# Backpropagation Path Search On Adversarial Transferability

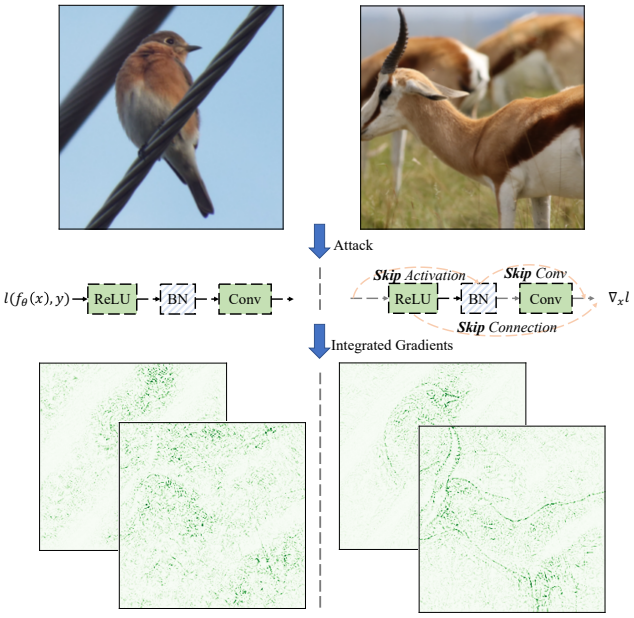
Zhuoer Xu[1], Zhangxuan Gu[1], Jianping Zhang[2*], Shiwen Cui[1], Changhua Meng[1], Weiqiang Wang[1],
[1]Tiansuan Lab, Ant Group
[2]Department of Computer Science and Engineering, The Chinese University of Hong Kong
{xuzhuoer.xze,guzhangxuan.gzx,donn.csw,changhua.mch,weiqiang.wwq}@antgroup.com
jpzhang@cse.cuhk.edu.hk

## Abstract

*Deep neural networks are vulnerable to adversarial examples, dictating the imperativeness to test the model's robustness before deployment. Transfer-based attackers craft adversarial examples against surrogate models and transfer them to victim models deployed in the black-box situation. To enhance the adversarial transferability, structure-based attackers adjust the backpropagation path to avoid the attack from overfitting the surrogate model. However, existing structure-based attackers fail to explore the convolution module in CNNs and modify the backpropagation graph heuristically, leading to limited effectiveness. In this paper, we propose backPropagation pAth Search (PAS), solving the aforementioned two problems. We first propose SkipConv to adjust the backpropagation path of convolution by structural reparameterization. To overcome the drawback of heuristically designed backpropagation paths, we further construct a Directed Acyclic Graph (DAG) search space, utilize one-step approximation for path evaluation and employ Bayesian Optimization to search for the optimal path. We conduct comprehensive experiments in a wide range of transfer settings, showing that PAS improves the attack success rate by a huge margin for both normally trained and defense models.*

## 1. Introduction

Deep neural networks (DNNs) are vulnerable to adversarial examples [32] despite their success in a wide variety of applications [13, 11, 17, 10]. It is imperative to devise effective attackers to identify the deficiencies of DNNs beforehand, which serves as the first step to improving the model's robustness. White-box attackers [26, 1, 2, 44] have complete access to the structures and parameters of victim models and effectively mislead them. However, DNNs are generally deployed in the black-box situation. To this end, transfer-based attackers, as typical black-box attackers form without access to information about victim models, have drawn increasing attention in the research community [24, 37, 42].




(a) Original Surrogate (b) PAS Surrogate

Figure 1: Feature attribution with Integrated Gradients for both the original surrogate and the surrogate with the searched backpropagation path (*i.e.*, PAS). PAS explicitly enhances the object attribution for classification and reduces the overfitting of the surrogate model to the irrelevant background, which demonstrates the effectiveness and interpretability of searched backpropagation path.

It is widely known that adversarial examples, crafted following a white-box situation against a surrogate model, are transferable to the black-box victim models due to the linear nature of DNNs [9]. To boost adversarial transferability, various methods have been proposed on different aspects,

*Corresponding author.

*e.g.*, momentum terms [4, 22], data augmentation [37, 5], structure augmentation [36, 12, 21, 7], ensemble [24, 38], and intermediate features [8, 43]. The common characteristic of the above attackers is that they reduce the overfitting of the attack on the surrogate model.

In this paper, we focus on structure-based attackers [36, 12, 21, 7], which directly rectify the backpropagation path to alleviate the overfitting issue and expose more transferability of adversarial attacks. For example, SGM [36] and LinBP [12] reduces the gradient from residual and nonlinear activation modules, respectively. However, existing structure-based attackers suffer from two critical problems circumventing their transferability. (1) They neglect the convolution module, which plays a significant role in extracting features as a basic but vital module in CNNs. The lack of adjustment for convolution in backpropagation prevents the exploitation of gradients from critical features and leads to limited effectiveness. (2) Their modification of the backpropagation path follows a heuristic manner by predefined hyper-parameters, so the selected path is non-optimal.

To explore the backpropagation of the convolution module, we follow SGM [36] to explore the backpropagation path with skip connections. Note that the inherent structure of convolution does not have skip connections for adjustment. Thus, we propose SkipConv, which decomposes the original convolution kernel into one skip kernel acting as a skip connection and the corresponding residual convolution kernel. With the two decomposed kernels, SkipConv calculates forward as usual but it is convenient to modify the backpropagation gradient via the skip kernel.

Meanwhile, we endeavor to not only resolve the heuristic problem but also unify existing structure-based attackers. Especially, we analogize the structure-based adversarial attack as a transferable backpropagation path search problem. We propose a unified and flexible framework for backPropagation pAth Search, which consists of search space, search algorithm, and evaluation metric, namely PAS. Intending to explore transferable backpropagation paths, we construct a DAG combining the skip paths of convolution, activation, and residual modules in DNNs as the search space. Next, we employ Bayesian Optimization to search for the optimal path and avoid heuristic designs. To reduce the additional overhead introduced by such a black-box search, we adopt a one-step approximation schema to efficiently evaluate the paths. Extensive experiments on the subsets of ImageNet from different surrogate models demonstrate the effectiveness of PAS against both normally trained and defense models in comparison with the baseline and state-of-the-art (SOTA) attackers.

