






# Breaking the Illusion of Security via Interpretation: Interpretable Vision Transformer Systems under Attack

Eldor Abdukhamidov , Mohammed Abuhamad , Simon S. Woo ,  
Hyoungshick Kim , and Tamer Abuhmed 

**Abstract**—Vision transformer (ViT) models, when coupled with interpretation models, are regarded as secure and challenging to deceive, making them well-suited for security-critical domains such as medical applications, autonomous vehicles, drones, and robotics. However, successful attacks on these systems can lead to severe consequences. Recent research on threats targeting ViT models primarily focuses on generating the smallest adversarial perturbations that can deceive the models with high confidence, without considering their impact on model interpretations. Nevertheless, the use of interpretation models can effectively assist in detecting adversarial examples. This study investigates the vulnerability of transformer models to adversarial attacks, even when combined with interpretation models. We propose an attack called “AdViT” that generates adversarial examples capable of misleading both a given transformer model and its coupled interpretation model. Through extensive experiments on various transformer models and two transformer-based interpreters, we demonstrate that AdViT achieves a 100% attack success rate in both white-box and black-box scenarios. In white-box scenarios, it reaches up to 98% misclassification confidence, while in black-box scenarios, it reaches up to 76% misclassification confidence. Remarkably, AdViT consistently generates accurate interpretations in both scenarios, making the adversarial examples more difficult to detect.

**Index Terms**—vision transformers, interpretation models, adversarial attack, adversarial perturbation, images

## I. INTRODUCTION

Deep learning approaches have attained cutting-edge performance in various applications, and the field continues to expand. Recently, Vision Transformers (ViTs) have been introduced as a new technique that classifies data by dividing it into spatially separated parts [14]. Generally, ViTs are considered more robust against adversarial attacks compared to Convolutional Neural Networks (CNNs) in image classification [11], [27]. Furthermore, ViT-based systems become even more robust when coupled with an interpretation model [20].

Adding interpretability as an integral component of machine learning pipelines improves their design, implementation, and adaptation by helping to detect and correct biases in the training dataset and identifying potential adversarial examples that could affect the final predictions. Moreover, interpretability ensures that only contextually relevant information is used for prediction. For example, Figure 1 shows examples of a regular adversarial attack and the corresponding interpretation.

Eldor Abdukhamidov, Hyoungshick Kim, Simon S. Woo, and Tamer Abuhmed are with the Department of Computer Science and Engineering, Sungkyunkwan University, Suwon, South Korea. (E-mail: abdukhamidov@skku.edu, swoo@skku.edu, hyoung@skku.edu, tamer@skku.edu). Mohammed Abuhamad is with the Department of Computer Science, Loyola University, Chicago, USA. (E-mail: mabuhamad@luc.edu).

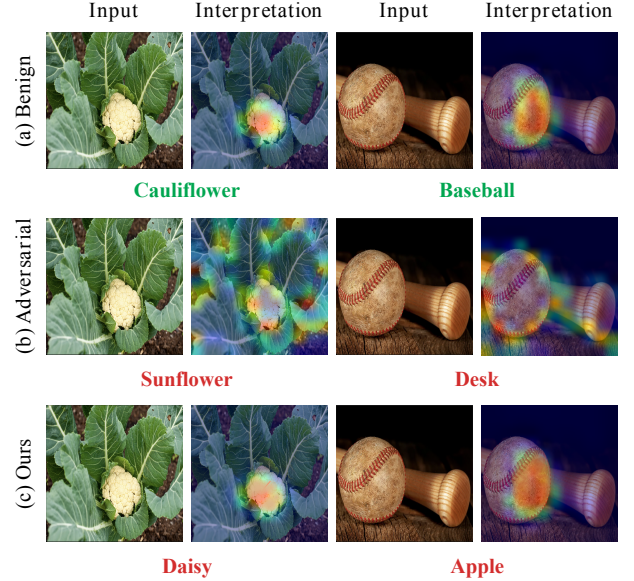


Fig. 1: Example images comparing (a) benign samples, (b) samples subject to a regular adversarial attack, and (c) samples subject to our proposed attack, along with their interpretations.

