a curvature-aware regularizer. For computationally efficient, we introduce an approximation method for the trace based on stochastic estimation and finite difference. We theoretically and empirically demonstrate that the trace of Hessian matrices for adversarial examples near local loss maxima is consistently negative. Following this insight, we propose **Negative Hessian Trace Regularization (NHTR)**, explicitly penalizing the negative Hessian trace to suppress curvature in all directions. Compared to existing first-order regularization methods, NHTR can generate adversarial examples at flatter local regions. Extensive experimental results on the ImageNet-compatible and CIFAR-10 datasets show that NHTR can significantly improve adversarial transferability than the state-of-the-art attacks.*

1. Introduction

Deep learning models have achieved exceptional performance in computer vision but remain susceptible to adversarial examples [4, 14, 20, 26, 32]. The adversarial examples are maliciously crafted by introducing imperceptible perturbations to natural examples, causing models to yield erroneous predictions. A central property of adversarial examples is transferability, *i.e.*, adversarial examples generated by attacking a surrogate model can also mislead other models. This property renders black-box attacks practically

effective, posing significant threats to safety-critical applications [7, 8, 47]. Therefore, methods for generating adversarial examples with high transferability, referred to as transfer-based attacks, have garnered increasing attention.

From the perspective of improving generalization, many works have been proposed to enhance the transferability of adversarial examples, which can be divided into three categories. Gradient optimization attacks improve transferability by employing advanced gradient calculation techniques to prevent adversarial examples from becoming trapped in suboptimal local maxima [4, 10, 13, 30, 40, 42, 49]. Input transformation attacks leverage data augmentation strategies to increase input diversity, thereby mitigating overfitting of adversarial examples to the surrogate model [5, 39, 41, 43]. Model ensemble attacks generate adversarial examples by simultaneously attacking multiple surrogate models, effectively reducing the upper bound on generalization error [1, 3, 4, 17, 46]. The latter two categories can be integrated with gradient optimization attacks to further enhance the transferability.

Inspired by the observation that a flatter loss landscape promotes better model generalization [12, 48], recent empirical and theoretical studies [3, 13, 30, 31] suggest that adversarial examples situated in flat maxima—regions where the loss peaks and grows slowly—tend to exhibit higher transferability. For instance, Reverse Adversarial Perturbation (RAP) [30] generates highly transferable adversarial examples by injecting worst-case perturbations into the optimization process of maximizing the loss to achieve flat maxima. Similarly, Penalized Gradient Norm (PGN) [13] enhances transferability by imposing an additional penalty on the gradient norm of the loss function to improve the flatness of the surface. While these approaches provide effective strategies for directing adversarial examples toward flatter regions, they only focus on the first-order regularization of the gradient and have yet explored higher-order geometry properties of the flat loss landscape.

In this work, we investigate the curvature of loss with respect to adversarial examples via the Hessian matrix and propose a novel attack objective to enhance adversarial transferability. We measure curvature using the trace of the

* Corresponding author.

Hessian matrix and introduce a computationally efficient approximation method based on stochastic estimation and finite difference. Intuitively, flat local maxima for adversarial examples can be achieved by simultaneously maximizing the expected loss within the neighborhood of the adversarial example and minimizing the associated Hessian trace. However, empirical and theoretical analysis reveals that the Hessian matrix becomes consistently negative semidefinite as adversarial examples converge toward local loss maxima, leading to a negative trace. Direct minimization of this trace would paradoxically amplify sharp curvature, driving adversarial examples toward sharp loss maxima. Following this insight, we propose **Negative Hessian Trace Regularization** (NHTR), which explicitly penalizes the negative trace to suppress curvature. As a second-order regularization, NHTR penalizes curvature in all directions and can find flatter loss maxima compared to first-order methods like PGN and RAP, therefore generating adversarial examples with higher transferability.

Our main contributions can be summarized as follows:

- We investigate the curvature of loss landscape with respect to adversarial examples via the trace of the Hessian matrix. For computationally efficient, we introduce an approximation method for the trace based on stochastic estimation and finite difference.
- We theoretically and empirically demonstrate that adversarial examples near local loss maxima consistently exhibit negative trace of Hessian matrices. Following this insight, we propose Negative Hessian Trace Regularization (NHTR), explicitly penalizing the negative Hessian trace to suppress curvature to find flatter loss maxima.
- We conduct a series of experiments demonstrating that NHTR efficiently generates adversarial examples with high transferability. Furthermore, we show that NHTR outperforms first-order regularization like PGN and RAP in achieving flat loss maxima.

2. Related Work

2.1. Transfer-based Attack

Gradient optimization attacks utilize the gradient of the surrogate model to optimize objective functions. The Fast Gradient Sign Method (FGSM) [14] first exploits the gradient direction to generate perturbation in one step. Further, MI-FGSM [4] and NI-FGSM [20] extend the FGSM into iterative strategies to avoid poor local optima. VMI [40] stabilizes update directions by reducing gradient variance during optimization. EMI [42] reinforces momentum by accumulating gradients from multiple data points in the direction of the previous gradient. MIG [24] uses the integrated gradients rather to optimize adversarial perturbations. ANDA [10] enhances the transferability by approximating the Bayesian posterior of perturbation.

Input transformation attacks apply random transformations for input images before calculating gradients. SIM [20] puts forward the scale-invariant property of the deep neural network, and then averages the gradients of multiple scaled images. Admix [41] nonlinearly mixes the input image with other images randomly selected from various classes to augment the input. SSA [23] randomly scales input images and then adds noise to them in the frequency domain. BSR [38] randomly shuffles and rotates the image blocks. These methods can be combined with gradient-based attacks to further enhance transferability.

Model ensemble attacks average losses [21] or logits [4] of multi-surrogate models, then apply a gradient optimization attack to generate adversarial examples. SVRE [46] reduces gradient variance within an ensemble setting during optimization. AdaEa [1] adaptively adjust the weights of each surrogate model using an adjustment strategy. SMER [35] method uses reinforcement learning to optimize ensemble reweighing, enhancing adversarial example transferability.