Our main contributions can be summarized as follows:

- We propose SkipConv, which decomposes a convolution kernel into one skip kernel and the residual kernel via structural reparameterization. Such decomposition is convenient for the exploration of the convolution module during backpropagation for boosting adversarial transferability.
- We analogize the structure-based adversarial attack as a transferable backpropagation path search problem. Thus, we propose a unified framework PAS for backpropagation path search. PAS employs Bayesian Optimization to search for transferable paths in DAG-based search space. The search overhead is further reduced by one-step approximation evaluation.
- We conduct comprehensive experiments in a wide range of transfer settings. PAS greatly improves the attack success rate for normally trained models in all cases and achieves a huge margin of 6.9%$\sim$24.3% improvement against defense models. The results demonstrate the generality of PAS with various surrogate models on two benchmarks.

## 2. Preliminary

Given a clean example $\boldsymbol{x}$ with class label $y$ and a victim model $f_\theta$ parameterized by $\theta$, the goal of an adversary is to find an adversarial example $\boldsymbol{x}_{adv}$, which is constrained by $L_p$ norm with a bound $\epsilon$, to fool the model into making an incorrect prediction:

$$f_\theta(\boldsymbol{x}_{adv}) \neq y, \text{ where } \|\boldsymbol{x}_{adv} - \boldsymbol{x}\|_p \leq \epsilon \tag{1}$$

In the white-box situation, FGSM [9] perturbs the clean example $\boldsymbol{x}$ for one step by the amount of $\epsilon$ along the gradient direction. As an iterative version, I-FGSM [19] perturbs $\boldsymbol{x}$ for $T$ steps with smaller step size $\eta$ and achieves a high attack success rate:

$$\boldsymbol{x}_{adv}^{t+1} = \Pi_\epsilon^{\boldsymbol{x}} \left( \boldsymbol{x}_{adv}^t + \eta \cdot \text{sign} \left( \nabla_{\boldsymbol{x}} l \left( f_\theta \left( \boldsymbol{x}_{adv}^t \right), y \right) \right) \right) \tag{2}$$

In the absence of access to the victim model $f_\theta$, transfer-based attackers craft adversarial examples against a white-box surrogate model $f_{\theta_s,\Gamma}$ with structure hyper-parameters $\Gamma$ (*e.g.*, hyper-parameters for residual and activation modules) to achieve Eq. (1):

$$\boldsymbol{x}_{adv}^{t+1} = \Pi_\epsilon^{\boldsymbol{x}} \left( \boldsymbol{x}_{adv}^t + \eta \cdot \text{sign} \left( \nabla_{\boldsymbol{x}} l \left( f_{\theta_s,\Gamma} \left( \boldsymbol{x}_{adv}^t \right), y \right) \right) \right) \tag{3}$$

Backpropagation is essential in the process of adversarial example generation. Classical DNNs consist of several layers, *i.e.*, $f = f_1 \circ \cdots \circ f_L$, where $i \in \{1, \dots L\}$ is the layer index, and $\boldsymbol{z}_i = f_i(\boldsymbol{z}_{i-1})$ indicates the intermediate output and $\boldsymbol{z}_0 = \boldsymbol{x}$. According to the chain rule in calculus, the gradient of the loss $l$ w.r.t. input $\boldsymbol{x}$ can be then decomposed as:

$$\frac{\partial l}{\partial \boldsymbol{x}} = \frac{\partial l}{\partial \boldsymbol{z}_L} \frac{\partial f_L}{\partial \boldsymbol{z}_{L-1}} \cdots \frac{\partial f_1}{\partial \boldsymbol{z}_0} \frac{\partial \boldsymbol{z}_0}{\partial \boldsymbol{x}} \tag{4}$$

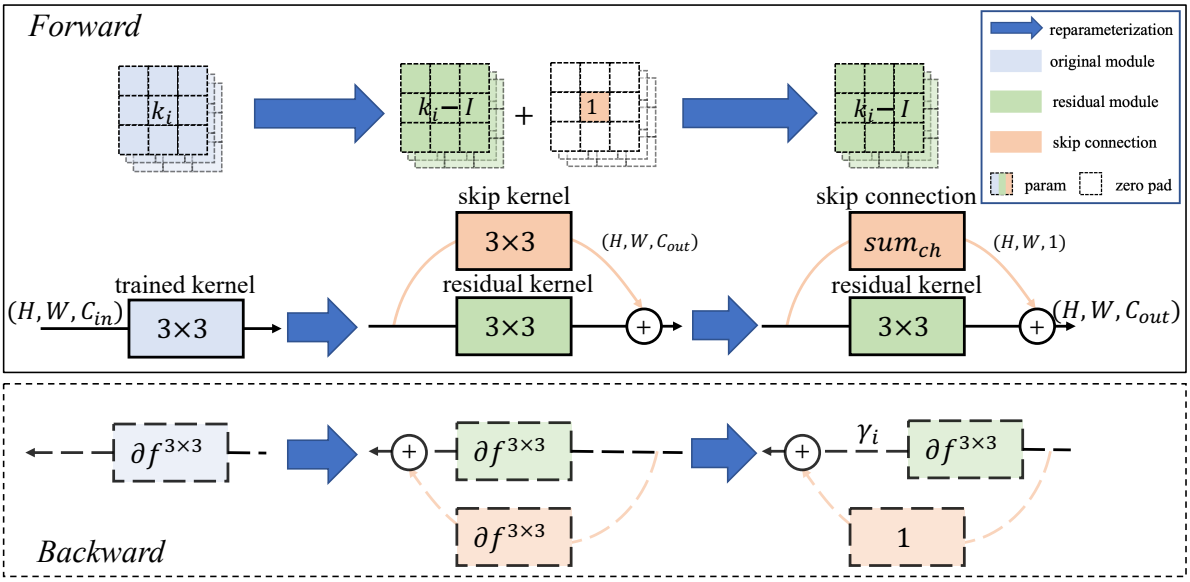



Figure 2: SkipConv: structural reparameterization of the convolution module. Take the $3 \times 3$ convolution as an example. According to convolution distributivity, the original kernel $k_i$ is decomposed into the sum of the skip kernel $I$ and the corresponding residual kernel $k_i - I$. The skip connection is implemented via structural reparameterization. SkipConv calculates forward as the original convolution but backpropagates the loss in a skip connection form. $\gamma_i$ is introduced to control the gradient from the residual kernel.