Until recently, it was believed that AI systems are more trustworthy and safe when integrated with interpretability methods and human involvement [7]. However, the image classification field has recently shown that explainable methods are vulnerable and potential targets for malicious manipulations [42], [1]. In [15], the study demonstrated that post-hoc methods are ineffective, resulting in considerable interpretation changes when a small amount of perturbation is applied to input samples. The study showed that two perceptually indistinguishable input images with the same predicted label and a small amount of perturbation could have significantly different interpretations. This is easily applicable to feature-importance interpretation methods, such as saliency maps, where the highlighted important pixels influence the model’s decision. The study showed that slightly perturbed samples could have considerably different interpretations.

As more systematic studies have been conducted on the security of CNN models, little is known regarding interpretable deep learning systems (IDLs) that employ transformer-based models. Recent studies have shown that transformer models are much more robust than CNNs [11], [27]. In this paper, we examine the security of various ViTs when coupled with interpretation models for the image classification task. We introduce a new Adversarial attack against ViTs, AdViT, that can generate adversarial samples that mislead the

target transformer classifiers, such as DeiT-B, DeiT-S, DeiT-T, Swin-B, Swin-L, T2T-ViT-7, T2T-ViT-10, ViT-B, and ViT-L, and deceive their coupled interpretation models, including Transformer Interpreter and IA-RED<sup>2</sup>, in both white-box and black-box settings. The key idea of AdViT is to exploit vulnerabilities in the interaction between the transformers and their interpretation models. Unlike traditional ViTs adversarial attacks that only focus on fooling the classification model, AdViT The attack strategically generates adversarial samples that mislead the target ViT classifiers and manipulate the output of their associated interpretation models. This dual deception is achieved through a novel attack technique that transforms the input image so that the changes are imperceptible to humans, making it even more challenging for detection methods that rely on interpretability as a security defense mechanism [34], [25]. In this study, we show that AdViT is applicable and efficient against models, such as ViT-B, Swin-T, MIT-B, and Vision-P, deployed in real-world scenarios, and is robust enough to circumvent various pre-processing defenses. We also discuss a possible interpretation-based method for detecting interpretation-guided adversarial samples using EfficientNet and a gradient-boosting classifier.

**Contributions.** Our contributions are as follows:

- **A Novel Joint Optimization Attack Framework:** We propose AdViT, the first interpretation-guided adversarial attack framework specifically designed for transformer-based IDLSes. AdViT introduces a new loss formulation that simultaneously optimizes for misclassification and interpretation similarity. The attack enables the generation of adversarial examples that fool the classification model and produce interpretations highly similar to their benign inputs.
- **Mutation-Based Black-box Attack:** We extend AdViT to black-box settings using a specialized mutation-based genetic algorithm (MGA). This approach significantly enhances transferability, allowing adversarial examples generated on a surrogate model to successfully deceive black-box ViT-based models and their interpreters, even when the attacker lacks complete model knowledge.
- **Comprehensive Evaluation:** We evaluate AdViT on multiple transformer-based architectures (DeiT, Swin, T2T-ViT, ViT) and two transformer-based interpretation methods (Transformer Interpreter, IA-RED<sup>2</sup>) in both white-box and black-box settings. The experiments show consistently high attack success rates, strong misclassification confidence, and closely-similar interpretation maps.
- **Robustness Against Real-World Models and Defenses:** We validate the practicality of AdViT by successfully attacking four ViT-based models deployed as real-world APIs. Moreover, we show that AdViT remains highly effective against several common defenses, including pre-processing transformations and adversarial training. We propose an interpretation-based ensemble detection strategy, suggesting potential technical countermeasures to mitigate the threat posed by AdViT.