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 [12, 48]. Such as SAM [12], an effective algorithm for obtaining a flatter loss landscape [12] by simultaneously minimizing the loss value and sharpness. Empirical and theoretical studies [3, 13, 30] suggest that transferability is also closely correlated with the flatness of the loss landscape at the adversarial examples. RAP [30] enhances loss flatness using the gradient of worst-case from neighborhood around adversarial examples. PGN [13] use the gradient norm of adversarial examples as a regularization, effectively positioning them at flat maxima. TPA [9] theoretically demonstrates that flatness is a contribution term of transferability and, together with the loss, forms a bound on it. These method validate the observation that adversarial examples at flat regions tend to exhibit better transferability.

2.3. Hessian-Based Generalization

The Hessian matrix plays a crucial role in analyzing the optimization and generalization of neural networks. By exploring the relationship between loss curvature and input-output behavior in deep learning, [25] provides new insights into the progressive sharpening phenomenon and investigates the generalization properties of flat minima. Wu et al. [45] introduce a normalized Hessian trace to measure the curvature of the loss landscape on both training and test sets, thereby reducing generalization error. These works aim to improve the generalization of neural networks via Hessian-based regularization. To the best of our knowledge, we first propose to penalize loss curvature at adversarial examples via Hessian trace regularization for finding flat maxima.

3. Methodology

3.1. Preliminaries

Let f denote a deep model which maps input variables \mathcal{X} to label variables \mathcal{Y} . Given an input image $x \in \mathcal{X}$ with its corresponding true label $y \in \mathcal{Y}$, let $\mathcal{B}_\epsilon(x) = \{\hat{x} : \|\hat{x} - x\|_p \leq \epsilon\}$ be a L_p -norm ball (e.g., L_1 -norm) around x with radius ϵ . The objective of untargeted adversarial attacks is formulated as the following constrained optimization problem:

$$\max_{x^{adv} \in \mathcal{B}_\epsilon(x)} \mathcal{L}(f(x^{adv}), y), \quad (1)$$

where $f(x^{adv})$ represents the output of model f for adversarial example $x^{adv} \in \mathcal{B}_\epsilon(x)$ and $\mathcal{L}(\cdot)$ is the loss function (e.g., cross-entropy loss). For simplicity of notation, we use $\mathcal{L}(x^{adv})$ to represent $\mathcal{L}(f(x^{adv}), y)$. Gradient optimization attacks [4, 13, 30, 40, 42] achieve the objective through advanced gradient iterations. For example, the iterative process of MI-FGSM [4] is expressed as follows:

$$\begin{aligned} x_{t+1}^{adv} &= \prod_{\mathcal{B}_\epsilon(x)} [x_t^{adv} + \alpha \cdot \text{sign}(\mathcal{M}_{t+1})], \\ \mathcal{M}_{t+1} &= \mu \mathcal{M}_t + \frac{\nabla_{x_t^{adv}} \mathcal{L}(x_t^{adv})}{\|\nabla_{x_t^{adv}} \mathcal{L}(x_t^{adv})\|_1}, \end{aligned} \quad (2)$$

where $x_0^{adv} = x$, 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^{adv} in the $\mathcal{B}_\epsilon(x)$ around x .

3.2. Hessian Trace Regularization

Previous works suggest that adversarial examples at a flat maxima tend to have good transferability. A central interpretation behind a flat maxima is an entire neighborhood having both high loss and low curvature. Assuming the loss function $\mathcal{L}(x)$ is second-order differentiable in a local region, we can analyze the curvature profile of the loss surface through the Hessian matrix, whose eigenvalues contain critical information about the local geometry. Let $\mathcal{H}(x)$ denote the Hessian matrix of the loss function $\mathcal{L}(x)$ with respect to the data point x , i.e.,

$$\mathcal{H}(x) = \nabla_x \nabla_x \mathcal{L}(x). \quad (3)$$

As the sum of eigenvalues, the Hessian trace $\text{tr}(\mathcal{H}(x))$ is an effective curvature metric. Penalizing Hessian trace constrains the magnitudes of Hessian eigenvalues, enhancing the chance of finding a flat local region. Following the above analysis, we propose an attack objective that maximizes the expected loss over local region $\mathcal{B}_\xi(x^{adv})$ with radius ξ around adversarial example x^{adv} while minimizing the maximum the Hessian trace within it, as follows:

$$\max_{x^{adv} \in \mathcal{B}_\epsilon(x)} \left[\mathbb{E}_{x' \in \mathcal{B}_\xi(x^{adv})} [\mathcal{L}(x')] - \lambda \max_{x' \in \mathcal{B}_\xi(x^{adv})} \text{tr}(\mathcal{H}(x')) \right], \quad (4)$$

where $\lambda \geq 0$ is the penalty coefficient. During the optimization process of this attack objective, the maximum Hessian trace is penalized to prevent adversarial examples from converging to sharp regions. However, it is impractical to directly calculate the maximum Hessian trace $\mathcal{B}_\epsilon(x)$ due to its NP-hard nature. Hence, we further propose to minimize the trace of the total Hessian of loss function with respect to the local region. Specifically, we denote the total Hessian as $\mathcal{H}(\mathcal{B}_\xi(x^{adv}))$, which can be expressed as the average of the Hessian matrices of the loss function for random samples:

$$\mathcal{H}(\mathcal{B}_\xi(x^{adv})) = \mathbb{E}_{x' \in \mathcal{B}_\xi(x^{adv})} [\mathcal{H}(x')]. \quad (5)$$

Therefore, using the obtained total Hessian 5, attack objective in Eq. 4 becomes,

$$\max_{x^{adv} \in \mathcal{B}_\epsilon(x)} \left[\mathbb{E}_{x' \in \mathcal{B}_\xi(x^{adv})} [\mathcal{L}(x')] - \lambda \text{tr}(\mathcal{H}(\mathcal{B}_\xi(x^{adv}))) \right]. \quad (6)$$

Moreover, as the trace of the total Hessian is equal to the average of the traces of each Hessian matrix,

$$\text{tr}(\mathcal{H}(\mathcal{B}_\xi(x^{adv}))) = \mathbb{E}_{x' \in \mathcal{B}_\xi(x^{adv})} [\text{tr}(\mathcal{H}(x'))], \quad (7)$$

we can obtain a more concise and general formalization, as follows:

$$\max_{x^{adv} \in \mathcal{B}_\epsilon(x)} \mathbb{E}_{x' \in \mathcal{B}_\xi(x^{adv})} [\mathcal{L}(x') - \lambda \text{tr}(\mathcal{H}(x'))]. \quad (8)$$

Generally, this attack objective introduces a penalized Hessian trace into the original loss function to seek flat local maxima and can achieve better transferability.