In this case, a single path is used for the gradient propagation backward from the loss to the input. Extending $f$ to a ResNet-like (with skip connections) network, the residual module in layer $i$ where $f_i^{res}(\boldsymbol{z}_{i-1}) = \boldsymbol{z}_{i-1} + f_i(\boldsymbol{z}_{i-1})$ decomposes the gradient as:

$$\begin{aligned}\frac{\partial l}{\partial \boldsymbol{z}_0} &= \frac{\partial l}{\partial \boldsymbol{z}_L} \cdots \frac{\partial f_i^{res}}{\partial \boldsymbol{z}_{i-1}} \cdots \frac{\partial \boldsymbol{z}_0}{\partial x} \\ &= \frac{\partial l}{\partial \boldsymbol{z}_L} \cdots \left(1 + \frac{\partial f_i}{\partial \boldsymbol{z}_{i-1}}\right) \cdots \frac{\partial \boldsymbol{z}_0}{\partial x}\end{aligned} \tag{5}$$

The residual module provides a gradient highway as mentioned in [13].

## 3. Methodology

In this section, we first introduce how to expand the backpropagation path of convolution via structural reparameterization in Sec. 3.1. Then, we demonstrate the three parts of PAS (*i.e.*, search space, search algorithm, and evaluation metric) in Sec. 3.2. Finally, we present the overall process.

### 3.1. Skip Convolution

Skip connections in backpropagation allow easier generation of highly transferable adversarial examples [36]. However, as a basic module to extract diverse features, convolution is neglected for adversarial transferability. Existing structure-based attackers lack the exploitation of gradients from critical features shared among different DNNs. We propose SkipConv to fill in the missing piece of the puzzle. The key of SkipConv is not to decompose the convolution in the backpropagation path without affecting the forward pass.

To achieve the characteristic, we reparameterize the structure of convolution $f_i^{conv}$ with kernel $k_i$ as shown in the Fig 2. Specifically, we decompose the original convolution kernel $k_i$ into the sum of two kernels, *i.e.*, $k_i = k_i^{res} + k_i^{skip}$. According to the distributivity of convolution, the decomposed kernels calculate forward as usual, *i.e.*, $f_i^{conv}(\boldsymbol{z}_{i-1}; k_i) = f_i^{conv}(\boldsymbol{z}_{i-1}; k_i^{res}) + f_i^{conv}(\boldsymbol{z}_{i-1}; k_i^{skip})$. Next, we view the skip convolution as an identity to make the decomposed convolution backpropagate loss in a skip connection, *i.e.*, $\partial f_i^{conv}(\boldsymbol{z}_{i-1}; k_i^{skip}) / \partial \boldsymbol{z}_{i-1} = \boldsymbol{z}_{i-1}$. Inspired by RepVGG [3], it is implemented by constructing a $1 \times 1$ skip kernel $I$ with all values 1 and zero-padding $I$ to the shape of $k_i$, *i.e.*, $f_i^{conv}(\boldsymbol{z}_{i-1}; I) = sum_{ch}(\boldsymbol{z}_{i-1})$.

All in all, we decompose the kernel as the sum of a constant skip kernel $I$ and the corresponding residual kernel $k_i - I$. The skip kernel plays the same role as the skip connection, only calculating the sum of each channel to change the channel dimension. In this way, we reparameterize the structure of convolution into the skip connection and the residual convolution:

$$\begin{aligned}f_i^{conv}(\boldsymbol{z}_{i-1}; k_i) &= sum_{ch}(\boldsymbol{z}_{i-1}) \\ &+ \gamma_i \cdot f_i^{conv}(\boldsymbol{z}_{i-1}; k_i - I) + C\end{aligned} \tag{6}$$

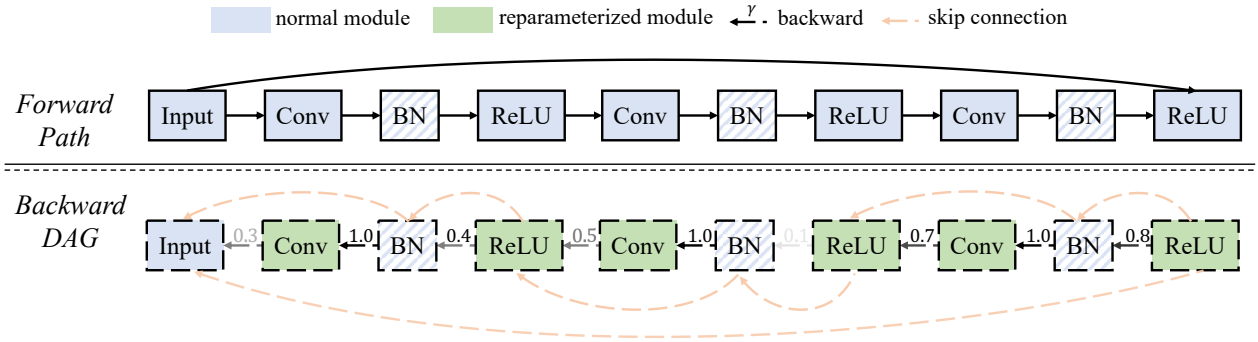



Figure 3: Example of backpropagation DAG. We construct the DAG by combining all the backpropagation paths of reparameterized modules. The color transparency indicates the weight $\gamma$ of the corresponding path to control its weight.

where the decay factor $\gamma_i \in [0, 1]$ is introduced as the weight of the residual gradient and $C$ is equal to $(1-\gamma_i) \cdot f_i^{conv}(\boldsymbol{z}_{i-1}; k_i - I)$ without gradient backward. Such SkipConv requires no fine-tuning since it calculates forward as usual. For backpropagation, $\gamma_i$ is used to relatively adjust the gradient, *i.e.*, $1 + \gamma_i \cdot \partial f_i^{conv}(\boldsymbol{z}_{i-1}; k_i - I)/\partial \boldsymbol{z}_{i-1}$.

## 3.2. PAS: Backpropagation Path Search

In this part, we introduce how PAS searches for highly transferable paths, which are evaluated by one-step approximation, in the DAG-based space.

### 3.2.1 Backpropagation DAG

Unlike works that use the existing backpropagation path of the surrogate model (*e.g.*, residual module), PAS reparameterizes original modules with skip connections and expands the path as a DAG via structural reparameterization.