**Organization.** The remainder of the paper is organized as follows: **Section II** introduces the notations, threat models, and targeted interpretation models; **Section III** describes the pro-

posed AdViT attack and its formulation; **Section IV** provides the experiments and results; **Section V** covers the related work; and **Section VI** concludes the paper.

## II. BACKGROUND

This section introduces the notations, concepts, and targeted interpretation models essential for analyzing and optimizing attacks against vision transformer-based IDLSes.

### A. Notations

This work focuses on targeting image transformer-based classification models via white-box and black-box attacks. A transformer model with  $n$  number of transformer blocks is defined as  $\mathcal{F} = (f_1 \circ f_2 \circ f_3 \circ \dots \circ f_n) \circ f_{cls}$ , where  $f_i$  is the  $i$ -th transformer block consisting of multi-head self-attention and feed-forward layers, while  $f_{cls}$  is the classification head, including the final norm layer with MLP-head. The model receives a sample image divided into  $m$  patches and produces the processed patches within self-attention layers. In terms of classification, the processed patches are submitted to the final classification head to generate the output. Each transformer block ( $f_i$ ) within the model helps extract the important features of the patches, and the classification head projects and relates the processed patches to the classes. In the paper,  $\mathcal{F}$  and  $\mathcal{F}'$  represent a white-box and black-box transformer classifier, respectively, such that  $\mathcal{F}(x) = c \in \mathcal{C}$  where  $x$  is the input and  $c$  is its category from a set of categories  $\mathcal{C}$ .

For interpretability, we consider post-hoc interpretations as this type of interpreter does not require any modification of the model architecture or parameters. An interpreter  $\mathcal{G}(x; \mathcal{F}) = m$  produces an attribution map ( $m$ ) that shows the importance of features in the input ( $x$ ) based on the output of  $\mathcal{F}$  (i.e., the value the  $i$ -th element in  $m$  ( $m[i]$ ) reflects the importance of the  $i$ -th element in  $x$  ( $x[i]$ )).

#### Threat Model: Adversarial Objectives.

The main goal of the attack is to make the classifier misclassify an adversarial sample  $\hat{x}$  and to make the interpreter generate a similar interpretation  $\hat{m}$  with its benign interpretation  $m$ . Specifically, the attack aims to generate  $\hat{x}$  such that ①  $\mathcal{F}(\hat{x}) \neq c$ ; ②  $\mathcal{G}(\hat{x}; \mathcal{F}) = \hat{m}$  s.t.  $\hat{m} \cong m$ ; and ③  $\hat{x}$  and the benign version  $x$  should be visually imperceptible.

**Threat Model: Adversarial Capabilities.** This work assumes both white-box and black-box settings. In the white-box scenario, the adversary has complete access to the transformer model  $\mathcal{F}$  and the interpreter  $\mathcal{G}$ . In the black-box scenario, the adversary has limited knowledge and access, i.e., query access, to the model  $\mathcal{F}'$  (e.g., output of the model).

### B. Targeted ViT Interpretation Models

This work considers two state-of-the-art interpreters: Transformer Interpreter [12] and IA-RED<sup>2</sup> [26]. These interpreters were chosen based on their unique approaches to model interpretation, which offer distinct advantages in terms of efficiency, interpretive depth, and the ability to identify potential vulnerabilities or biases in transformer models.

The Transformer Interpreter was selected for its comprehensive analysis of the model's decision-making process, which



is achieved through the combination of Layer-wise Relevance Propagation (LRP) and Deep Taylor Decomposition (DTD) techniques. By providing both high-level and fine-grained interpretations, this method offers a detailed understanding of the model's behavior, making it an ideal choice for studying the impact of adversarial attacks on transformer models.