3.3. Hessian Trace Calculation

Calculating the Hessian trace is computationally intensive, especially for large-size input images. To address this error, we first introduce Hutchinson's trace estimator [18], giving an unbiased estimate of the trace of matrix:

$$\text{tr}(\mathcal{H}(x)) = \mathbb{E} [v \mathcal{H}(x) v^T] \quad (9)$$

provided that the elements of $v \sim N(0, I)$, where I has the same dimensions as x . Therefore, to estimate the trace $\text{tr}(\mathcal{H}(x))$, we calculate the expectation of the directional second derivative $v \mathcal{H}(x) v^T$. We further approximate this expression with finite differences.

Theorem 1. *Given that loss function $\mathcal{L}(x)$ is second-order differentiable, for $v \sim N(0, I)$, we have,*

$$v \mathcal{H}(x) v^T = \frac{1}{h^2} (\mathcal{L}(x + hv) + \mathcal{L}(x - hv) - 2\mathcal{L}(x)), \quad (10)$$

where h denotes the discretization step size.

Proof. As the loss function $\mathcal{L}(x)$ is second-order differentiable, given a discretization step size h , according to the Taylor expansion, we have,

$$\mathcal{L}(x + hv) \approx \mathcal{L}(x) + hv\nabla_x\mathcal{L}(x) + h^2v\mathcal{H}(x)v^T,$$

$$\mathcal{L}(x - hv) \approx \mathcal{L}(x) - hv\nabla_x\mathcal{L}(x) + h^2v\mathcal{H}(x)v^T,$$

therefore, add the above equations and simplify, we have an approximation of $v\mathcal{H}(x)v^T$.

By Theorem 1, we can instantiate $\text{tr}(\mathcal{H}(x))$ as:

$$\text{tr}(\mathcal{H}(x)) = \mathbb{E}_v \left[\frac{1}{h^2} (\mathcal{L}(x + hv) + \mathcal{L}(x - hv) - 2\mathcal{L}(x)) \right]. \quad (11)$$

The above estimation of Hessian trace involves an expectation over v , uniformly penalizing curvature across all directions. Following previous works [11, 27], we focus on the gradient direction, which corresponds to the high-curvature direction. Thus, we use perturbation solely along the gradient direction and finally consider the trace,

$$\begin{aligned} \text{tr}(\mathcal{H}(x)) &\approx \frac{1}{h^2} (\mathcal{L}(x + hv) + \mathcal{L}(x - hv) - 2\mathcal{L}(x)), \\ v &= \nabla_x\mathcal{L}(x) / \|\nabla_x\mathcal{L}(x)\|_1, \end{aligned} \quad (12)$$

As the local region around x is constrained by the L_1 -norm. Consequently, setting v to be the L_1 -norm of the gradient is more relevant.

3.4. Negative Hessian Trace Regularization

Adversarial examples are expected to converge toward the local loss maxima. We theoretically demonstrate that the Hessian matrix is negative semi-definite at loss maxima, which indicates the trace is negative. Let x^* is a local maxima point, we have

$$\text{tr}(\mathcal{H}(x^*)) < 0. \quad (13)$$

Furthermore, we prove that the examples near the local maxima also satisfy the Hessian trace to be negative. Therefore, we can observe that in the optimization process of adversarial examples toward loss maxima, the trace of the total Hessian in the neighborhood near adversarial examples is consistently negative. A detailed proof is provided in **Appendix**. While the curvature regularizer in Eq. 8 aims to reduce the Hessian trace, it obtains a counterproductive effect, aggravating adversarial examples move toward sharp maxima. Therefore, we use negative hessian trace as a regularizer, which modifies the exception in Eq. 8 as follows:

$$\begin{aligned} &\mathbb{E}_{x'} [\mathcal{L}(x') - \lambda(-\mathcal{H}(x'))] \\ &= \mathbb{E}_{x'} [\mathcal{L}(x') - 2\lambda\mathcal{L}(x') + \lambda(\mathcal{L}(x' + hv) + \mathcal{L}(x' - hv))], \end{aligned} \quad (14)$$

where the $\frac{1}{h^2}$ is absorbed by the penalty coefficient λ .

Algorithm 1: Optimize Adversarial Example using Negative Hessian Trace Regularization with MI-FGSM as the Baseline.

Input: Surrogate network f ; a natural example x with ground-truth label y ;

Parameters : The perturbation magnitude ϵ ; the number of iteration T ; the decay factor μ ; the upper bound of neighborhood, ξ ; the number of randomly sampled examples, N ; the discretization step size, h ; the balanced coefficient, δ ;

Output: An adversarial example x^{adv} ;

```

1 Initialize:  $\alpha = \epsilon/T$ ;  $\mathcal{M}_0 = 0$ ;  $x_0^{adv} = x$ ;
2 for  $t = 1$  to  $T$  do
3   Set  $g = 0$ 
4   for  $n = 1$  to  $N$  do
5     Randomly sample an example  $x' \in \mathcal{B}_\xi(x_t^{adv})$ 
6     Calculate the gradient at the sample  $x'$ ,
        $g' = \nabla_{x'}\mathcal{L}(x')$ 
7     Add perturbation along the gradient direction
        $x_1^* = x' - h\frac{g'}{\|g'\|_1}$ ,  $x_2^* = x' + h\frac{g'}{\|g'\|_1}$ 
8     Calculate the gradient
        $g^* = \frac{1}{2}(\nabla_{x_1^*}\mathcal{L}(x_1^*) + \nabla_{x_2^*}\mathcal{L}(x_2^*))$ 
9     Accumulate the updated gradient by
        $g = g + \frac{\delta g' + (1-\delta)g^*}{N}$ 
10  end
11  Update momentum
12  Get  $\mathcal{M}_{t+1} = \mu\mathcal{M}_t + \frac{g}{\|g\|_1}$ 
13  Update adversarial example
14   $x_{t+1}^{adv} = \prod_{\mathcal{B}_\epsilon(x)} [x_t^{adv} + \alpha \cdot \text{sign}(\mathcal{M}_{t+1})]$ 
15 end
16 return:  $x^{adv} = x_T^{adv}$ .
```