**Skip Activation.** ReLU is a common activation module in neural networks. [12] demonstrates that the gradient of ReLU is sparse, which degrades adversarial transferability. The gradient of ReLU is propagated backward as $\partial f_i^{ReLU}/\partial \boldsymbol{z}_{i-1} = W_i M_i W_{i+1}$, where $M_i$ is a diagonal matrix whose entries are 1 if the corresponding entris of $W_i^T \boldsymbol{z}_{i-1}$ are positive and 0 otherwise. LinBP [12] skips the ReLU module and renormalizes the gradient passing backward as $\partial f_i^{ReLU}/\partial \boldsymbol{z}_{i-1} = \alpha_i \cdot W_i W_{i-1}$ where $\alpha_i = \|W_i M_i W_{i-1}\|_2 / \|W_i W_{i-1}\|_2$. However, the scalar $\alpha_i$ used for normalization needs to be calculated based on the weight of the front and back layers. We further devise an approximation for $\alpha_i$ and reparameterize ReLU as follows:

$$f_i^{ReLU}(\boldsymbol{z}_{i-1}) = \hat{\alpha_i} \cdot (\boldsymbol{z}_{i-1} + ReLU(-\boldsymbol{z}_{i-1})) + (1 - \hat{\alpha_i}) \cdot ReLU(\boldsymbol{z}_{i-1}) \tag{7}$$

where $\hat{\alpha_i} = \|M_i\|_2 / \|z_{i-1}\|_2$ uses the sparsity as the estimation of the re-normalizing factor.

**Skip Gradient.** SGM [36] introduces a decay parameter to control gradients from the existing skip connections, *i.e.*, $\partial f_i^{res}/\partial \boldsymbol{z}_{i-1} = 1 + \gamma \cdot \partial f_i / \partial \boldsymbol{z}_{i-1}$. We use the same SkipGrad implemented by structural reparameterization:

$$f_i^{res}(\boldsymbol{z}_{i-1}) = \boldsymbol{z}_{i-1} + \gamma_i \cdot f_i(\boldsymbol{z}_{i-1}) + C \tag{8}$$

where $C = (1-\gamma_i) \cdot f_i(\boldsymbol{z}_{i-1})$ without gradient backward.

**Directed Acyclic Graph.** We reparameterize the structure of diverse modules in DNNs and control the weight of paths by $\gamma$. For each module's backpropagation path, we control the gradient backward via SkipConv and LinReLU. For cross-module paths, we use the existing skip connection as a highway for adversarial transferability. By combining all the paths of the above modules, we construct the Directed Acyclic Graph (DAG) for gradient propagation backward. As shown in Fig 3, we use $\Gamma = \{\gamma_i\}$ to control the weight of the residual path, and hence black-box optimization can be used to search for the most transferable paths.

### 3.2.2 One-Step Approximation for Path Evaluation.

To guide the search on the DAG, we evaluate the searched paths and further propose the one-step approximation to alleviate the large overhead consumed in the search process. Intuitively, the highly transferable paths have a high attack success rate on all data for any victim model. Thus, we evaluate the impact of different steps and samples for path evaluation, which will be detailed in the appendix. The results experimentally verify that the one-step attack success rate of samples on only one white-box validation model is sufficient to distinguish paths' transferability. Based on this, an approximate schema is adopted, *i.e.*, we use such one-step evaluation as the estimation of transferability:

$$s(\Gamma; \theta_s, \theta_m, N) = \frac{1}{N} \sum_{i=0}^{N} \mathbf{1}\{f_{\theta_m}( \underbrace{\boldsymbol{x}^{(i)} + \epsilon \cdot \text{sign}(\nabla_{\boldsymbol{x}^{(i)}} l(f_{\theta_s,\Gamma}(\boldsymbol{x}^{(i)}), y^{(i)})}_{\text{one-step adversarial examples crafted against path } \Gamma} )) \neq y^{(i)}\} \tag{9}$$

Related techniques have been used in neural architecture search for evaluation [23], gradient-based hyperparameter tuning [25] and fast adversarial training for attacks [16, 39].

#### 3.2.3 Unified Framework

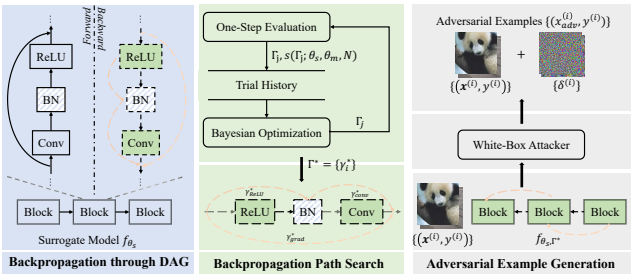



Figure 4: Overview of PAS. **Left Block** demonstrates that PAS keeps the forward path and reparameterizes the modules in the backward pass. Structural reparameterization expands the original graph into a DAG with reparameterized skip connections (dash lines). **Middle Block** employs search algorithms (e.g., Bayesian Optimization) to find backpropagation paths based on their transferability evaluation. **Right Block** illustrates that adversarial examples are crafted via the searched path of PAS surrogate.

To optimize the above objective Eq. (9), we use Bayesian Optimization[1] to search the structure parameters $\Gamma$ and combine it with Hyperband [20] to allocate resources for each trial of the sampled path. The overall procedure is shown in Algorithm 1. We first search for the most transferable path $\Gamma^*$ of the surrogate model according to Eq. (9) and then craft adversarial examples, which are transferred to unaccessible victim models.

In the search process, PAS reparameterizes the structure of the surrogate model and initializes $\Gamma$. Bayesian Optimization is used to sample the backpropagation path $\Gamma_k$. According to the sampled paths, adversarial examples for the validation dataset are crafted against $f_{\theta_s,\Gamma_k}$. PAS calculates the attack success rate on the validation model and uses it as the feedback for Bayesian Optimization for the next iteration. When predefined resources are exhausted, PAS uses the optimal structure $\Gamma^*$ to craft adversarial examples on the test set and transfers them to all victim models.

**Flexibility.** As a unified framework, PAS consists of three parts for extensions. More effective and efficient searches can be achieved through different search algorithms and evaluation metrics. It is flexible to use the new proposed augmentation in the backpropagation path to improve the diversity of the DAG-based search space. For example, we explore structural reparameterization for the Transformer's modules and use PAS to improve adversarial transferability between CNNs and Transformers in future work.