On the other hand, IA-RED<sup>2</sup> was chosen for its ability to enhance the interpretive depth while minimizing redundancy in the interpretation. By identifying and removing redundant features, this method provides a more concise and informative interpretation, which can be particularly useful in identifying potential vulnerabilities or biases in the model. This focus on the most relevant features makes IA-RED<sup>2</sup> a valuable tool for analyzing the robustness of transformer models against adversarial attacks.

By comparing the results obtained using both interpretation methods, we can gain in-depth understanding of the strengths and weaknesses of transformer models in the face of adversarial attacks and identify potential strategies for improving their robustness.

**Transformer Interpreter [12].** The Transformer Interpreter is based on the Deep Taylor Decomposition [23], which propagates local relevance through layers, including skip connections and attention layers. It adopts LRP-based Propagation [8] relevance to measure the importance scores of a given sample for every layer of the transformer model, combining all those scores by relevancy scores and class-specific gradients.

The mathematical foundation of this interpreter is articulated through the equation  $C = \bar{A}^{(1)} \cdot \bar{A}^{(2)} \cdot \dots \cdot \bar{A}^{(i)}$ , where  $\bar{A}^{(i)}$  represents a modified attention map for a given block  $i$ . Each  $\bar{A}^{(i)}$  is formulated as  $\bar{A}^{(i)} = I + \mathbb{E}_h(\nabla A^{(i)} \odot R^{(s_i)})^+$ , encapsulating the attention coefficients for each token within the block. Here,  $\mathbb{E}_h$  denotes the mean across the attention heads,  $R^{(s_i)}$  the relevance score linked to the softmax operation's layer  $s_i$ ,  $\odot$  symbolizes the Hadamard operation for element-wise multiplication, and  $^+$  signifies the rectification operation  $\max(0, a)$ , effectively isolating positive contributions. An important feature of this interpreter is how it deals with skip connections in transformer blocks. It uses an identity matrix  $I$  for each token, which helps to prevent important details from getting lost as they move through different layers. This means that it can keep track of changes in the input as it goes through the transformer model, ensuring that nothing important is missed.

**IA-RED<sup>2</sup> [26].** The main idea of the Interpretability-Aware Redundancy Reduction (IA-RED<sup>2</sup>) interpreter is derived from the architecture of the multi-head self-attention layer (MSA), called the multi-head interpreter, to evaluate whether a given patch token is important. The model is divided into several groups, each containing MSA and feed-forward network (FFN) blocks and one multi-head interpreter. The multi-head interpreter evaluates the input before passing it to the blocks inside each group to calculate the informative score  $I_{ij}$ , where  $i$  and  $j$  are the positions of the input token and the group, respectively. If the score ( $I_{ij}$ ) is below the threshold (e.g., 0.5), the patch  $x_i$  is considered uninformative and will be ignored

in the following groups. The score ( $I_{ij}$ ) is calculated as:

$$I_{ij} = \frac{1}{H} \sum_h \phi(F_q^h(x_i) * F_k^h(P_j)),$$

where  $P_j$  is the policy token in the  $j$ -th multi-head interpreter to estimate the importance of the input token  $x_i$ ,  $H$  is the number of heads,  $F_q^h$  and  $F_k^h$  are the linear layers of the  $h$ -th head for the patch and policy token,  $*$  is the dot product,  $\phi$  is the sigmoid activation function. The subscript  $q$  stands for 'query,' a term borrowed from the transformer architecture, where each input element is transformed into a query vector. The subscript  $k$ , i.e., 'key,' is another concept from transformer models, where elements are also transformed into key vectors for comparison against queries.

The reinforcement method is used to optimize the interpreter to make it more efficient and accurate. The framework optimizes each multi-head interpreter using the expected gradient as follows:

$$\nabla_{W_j} J = E_{u \sim \pi} \left[ A \nabla_{W_j} \sum_{i=0}^N \log [I_{ij} u_i + (1 - I_{ij})(1 - u_i)] \right],$$

where  $E_{u \sim \pi}$  is the expected reward for the gradient computation,  $A$  is the reward score,  $W_j$  is the representation of the parameters in the  $j$ -th multi-head interpreter,  $I_{ij}$  is the informative score of the  $i$ -th input token in the  $j$ -th multi-head interpreter, and  $u_i$  is the configuration parameter accepting only binary values, in which 0 means ignoring the token.