Set a balanced coefficient $\delta = 1 - 2\lambda$, we have final attack objective,

$$\max_{x^{adv} \in \mathcal{B}_\epsilon(x)} \mathbb{E}_{x'} \left[\delta \cdot \mathcal{L}(x') + \frac{1-\delta}{2} (\mathcal{L}(x' + hv) + \mathcal{L}(x' - hv)) \right], \quad (15)$$

where $x' \in \mathcal{B}_\xi(x^{adv})$ and v is set as $\nabla_x\mathcal{L}(x) / \|\nabla_x\mathcal{L}(x)\|_1$.

We term this novel attack objective as **Negative Hessian Trace Regularization (NHTR)** and calculate the expectation by employing a random sampling estimation through uniform sampling within the neighborhood. In essence, our NHTR jointly maximizes the worst-case and best-case perturbations near the random examples from the neighborhood, promoting adversarial examples toward a smoother, flatter region with loss maxima. Since NHTR is derived by optimizing Eq. 8, it can be seamlessly integrated with existing gradient optimization attacks and input transformation attacks, leveraging their strengths to further improve adversarial transferability. The details of integrating NHTR with MI-FGSM are outlined in Algorithm 1.

Attack	Inc-v3 \Rightarrow						Inc-v4 \Rightarrow					
	Res-152	Dense-121	IncRes-v2	Swin-B	Deit-B	Avg.	Res-152	Dense-121	IncRes-v2	Swin-B	Deit-B	Avg.
MI	29.7	50.8	46.4	12.1	17.1	31.2	35.1	53.0	46.6	14.5	17.6	33.4
NI	42.9	67.1	68.4	17.6	26.8	44.6	53.4	74.6	73.9	28.4	31.9	52.4
VMI	48.7	69.2	70.0	24.0	28.9	48.2	55.2	71.5	70.5	28.9	30.4	51.3
VNI	47.6	70.0	73.1	21.0	26.1	47.6	52.2	69.7	72.2	26.3	26.5	49.4
MIG	40.1	65.0	63.9	17.3	26.0	42.5	49.4	71.9	67.6	20.3	29.7	47.8
EMI	49.3	74.0	76.5	19.0	24.4	48.6	55.1	77.2	77.0	22.0	25.4	51.3
RAP	55.7	77.6	76.2	30.8	28.7	53.8	57.3	75.0	70.6	41.2	41.2	57.1
PGN	65.4	83.3	89.6	37.8	44.6	64.1	69.9	83.8	87.7	43.2	47.9	66.5
ANDA	68.3	85.6	82.3	38.7	43.2	63.6	70.5	82.6	84.3	45.7	48.0	66.2
NHTR	71.4	89.0	94.2	42.6	51.3	69.7	74.4	89.9	93.2	50.8	53.1	72.3

Attack	Res-101 \Rightarrow						ViT-B \Rightarrow					
	Res-152	Dense-121	IncRes-v2	Swin-B	Deit-B	Avg.	Res-152	Dense-121	IncRes-v2	Swin-B	Deit-B	Avg.
MI	54.8	50.0	18.9	12.6	13.4	29.9	27.3	41.7	34.3	36.6	66.7	41.3
NI	73.2	67.0	33.9	18.7	18.6	42.3	35.8	51.7	46.3	40.1	77.1	50.2
VMI	71.3	63.9	40.3	27.4	28.1	46.2	33.4	49.4	40.3	47.4	76.0	49.3
VNI	81.7	71.1	45.5	28.1	26.7	50.6	31.3	47.5	37.1	44.1	74.0	46.8
MIG	66.7	69.2	51.4	32.0	36.6	51.2	36.5	54.9	34.4	39.2	74.5	47.9
EMI	87.1	78.5	43.6	20.0	20.4	49.9	40.2	56.2	49.4	49.6	85.0	56.1
RAP	89.3	79.3	62.8	43.7	39.6	62.9	42.3	60.0	50.1	51.2	87.9	58.3
PGN	86.7	83.7	72.3	50.9	52.7	69.3	45.5	60.3	57.2	61.0	90.4	62.9
ANDA	89.6	86.5	73.6	54.2	52.9	71.4	46.4	60.5	56.4	60.7	91.5	63.1
NHTR	94.9	93.8	83.6	61.1	60.6	78.8	51.4	66.1	60.3	66.6	94.8	67.8

Table 1. The untargeted black-box attack success rates of various gradient-based attacks in the single model setting. The adversarial examples are crafted on Inc-v3, Inc-v4, ResNet-101, and ViT-B by MI, NI, PI, VMI, VNI, EMI, RAP, MIG, PGN, ANDA and our NHTR attack methods, respectively. The target models are all normally trained models.

4. Experiments

4.1. Experimental Setup

Dataset. Following the protocol established in previous studies [30, 40, 42, 48], we conduct our experiments on two widely used datasets, ImageNet-compatible [19] and CIFAR-10, each containing 1,000 images along with their corresponding true labels. The detailed experimental setting and results of attacks on CIFAR-10 are shown in **Appendix**.

Black-box models. To validate the effectiveness of our methods, we test attack performance in comprehensive models, including both Convolutional Neural Networks (CNNs) and Vision Transformers (ViTs): Inception-v3 (Inc-v3) [33], Inception-v4 (Inc-v4), Inception-ResNet-v2 (IncRes-v2) [34], ResNet-101 (Res-101), ResNet-152 (Res-152) [15], DenseNet-121 (Dense-121) [16], ViT-Base (ViT-B) [6], Swin-Base (Swin-B) [22] and Deit-Base (Deit-B) [36], all available from **Timm** (a widely used pretrained models library) [44]. We select four widely used network architectures: Res-101, Inc-v3, Inc-v4, and ViT-B as surrogate models, and the remaining as target models. We also consider adversarially trained models, including Inc-v3_{ens3}, Inc-v3_{ens4}, and Inc-Res-v2_{ens}. Besides, we consider defense methods including RS [37], NRP [28], Diff-

pure [29] and RDC [2], which have demonstrated robustness against black-box attacks.