[1] https://optuna.org/

**Algorithm 1** PAS: Backpropagation Path Search on Adversarial Transferability

**Input**: Surrogate model $f_{\theta_s}$, validation model $f_{\theta_m}$, perturbation bound $\epsilon$, the number of attack steps $T$, the number of trials $N_t$

Reparameterize the structure of $f_{\theta_s}$ as $f_{\theta_s,\Gamma}$
**for** $j = 1, \ldots, N_t$ **do**
  sample $\Gamma_j$ by Bayesian Optimization according to the trial history
  evaluate $\Gamma_j$ by Eq. (9) and add it to the history
**end for**
select the most transferable path $\Gamma^*$ according to the history
**return** adversarial examples crafted against $f_{\theta_s,\Gamma^*}$

## 4. Experiments

In this section, we conduct extensive experiments to answer the following questions: Is PAS effective to craft adversarial examples with high transferability against normally trained (**RQ1**) and defense (**RQ2**) models? How do different parts of PAS affect its performance? (**RQ3**) How does PAS affect adversarial transferability? (**RQ4**)

### 4.1. Experiment Setup

**Dataset.** To compare with baselines, we report the results on two datasets: 1) Subset1000: ImageNet-compatible dataset in the NIPS 2017 adversarial competition [18], which contains 1000 images; 2) Subset5000: a subset of ImageNet validation images, which contains 5000 images and is used by SGM and IAA. We check that all models are almost approaching the $100\%$ classification success rate.

**Models.** We conduct experiments on both normally trained models and defense models. For normal trained models, we consider 8 models containing VGG19 [28], ResNet-152 (RN152) [13], DenseNet-201 (DN201) [15], Squeeze-and-Excitation network (SE154) [14], Inception-v3 (IncV3) [31], Inception-v4 (IncV4), Inception-Resnet-v2 (IncRes) [30] and ViT-B/16 (ViT) [6]. For defense models, we select 3 robustly trained models using ensemble adversarial training [33]: the ensemble of 3 IncV3 models ($\text{IncV3}_{\text{ens3}}$), the ensemble of 4 IncV3 models ($\text{IncV3}_{\text{ens4}}$) and the ensemble of 3 IncResV2 models ($\text{IncResV2}_{\text{ens3}}$). We choose different models (*i.e.*, RN152, DN201, RN50, RN121, IncV4, and IncResV2) as surrogate models to compare with different baselines. VGG19 is used as the validation model for path evaluation except in **RQ3**.

**Baseline Methods.** To demonstrate the effectiveness of PAS, we compare it with existing competitive methods, *i.e.*,

| | Attacker | RN152 | DN201 | SE154 | IncV3 | IncV4 | IncRes | ViT | $IncV3_{ens3}$ | $IncV3_{ens4}$ | $IncRes_{ens3}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| RN152 | I-FGSM | 99.91 | 51.00 | 26.32 | 23.50 | 22.58 | 18.72 | 5.10 | 12.20 | 10.80 | 5.70 |
| | Ens | 99.94 | 56.68 | 38.64 | 27.56 | 30.68 | 24.06 | 6.16 | 13.22 | 10.90 | 6.92 |
| | SVRE | 99.26 | 70.54 | 49.16 | 43.46 | 40.86 | 29.70 | 9.88 | 20.82 | 19.84 | 12.08 |
| | MI | 99.82 | 75.79 | 53.00 | 46.50 | 43.32 | 33.08 | 15.28 | 24.20 | 22.04 | 16.10 |
| | DI | 99.78 | 77.81 | 57.49 | 50.28 | 47.16 | 35.10 | 10.40 | 35.97 | 32.81 | 20.16 |
| | SGM | 99.87 | 82.76 | 61.90 | 53.16 | 49.24 | 43.30 | 11.72 | 31.57 | 27.77 | 20.84 |
| | IAA | 99.87 | 95.06 | 82.46 | 76.34 | 71.04 | 58.34 | / | 43.28 | 37.88 | 26.78 |
| | PAS | **99.96** | **96.76** | **84.98** | **83.82** | **78.82** | **77.18** | **50.26** | **59.34** | **54.46** | **44.74** |
| DN201 | I-FGSM | 59.08 | 99.89 | 40.60 | 33.80 | 32.46 | 23.80 | 6.54 | 18.16 | 15.30 | 10.40 |
| | Ens | 60.46 | 99.96 | 44.02 | 33.16 | 37.34 | 27.94 | 8.08 | 20.48 | 17.78 | 11.48 |
| | SVRE | 71.50 | 99.66 | 56.50 | 46.74 | 49.16 | 32.86 | 12.50 | 25.58 | 22.24 | 16.26 |
| | MI | 76.39 | 99.84 | 64.38 | 59.62 | 54.85 | 39.40 | 17.84 | 31.79 | 28.21 | 20.60 |
| | DI | 78.18 | 99.81 | 61.75 | 60.04 | 56.15 | 40.56 | 10.80 | 42.76 | 42.01 | 34.28 |
| | SGM | 86.60 | 99.67 | 72.20 | 62.34 | 56.36 | 45.42 | 17.66 | 41.45 | 37.85 | 29.41 |
| | IAA | 93.82 | **99.78** | 87.98 | 88.26 | 87.02 | 79.12 | / | 61.02 | 53.80 | 46.34 |
| | PAS | **96.06** | 99.76 | **90.94** | **91.00** | **88.12** | **85.96** | **51.74** | **75.08** | **72.22** | **62.28** |

Table 1: Attack success rate (%) against normally trained and defense models on Subset5000 compared to classical, ensemble-based, and structure-based attackers. The best results are in **bold**. / indicates the lack of results.

classical attackers I-FGSM [19], MI [4], DI [37] and Admix [34]; structure-based attackers SGM [36], LinBP [12], IAA [45], LLTA [7]; feature-level attackers FDA [8], FIA [35] and NAA [43]; and ensemble-based attackers Ens [24] and SVRE [38]. Since part hyperparameters differ between these methods, we directly use their paper results.

**Metrics.** Following the most widely adopted setting, we use the attack success rate as the metric. Specifically, the Attack Success Rate (ASR) is defined as the percentage of adversarial examples that successfully mislead the victim model among all adversarial examples generated by the attacker.

**Hyperparameter.** For the search process in PAS, we conduct $N_t = 2000$ trials to search on the DAG for each surrogate model, which evaluates the transferability of the backpropagation path on 256 examples in one-step attacks against the validation model (*i.e.*, VGG19). For a fair comparison of different datasets with baselines, we use the corresponding baselines' hyperparameter setting.

**Extra overhead.** The extra overhead of PAS comes from the search process. It is approximately 20 times more to generate adversarial samples than a 10-step attack on Subset1000. Note that the overhead of PAS is fixed for a given surrogate model. When the searched path is used for Subset5000 or a larger test set, the relative overhead is linearly reduced.

### 4.2. Attack Normally Trained Models (RQ1)

In this part, we investigate the transferability of attackers against normally trained models on Subset5000. We report the attack success rates of PAS, baselines, ensemble-based and structure-based attackers with RN152 and DN121 as the surrogate model on Subset5000 in Tab. 1.

Tab. 1 demonstrates that PAS beats other attackers in all black-box scenarios. Averagely, PAS achieves 88.13% ASR for RN152, which is 5.62% higher than IAA and 20.90% higher than SGM. For DN201, PAS achieves an average improvement of 2.25% in comparison with IAA, and we observe a better improvement for PAS in victim models, which are more difficult to attack (*e.g.*, 6.84% improvement against IncRes). Since SGM manually tunes the decay factors for SkipGrad and IAA uses Bayesian Optimization for SkipGrad and LinReLU, we owe the improvement to both the DAG search space and the efficient one-step approximation of PAS, which boosts adversarial transferability.

As access to the validation model (*i.e.*, VGG19) is permitted, we compare PAS with the naive and SOTA ensemble-based attackers. Tab. 1 shows the attack success rates of Ens and SVRE by simply ensemble VGG19 with white-box surrogate models. The results demonstrate that PAS takes full advantage of the additional surrogate models to evaluate the transferability of backpropagation paths, and improves the success rate by a huge margin.

Moreover, the improvement of attack success rates in various victim models (*e.g.*, classical CNNs and ViT, which is a transformer-based vision model), shows the searched path is not overclaimed to the validation model and PAS is effective in improving the adversarial transferability.

### 4.3. Attack Defense Models (RQ2)

In this part, to further verify the superiority of PAS, we conduct a series of experiments against defense models. We illustrate the attacking results against competitive baseline

| | Attacker | $IncV3_{ens3}$ | $IncV3_{ens4}$ | $IncRes_{ens3}$ | | Attacker | $IncV3_{ens3}$ | $IncV3_{ens4}$ | $IncRes_{ens3}$ |
|---|---|---|---|---|---|---|---|---|---|
| RN50 | I-FGSM | 17.3 | 18.5 | 11.2 | DN121 | I-FGSM | 21.8 | 21.5 | 13.1 |
| | SGM | 30.4 | 28.4 | 18.6 | | SGM | 36.8 | 36.8 | 22.5 |
| | LinBP | 34.5 | 32.5 | 20.9 | | LinBP | 39.3 | 38.3 | 22.6 |
| | LLTA | 50.6 | 47.3 | 33.6 | | LLTA | 59.1 | 60.5 | 46.8 |
| | PAS | **72.8** | **70.4** | **57.9** | | PAS | **70.9** | **70.8** | **57.4** |

Table 2: Attack success rate (%) against robustly trained models on Subset1000 compared to structure-based attackers. The best results are in **bold**.

| | Attacker | $IncV3_{ens3}$ | $IncV3_{ens4}$ | $IncRes_{ens3}$ | | Attacker | $IncV3_{ens3}$ | $IncV3_{ens4}$ | $IncRes_{ens3}$ |
|---|---|---|---|---|---|---|---|---|---|
| IncV4 | MI-PD | 23.9 | 24.5 | 12.5 | IncRes | MI-PD | 28.8 | 26.7 | 16.3 |
| | FIA-MI-PD | 45.5 | 42.1 | 23.5 | | FIA-MI-PD | 49.7 | 44.9 | 31.9 |
| | NAA-MI-PD | 55.4 | 53.6 | 34.4 | | NAA-MI-PD | 61.9 | 59.0 | 48.3 |
| | Admix-MI-DI | 62.4 | 69.3 | 39.7 | | Admix-MI-DI | 70.5 | 63.7 | 55.3 |
| | PAS-MI-DI | **71.5** | **66.8** | **49.7** | | PAS-MI-DI | **76.9** | **71.2** | **59.8** |

Table 3: Attack success rate (%) against robustly trained models on Subset1000 compared to classical and feature-level attackers. The best results are in **bold**.

methods under various experimental settings.

Tab. 1 shows the ASR on Subset5000. The advantages of PAS are more noticeable against defense models. The average ASR is 52.85% and 69.86% for RN152 and DN201, respectively, which is 16% more than the second-best IAA.

For the commonly used Subset1000, we directly attack defense models since most of the existing attackers have achieved a 90% attack success rate against normally trained models. The comparisons between PAS and the feature-level and structure-based attackers are presented in Tab. 2 and Tab. 3, respectively.

According to the experimental results, highly transferable attacks are crafted against defense models in the average of 23.2% and 10.9% by PAS. Although LLTA tunes the data augmentation and backpropagation structure through meta tasks, PAS searches the DAG and achieves higher transferability in Tab. 2, which shows the improvement that comes with a larger search space.

We further demonstrate that the adversarial transferability of PAS can be exploited in combination with existing methods. In contrast to the results in LLTA that DI conflicts with LinBP and leads to large performance degradation, we combine PAS with DI for transferability gains. As shown in Tab. 3, when combined with both MI and DI, PAS improves the SOTA transferability by a huge margin consistently against robustly trained models by at least 11.5%.