Baselines. We compare the proposed NHTR with various gradient optimization attacks, including MI-FGSM (MI) [4], NI-FGSM (NI) [20], VMI, VNI [40], EMI [42], MIG [24], and ANDA [10]. We also integrate NHTR with various input transformations to validate its generality, such as DIM [39], SI-NI [20], Admix [41], SSA [23], BSR [38]. Additionally, we conduct a detailed comparison with other finding flat maxima methods, *i.e.*, PGN [48] and RAP[30].

Hyper-parameters. We set the maximum perturbation ϵ as $16/255$, the number T of iterations as 10, the step size as $\alpha = \epsilon/T$, and the decay factor for MI as 1.0. For our NHTR, we set the discretization step size h as 2α , the radius ξ of neighborhood is set as $3 \cdot \epsilon$, and the number of sampling from neighborhood as 20. For the compared methods, we use the optimal hyper-parameters as reported in their respective papers. All experiments are built on a GeForce RTX 4090 GPU using the PyTorch implementation.

Extra Experiments. Due to space limitations, we report additional experiments in the **Appendix**, including: i) ablation studies on the maximum perturbation, the discretization step size, and the sampling numbers from neighborhood; ii) attack performance to Large Visual Language Models.

Attack	Inc-v3 \Rightarrow							Inc-v4 \Rightarrow						
	Inc-v3 _{ens3}	Inc-v3 _{ens4}	Inc-v2 _{ens}	RS	NRP	Diffpure	RDC	Inc-v3 _{ens3}	Inc-v3 _{ens4}	Inc-v2 _{ens}	RS	NRP	Diffpure	RDC
VMI	41.2	42.3	24.8	70.6	40.3	6.9	10.5	41.0	40.2	26.3	71.3	41.5	7.5	11.7
EMI	33.2	31.1	18.2	69.4	35.2	6.2	10.1	29.6	28.2	16.8	68.2	38.1	7.1	11.3
RAP	54.3	48.4	38.7	71.6	29.5	7.6	12.7	54.3	48.4	45.7	72.5	31.9	8.5	14.1
PGN	65.4	65.9	45.5	65.3	54.2	7.1	15.4	66.3	64.2	46.2	66.4	53.1	8.0	16.5
ANDA	66.8	64.7	48.4	74.6	50.9	8.6	17.8	64.2	61.1	47.8	73.9	51.8	9.4	18.6
NHTR	70.8	70.5	50.5	78.9	60.1	11.3	20.1	69.3	67.8	48.4	78.4	61.6	11.6	21.5

Table 2. The untargeted black-box attack success rates (%) of transfer-based methods against robust defense methods, using Inc-v3 and Inc-v4 as surrogate models respectively. Notably, for DiffPure and RDC, we set the maximum perturbation of all attack methods as $\epsilon = 4/255$ and $8/255$ to align with their default parameter settings respectively.

Attack	Normally trained models						Adversarially trained models				
	Res-152	Dense-121	IncRes-v2	Swin-B	Deit-B	Avg.	Inc-v3 _{adv}	Inc-v3 _{ens3}	Inc-v3 _{ens4}	Inc-v2 _{ens}	Avg.
MI	65.1	68.7	64.2	24.9	27.6	50.1	30.3	26.9	24.8	15.5	24.4
VMI	88.7	90.6	90.3	47.4	46.0	72.6	55.7	53.1	49.8	37.9	49.1
EMI	92.2	93.9	93.5	48.4	46.0	74.8	50.9	46.1	44.9	29.6	42.9
RAP	91.3	94.7	92.3	44.5	42.6	73.1	44.2	27.4	25.5	14.8	28.0
PGN	87.7	93.1	94.1	68.0	71.1	82.8	82.9	82.6	81.8	72.5	80.0
NHTR	95.5	97.4	97.5	79.8	82.3	90.5	88.8	89.2	87.7	79.3	86.3

Table 3. The untargeted black-box attack success rates (%) of various gradient optimization attacks on normally and adversarially trained models in the model ensemble setting. The adversarial examples are generated on the ensemble models, including Inc-v3, Inc-v4, and Res-101. Here, the bolded indicates the highest attack success rates.

4.2. Comparison with SOTA methods

Attack single model. We compare the attacking performance of NHTR with state-of-the-art (SOTA) attacking methods. The adversarial examples are generated on four different models: Inc-v3, Inc-v4, Res-101, and ViT-B, respectively. We report the black-box attack success rates on normally trained models and adversarial robust methods in Table 1 and 2, respectively. The attack success rates indicate the misclassification rates of the target models when using adversarial examples as inputs. We observe a significant improvement in attack success rates on each target model when using the adversarial examples generated by NHTR. For example, when using Inc-v3 as the surrogate model, NHTR achieves an attack success rate of 94.2% on IncRes-v2, which is 47.8% higher than that of MI and 24.2% higher than VMI. When evaluated against adversarially trained defenses, our NHTR attack method demonstrates consistent superiority over gradient-based attacks, achieving a minimum 5% enhancement in attack success rate compared to state-of-the-art methods on average. These results convincingly validate the efficacy of our NHTR against both normally trained models and adversarially defense methods.

Attack ensemble model. We further validate the effectiveness of our NHTR method in an ensemble-model setting. We adopt the ensemble strategy of averaging the logit outputs of different models, which has been proven to be op-

timal in [4]. The set of models includes Inc-v3, Inc-v4, and Res-101. We verify the transferability of the generated adversarial examples on both normally and adversarially trained models. The results are presented in Table 3, which demonstrate that our NHTR method consistently achieves the highest attack success rates in the black-box setting. Compared to previous gradient-based attack methods, NHTR achieves an average success rate of 90.5% on normally trained models, outperforming VMI, RAP, and PGN by 17.9%, 17.4%, and 7.7%, respectively. Additionally, NHTR achieves an average success rate of 86.3% on adversarially trained models, outperforming VMI and EMI by 37.2% and 43.4%. These results confirm the superiority of our proposed method in adversarial attacks.

4.3. Comparison with Finding Flat Methods

To date, several methods focused on finding flat local maxima have been proposed, *i.e.*, PGN [48] and RAP [30]. We show the attack performance of our NHTR in comparison with PGN and RAP in Table 1 and 2. Our NHTR consistently performs better. For instance, when using Res-101 as the surrogate model, NHTR achieves an attack success rate of 93.8% on Dense-121, which is 10% higher than that of PGN and 14% higher than RAP. To comprehensively demonstrate the superiority of our approach, we further provide the following three in-depth comparatives.

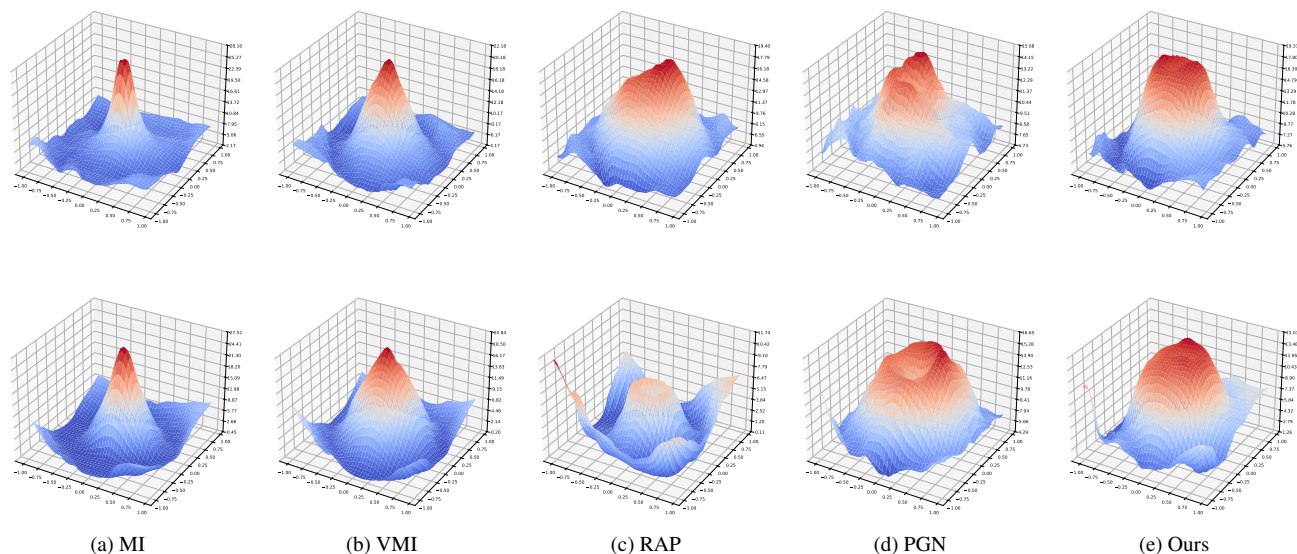


Figure 1. Visualization of loss surfaces along two random directions for two randomly sampled adversarial examples on the surrogate model Inc-v3. The center of each 2D plot represents the adversarial example generated by different attack methods—MI, VMI, RAP, PGN, and NHTR—from (a) to (e).

Comparison on Hessian trace. We calculate the trace of Hessian matrix for the adversarial examples generated by RAP, PGN and our NHTR. For ensuring fair implementation, we calculate the average of the metrics of all adversarial examples generated on Inc-v3. As reported in Table 4 row 1, NHTR can significantly reduce the Hessian trace compared to the other two strategies, which indicates smoother, flatter region.

Comparison on time cost. We verify the time cost comparison between our method, PGN and RAP on a single NVIDIA GeForce 4090 GPU. For a batch with 10 images (299×299), PGN runs in **21.1s**, NHTR in 33.3s, and RAP in 55.5s due to more iterations, as shown in Table 4 row 2. For 10 and 100 batches. NHTR takes 104.4s/812.8s, PGN takes 74.7s/689.1s, and RAP takes 442.6s/4071.8s. Qualitatively, PGN computes gradients twice per iteration ($O(2n)$, n is image size), NHTR three times ($O(3n)$), and RAP eight times ($O(8n)$). The observed runtimes align with the qualitative analysis. These results highlight that our method improves the transferability of adversarial examples while effectively managing computational costs.

Visualization of loss landscape. To validate that our NHTR guides adversarial examples toward convergence in flat maxima regions, we compare the loss surface maps of adversarial examples generated by baseline methods with those generated by NHTR. The surrogate model used is Inc-v3. The loss surface maps are visualized in Figure 1, where each row represents the visualization of one image. For example, the adversarial examples generated by MI-FGSM are located in sharpness maxima, while those generated by

NHTR are located in smoother and flatter regions.

Method	RAP	PGN	NHTR
Hessian Trace (Negative)	12.5	9.3	5.2
Time Cost (s)	55.6	21.1	33.3

Table 4. The trace of Hessian matrix at adversarial examples generated by RAP, PGN, and NHTR and the comparison of the time cost of these methods.

4.4. Integration with Input Transformation Attacks

With its high flexibility and generality, our NHTR can be simultaneously integrated with gradient-based attacks and input transformation methods to further enhance adversarial example transferability. To assess performance, we combine NHTR with advanced input transformation methods, *i.e.*, DIM, Admix, SSA and BSR, using MI-FGSM as the baseline method. Inc-v3 is selected as surrogate model. We report the results in Table 5. The results show that combined attacks demonstrates clear improvements over the baseline attack. For instance, when targeting ResNet-152, NHTR improves the attack success rates of the three input transformation attacks by **36.4%**, **34.8%**, and **28.1%**, respectively, with MI-FGSM as the baseline method. This demonstrates the scalability and capacity of our method to effectively enhance the attack success rates of transfer-based black-box attacks when combined with existing approaches.