All in all, the experimental results identify higher adversarial transferability of PAS against defense models. Compared with existing attackers, PAS achieves a 6.9%$\sim$24.3% improvement in ASR and demonstrates the generality with various surrogate models on two benchmarks.

| | | Normal | Defense | Total |
|---|---|---|---|---|
| | PAS | 90.43 | 66.63 | 83.94 |
| | Random | 33.34 | 20.30 | 29.43 |
| DAG | w/ SkipGrad | 57.36 | 23.80 | 48.21 |
| | w/ SkipConv | 76.16 | 38.33 | 65.85 |
| Val. Model | RN50 | 87.81 | 60.52 | 80.37 |
| | IncV3 | 91.05 | 63.93 | 83.65 |
| | RN18 | 93.10 | 66.97 | 85.97 |
| | DN121 | 94.04 | 72.53 | 88.17 |

Table 4: The contribution of each part in PAS (*i.e.*, search algorithm, DAG, and validation models for path evaluation). We show the statistics of attack success rate (%) against all victims. w/ indicates the search space with the skip module.

### 4.4. Ablation Study (RQ3)

In this part, we conduct the ablation study to verify the contribution of each part in PAS by different search algorithms, removing skip modules in DAG and path evaluation with different validation models.

**Search algorithm.** We use a random search as the baseline. The result shows that randomly sampled paths lead to performance degradation. It not only validates the effectiveness of the search algorithm but also shows the importance of the paths' design.

**DAG space.** We utilize PAS on different search spaces to search for the backpropagation path and observe the ASR. We compare the commonly used SkipGrad with the pro-

posed SkipConv and the whole DAG search space The experimental results are reported in Tab. 4. SkipConv leverages the gradient from the critical features and achieves the highest attack success rate among all DAGs with a single skip module. The most transferable path is searched for by combining all skip modules and achieves at least a 13.02% improvement compared with the variants.

**Path evaluation.** In **RQ1**, we show that PAS does not overfit the validation model by the improvement of transferability on different structures of the victim model (*e.g.*, ResNet-like models, transformer-like models, and ensemble models). Further, we investigate the impact of different validation models. We select RN50, IncV3, RN18, and DN201 as validation models in Tab. 4 for path evaluation. It is surprising that even when using the surrogate model itself (*i.e.*, RN50) for evaluation, the searched path is not fully overfitted. Furthermore, despite the similar structure of RN18, the second-ranking is still achieved. According to the results, we conclude that using a one-step approximation for path evaluation plays the role of regularization that reduces the overfitting of the searched path to the validation model.

### 4.5. Adversarial Transferability with PAS (RQ4)

As network architecture shifts from manual to automated design, PAS attempts to directly use a validation model as the approximation of adversarial transferability and search highly transferable paths in backpropagation. In this part, Based on the paths searched on different architectures, we explain how PAS affects adversarial transferability and provide more insights for transfer-based attackers.

**Feature attribution.** Fig 1 shows the heat map of each input feature importance attributed by integrated gradients on the normal and PAS-reparameterized surrogate models. We observe that PAS explicitly enhances the attribution of the object for classification and excludes the influence of irrelevant background. Intuitively, it is more transferable to perturb the object. Therefore, PAS makes the surrogate model focus on class-related objects by DAG searching to boost adversarial transferability.

**Critical feature.** PAS improves adversarial transferability by selecting critical features through the weighted gradient of features in DAG. We plot the distribution of gradient weights $\Gamma = \{\gamma_i\}$ in Fig 5 and find that the final layers often keep a much smaller $\gamma_i$, *i.e.*, skipping their gradients is effective. The decay weights of intermediate modules are irregular, which means that gradients are selectively retained. To avoid significant loss of information, it is absent that gradients are all skipped in several neighboring layers (*i.e.*, smaller $\gamma$). The above conclusion is consistent with feature-level transfer-based attacks contaminating specific intermediate features.

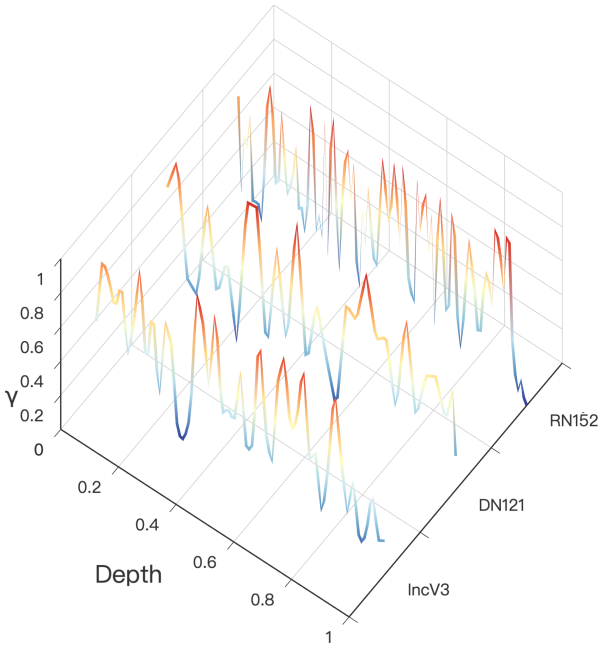



Figure 5: Examples of gradient weights $\Gamma = \{\gamma_i\}$ w.r.t its depth. The depth indicates the module's discrete index, which is then scaled to a continuous value $\in [0, 1]$. The final modules maintain a much smaller decay weight, and the intermediate gradients are selectively retrained.

## 5. Related Work

Black-box attackers can be roughly divided into query-based and transfer-based schemes. Query-based attackers estimate gradient with queries of the prediction to the victim model [27, 29]. Due to the lack of access to numerous queries in reality, part of query-based attackers focus on improving efficiency and reducing queries. In contrast, transfer-based attackers do not require any query and can be applied to unaccessible victim models.

To boost adversarial transferability, various methods have been proposed to reduce the overfitting of the attack on the surrogate model. Regarding adversarial example generation as an optimization process, [4, 22] leverage momentum terms to escape from poor local optima. To avoid overfitting with the surrogate model and specific data pattern, data augmentation [37, 5, 41] and model augmentation [24, 38] are effective strategies. Since the most critical features are shared among different DNNs, feature-level attackers [8, 43] destroy the intermediate feature maps.

From the backpropagation perspective, structure-based attackers directly rectify the backpropagation path and leverage more gradients from more useful modules to avoid overfitting. SGM [36] reduces the gradient from deep residual modules via skip connection. LinBP [12] and ConBP [40] use more linear and smoother gradients to re-

place the sparse gradients of the ReLU module.