Base	Attack	Normally trained models					Adversarially trained models			
		Res-152	Dense-121	IncRes-v2	Swin-B	Deit-B	Inc-v3 _{adv}	Inc-v3 _{ens3}	Inc-v3 _{ens4}	Inc-v2 _{ens}
DIM	MI	42.3	66.3	66.2	16.9	25.9	39.1	35.5	35.7	19.9
	+NHTR	78.7	91.5	95.6	53.6	64.3	74.3	75.5	75.6	54.1
Admix	MI	44.8	67.3	63.5	18.3	26.4	30.9	28.9	29.6	15.6
	+NHTR	79.6	92.3	94.0	56.4	65.8	75.6	76.0	75.9	56.7
SSA	MI	54.2	70.7	69.5	31.6	40.6	46.7	45.4	44.6	24.9
	+NHTR	82.3	93.5	96.6	70.5	69.4	78.2	79.4	83.8	67.3
BSR	MI	65.6	89.6	87.2	29.0	36.1	48.6	48.5	46.9	28.1
	+NHTR	84.7	94.5	98.2	74.6	72.5	79.2	81.7	85.1	68.8

Table 5. The untargeted attack success rates (%) of our NHTR method, when it is integrated with DIM, Admix, SSA, and BSR, respectively.

4.5. Visualization of attack performance

To intuitively show influence, we visualize the heatmaps of the ResNet-101 model on clean images and adversarial examples generated by NHTR and MI in Figure 2. The surrogate model is Inc-v3. Figure 2 (a) shows the attention of the ResNet-101 model on the clean images. Red areas indicate regions where the model focuses more. As shown in Figure 2 (b), the attention of the ResNet-101 model fails to change on the adversarial examples generated by MI, while the attention on the adversarial examples generated by NHTR dramatically changes as shown in Figure 2 (c).

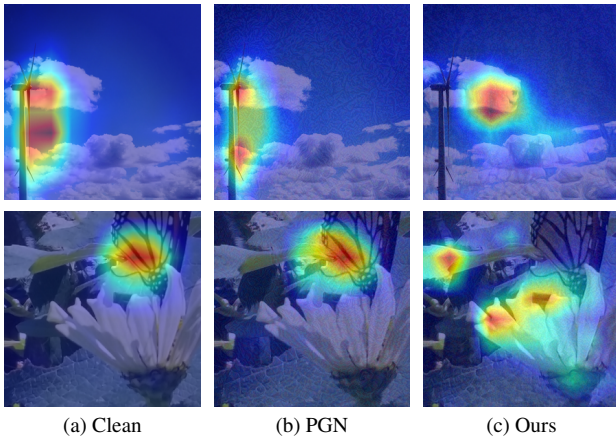


Figure 2. (a-c) show the heatmaps on clean images and corresponding adversarial examples generated by MI and NHTR.

4.6. Ablation Study

1. **The balanced coefficient δ .** In Sec. 3.3, we introduce a balanced coefficient δ to represent the penalty coefficient λ . As shown in Figure 3 (a), we study the influence of the δ in the black-box attack. As we increase δ , the transferability achieves the peak for these black-box models when $\delta = 0.3$. Therefore, we set $\delta = 0.3$ in our experiments.

2. **The upper bound ξ of neighborhood.** For the proposed NHTR, we randomly sample from the neighborhood around adversarial examples and calculate the expected loss. We examine the impact of ξ on the transferability of adversarial examples and report attack success rates as ξ increases. As shown in Figure 3 (b), attack success rates on the target models peak at $\xi = 3 \cdot \epsilon$.

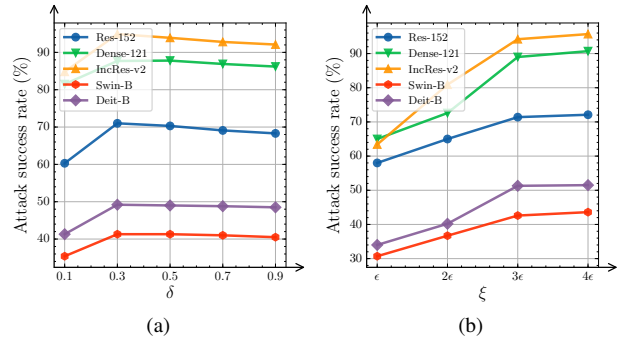


Figure 3. (a-b) show the impact of δ and ξ on the transferability of adversarial examples, respectively.

5. Conclusion

In this work, we investigate the curvature of loss via the Hessian matrix trace and introduce a computationally efficient approximation method based on stochastic estimation and finite difference. We empirically and theoretically demonstrate that adversarial examples near local loss maxima exhibit negative definite Hessian matrices. Therefore, we propose Negative Hessian Trace Regularization (NHTR), explicitly penalizing the negative Hessian trace to suppress curvature to find flat loss maxima. Extensive experimental results show that NHTR can significantly improve adversarial transferability than the state-of-the-art attacks. Moreover, compared to existing first-order regularization methods, NHTR can generate adversarial examples at flatter local regions.

6. Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants (52071111); the Fundamental Research Funds for the Central Universities (GK762026011565, GK762026011566); and the Hainan Provincial Natural Science Foundation of China (623CXTD394).

References

- [1] Bin Chen, Jiali Yin, Shukai Chen, Bohao Chen, and Ximeng Liu. An adaptive model ensemble adversarial attack for boosting adversarial transferability. In *ICCV*, pages 4489–4498, 2023.
- [2] Huanran Chen, Yinpeng Dong, Zhengyi Wang, Xiao Yang, Chengqi Duan, Hang Su, and Jun Zhu. Robust classification via a single diffusion model. In *ICML*, 2024.
- [3] Huanran Chen, Yichi Zhang, Yinpeng Dong, Xiao Yang, Hang Su, and Jun Zhu. Rethinking model ensemble in transfer-based adversarial attacks. In *ICLR*, 2024.
- [4] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, pages 9185–9193, 2018.
- [5] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *CVPR*, pages 4312–4321, 2019.
- [6] 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.
- [7] Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A Kai Qin, and Yun Yang. Adversarial camouflage: Hiding physical-world attacks with natural styles. In *CVPR*, pages 1000–1008, 2020.
- [8] Kevin Eykholt, Ivan Evtimov, Earlene 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*, pages 1625–1634, 2018.
- [9] Mingyuan Fan, Xiaodan Li, Cen Chen, Wenmeng Zhou, and Yaliang Li. Transferability bound theory: Exploring relationship between adversarial transferability and flatness. In *NeurIPS*, 2024.
- [10] Zhengwei Fang, Rui Wang, Tao Huang, and Liping Jing. Strong transferable adversarial attacks via ensembled asymptotically normal distribution learning. In *CVPR*, pages 24841–24850, 2024.
- [11] Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Stefano Soatto. Empirical study of the topology and geometry of deep networks. In *CVPR*, pages 3762–3770, 2018.
- [12] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *ICLR*, 2021.
- [13] Zhijin Ge, Xiaosen Wang, Hongying Liu, Fanhua Shang, and Yuanyuan Liu. Boosting adversarial transferability by achieving flat local maxima. In *NeurIPS*, 2023.
- [14] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.
- [15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
- [16] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017.
- [17] Hao Huang, Ziyang Chen, Huanran Chen, Yongtao Wang, and Kevin Zhang. T-SEA: transfer-based self-ensemble attack on object detection. In *CVPR*, pages 20514–20523, 2023.
- [18] Michael F Hutchinson. A stochastic estimator of the trace of the influence matrix for laplacian smoothing splines. *Communications in Statistics-Simulation and Computation*, 18(3):1059–1076, 1989.
- [19] Alex K, Ben Hamner, and Ian Goodfellow. Nips 2017: Non-targeted adversarial attack. <https://kaggle.com/competitions/nips-2017-non-targeted-adversarial-attack>, 2017. Kaggle.
- [20] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *ICLR*, 2020.
- [21] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *ICLR*, 2017.
- [22] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021.
- [23] Yuyang Long, Qilong Zhang, Boheng Zeng, Lianli Gao, Xianglei Liu, Jian Zhang, and Jingkuan Song. Frequency domain model augmentation for adversarial attack. In *ECCV*, pages 549–566, 2022.
- [24] Wenshuo Ma, Yidong Li, Xiaofeng Jia, and Wei Xu. Transferable adversarial attack for both vision transformers and convolutional networks via momentum integrated gradients. In *ICCV*, pages 4630–4639, 2023.
- [25] Lachlan Ewen MacDonald, Jack Valmadre, and Simon Lucey. On progressive sharpening, flat minima and generalisation. *arXiv preprint arXiv:2305.14683*, 2023.
- [26] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *CVPR*, pages 2574–2582, 2016.
- [27] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. In *CVPR*, pages 9078–9086, 2019.
- [28] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *CVPR*, pages 262–271, 2020.

- [29] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Animashree Anandkumar. Diffusion models for adversarial purification. In *ICML*, pages 16805–16827, 2022.
- [30] Zeyu Qin, Yanbo Fan, Yi Liu, Li Shen, Yong Zhang, Jue Wang, and Baoyuan Wu. Boosting the transferability of adversarial attacks with reverse adversarial perturbation. In *NeurIPS*, 2022.
- [31] Chunlin Qiu, Yiheng Duan, Lingchen Zhao, and Qian Wang. Enhancing adversarial transferability through neighborhood conditional sampling. *arXiv preprint arXiv:2405.16181*, 2024.
- [32] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014.
- [33] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016.
- [34] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, pages 2818–2826, 2017.
- [35] Bowen Tang, Zheng Wang, Yi Bin, Qi Dou, Yang Yang, and Heng Tao Shen. Ensemble diversity facilitates adversarial transferability. In *CVPR*, pages 24377–24386, 2024.
- [36] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357, 2021.
- [37] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. Ensemble adversarial training: Attacks and defenses. In *ICLR*, 2018.
- [38] Kunyu Wang, Xuanran He, Wenxuan Wang, and Xiaosen Wang. Boosting adversarial transferability by block shuffle and rotation. In *CVPR*, 2024.
- [39] Kunyu Wang, Xuanran He, Wenxuan Wang, and Xiaosen Wang. Boosting adversarial transferability by block shuffle and rotation. In *CVPR*, pages 24336–24346, 2024.
- [40] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *CVPR*, pages 1924–1933, 2021.
- [41] Xiaosen Wang, Xuanran He, Jingdong Wang, and Kun He. Admix: Enhancing the transferability of adversarial attacks. In *ICCV*, pages 16138–16147, 2021.
- [42] Xiaosen Wang, Jiadong Lin, Han Hu, Jingdong Wang, and Kun He. Boosting adversarial transferability through enhanced momentum. In *BMVC*, page 272, 2021.
- [43] Xiaosen Wang, Zeliang Zhang, and Jianping Zhang. Structure invariant transformation for better adversarial transferability. In *ICCV*, pages 4584–4596, 2023.
- [44] Ross Wightman. Pytorch image models. <https://github.com/rwightman/pytorch-image-models>, 2019.
- [45] Tao Wu, Tie Luo, and Donald C Wunsch II. Cr-sam: Curvature regularized sharpness-aware minimization. In *AAAI*, pages 6144–6152, 2024.
- [46] Yifeng Xiong, Jiadong Lin, Min Zhang, John E. Hopcroft, and Kun He. Stochastic variance reduced ensemble adversarial attack for boosting the adversarial transferability. In *CVPR*, pages 14963–14972, 2022.
- [47] Yang Zhang, Hassan Foroosh, Philip David, and Boqing Gong. Camou: Learning physical vehicle camouflages to adversarially attack detectors in the wild. In *ICLR*, 2018.
- [48] Yang Zhao, Hao Zhang, and Xiuyuan Hu. Penalizing gradient norm for efficiently improving generalization in deep learning. In *ICML*, pages 26982–26992, 2022.
- [49] Hegui Zhu, Yuchen Ren, Xiaoyan Sui, Lianping Yang, and Wuming Jiang. Boosting adversarial transferability via gradient relevance attack. In *ICCV*, pages 4741–4750, 2023.