The above attackers are designed in a heuristic manner. To automate the parts that require expert solutions, the idea of AutoML [20, 23] can be applied to adversarial attack. For a particular domain, AutoML summarizes a large search space of parameters and configurations (e.g., DAG) based on expert experience and searches for the optimal solutions using methods such as black-box optimization. During the search process, a certain metric is required to evaluate each solution. [7, 21] enhances adversarial transferability through automatic search in the designed space. However, the restricted search space leads to limited transferability. Focusing on this branch, we first expand the backpropagation path of convolution modules inspired by RepVGG [3] and form the backpropagation DAG. Then, we propose the unified framework to efficiently search in the DAG. While NAS (*i.e.*, the common AutoML application) searches the forward architecture of mixture operations, PAS reparameterizes the surrogate's backward path with skip connections and finds the transferable path.

## 6. Conclusion

In this paper, we focus on structure-based attackers and propose PAS to search backpropagation paths for adversarial transferability. To adjust the backpropagation path of convolution, we propose SkipConv, which calculates forward as usual but backpropagates loss in a skip connection form through structural reparameterization. Then, we construct a DAG-based search space by combining the backpropagation paths of various modules. To search for the optimal path, we employ Bayesian Optimization and further reduce the search overhead by a one-step approximation for path evaluation. The results of comprehensive attack experiments in a wide range of transfer settings show that PAS considerably improves the attack success rate for both normally trained and defense models. We explore structural reparameterization for the Transformer's modules, use PAS to improve adversarial transferability between CNNs and Transformers, and find better paths by advanced search algorithms and search objectives in future work.

## References


[1] Nicholas Carlini and David A. Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, pages 39–57. IEEE Computer Society, 2017.

[2] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

[3] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13733–13742, 2021.

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

[5] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 4312–4321. Computer Vision Foundation / IEEE, 2019.

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2020.

[7] Shuman Fang, Jie Li, Xianming Lin, and Rongrong Ji. Learning to learn transferable attack. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 571–579, 2022.

[8] Aditya Ganeshan, B S Vivek, and R. Venkatesh Babu. Fda: Feature disruptive attack. *international conference on computer vision*, 2019.

[9] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[10] Zhangxuan Gu, Haoxing Chen, Zhuoer Xu, Jun Lan, Changhua Meng, and Weiqiang Wang. Diffusioninst: Diffusion model for instance segmentation. *arXiv preprint arXiv:2212.02773*, 2022.

[11] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: a factorization-machine based neural network for ctr prediction. In *Proc. of IJCAI*, 2017.

[12] Yiwen Guo, Qizhang Li, and Hao Chen. Backpropagating linearly improves transferability of adversarial examples. *Advances in Neural Information Processing Systems*, 33:85–95, 2020.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[14] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

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

[16] Xiaojun Jia, Yong Zhang, Baoyuan Wu, Jue Wang, and Xiaochun Cao. Boosting fast adversarial training with learnable adversarial initialization. *IEEE Transactions on Image Processing*, 31:4417–4430, 2022.

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

[18] Alexey Kurakin, Ian Goodfellow, Samy Bengio, Yinpeng Dong, Fangzhou Liao, Ming Liang, Tianyu Pang, Jun Zhu, Xiaolin Hu, Cihang Xie, et al. Adversarial attacks and defences competition. In *The NIPS'17 Competition: Building Intelligent Systems*, pages 195–231. Springer, 2018.

[19] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018.

[20] Lisha Li, Kevin Jamieson, Giulia DeSalvo, Afshin Rostamizadeh, and Ameet Talwalkar. Hyperband: A novel bandit-based approach to hyperparameter optimization. *The Journal of Machine Learning Research*, 18(1):6765–6816, 2017.

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

[22] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *International Conference on Learning Representations*, 2019.

[23] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: differentiable architecture search. *CoRR*, abs/1806.09055, 2018.

[24] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

[25] Jelena Luketina, Mathias Berglund, Klaus Greff, and Tapani Raiko. Scalable gradient-based tuning of continuous regularization hyperparameters. 2016.

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

[27] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

[28] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[29] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

[30] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alexander A Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.

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

[32] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

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

[34] Xiaosen Wang, Xu He, Jingdong Wang, and Kun He. Admix: Enhancing the transferability of adversarial attacks. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16138–16147, 2021.

[35] Zhibo Wang, Hengchang Guo, Zhifei Zhang, Wenxin Liu, Zhan Qin, and Kui Ren. Feature importance-aware transferable adversarial attacks. *international conference on computer vision*, 2021.

[36] Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. In *International Conference on Learning Representations*, 2019.

[37] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019.

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

[39] Zhuoer Xu, Guanghui Zhu, Changhua Meng, Zhenzhe Ying, Weiqiang Wang, GU Ming, Yihua Huang, et al. A2: Efficient automated attacker for boosting adversarial training. In *Advances in Neural Information Processing Systems*.

[40] Chaoning Zhang, Philipp Benz, Gyusang Cho, Adil Karjauv, Soomin Ham, Chan-Hyun Youn, and In So Kweon. Backpropagating smoothly improves transferability of adversarial examples. In *CVPR 2021 Workshop Workshop on Adversarial Machine Learning in Real-World Computer Vision Systems and Online Challenges (AML-CV)*, volume 2, 2021.

[41] Jianping Zhang, Jen-tse Huang, Wenxuan Wang, Yichen Li, Weibin Wu, Xiaosen Wang, Yuxin Su, and Michael R Lyu. Improving the transferability of adversarial samples by path-augmented method. *arXiv preprint arXiv:2303.15735*, 2023.

[42] Jianping Zhang, Yizhan Huang, Weibin Wu, and Michael R Lyu. Transferable adversarial attacks on vision transformers with token gradient regularization. *arXiv preprint arXiv:2303.15754*, 2023.

[43] Jianping Zhang, Weibin Wu, Jen-tse Huang, Yizhan Huang, Wenxuan Wang, Yuxin Su, and Michael R Lyu. Improving

adversarial transferability via neuron attribution-based attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14993–15002, 2022.

[44] Jianping Zhang, Zhuoer Xu, Shiwen Cui, Changhua Meng, Weibin Wu, and Michael R Lyu. On the robustness of latent diffusion models. *arXiv preprint arXiv:2306.08257*, 2023.

[45] Yao Zhu, Jiacheng Sun, and Zhenguo Li. Rethinking adversarial transferability from a data distribution perspective. In *International Conference on Learning Representations*, 2021.