### III. OVERVIEW OF ADViT

This section describes the AdviT attack strategy to create dual-objective adversarial examples capable of deceiving both a targeted transformer model and its interpretation model for white- and black-box scenarios.

#### A. AdviT: Attack Formulation

**Leveraging Block-specific Features.** Creating effective perturbations across all input patches requires identifying the features within each transformer block  $\{f_i\}_{i=1}^n$  that are most relevant to the target class. Our approach is inspired by the transferability logic of the ATViT attack [24], but unlike ATViT, which requires training or adapting the classification layer, we do not modify or retrain the underlying transformer weights. Instead, we introduce a mechanism that shares a single classification head  $g$  with each transformer block  $f_i$ . Formally, define:

$$\begin{aligned} \mathcal{F} &= \{f_i\}_{i=1}^n \circ g \quad \text{where, } \mathcal{F}_1 = f_1 \circ g, \quad \text{and} \\ \mathcal{F}_j &= (f_1 \circ \dots \circ f_j) \circ g \quad \text{for } j = \{2, \dots, n\}, \end{aligned}$$

where:  $f_i$  is the  $i$ -th transformer block (including multi-head self-attention and feed-forward layers),  $g$  is the shared classification head (e.g., a normalization layer plus an MLP),  $\circ$  denotes function composition of transformer blocks and the classification head.

In this formulation,  $\mathcal{F}_j$  represents a partial model that processes the input through the first  $j$  blocks  $f_1, f_2, \dots, f_j$ , and then applies  $g$  to produce logits. By connecting each

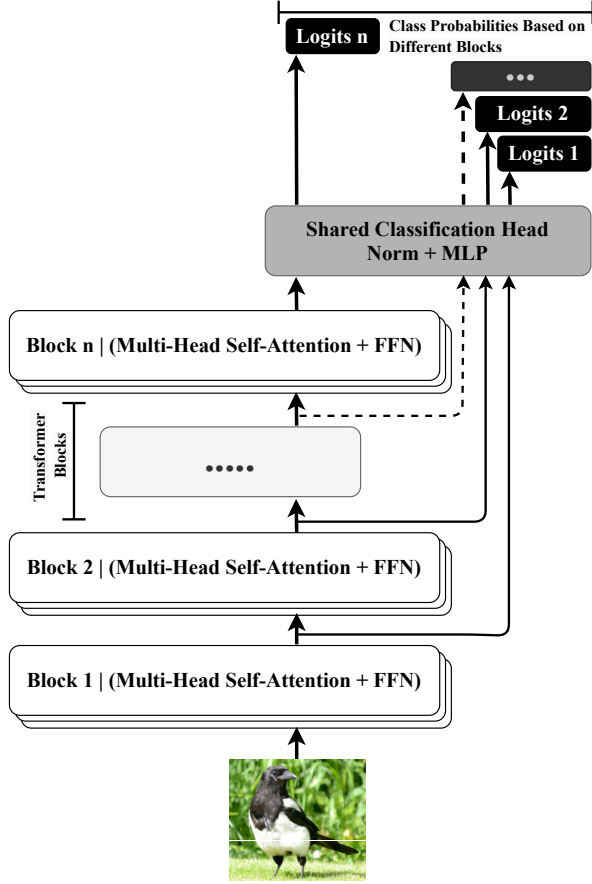


Fig. 2: Modified transformer architecture to extract discriminative information of a given input from all blocks to enable AdViT adversarial attack. Unlike traditional transformer models where only the final block ( $Block_n$ ) connects to the classification layer, this design maps each intermediate block  $Block_i$  to a shared classification head (Norm + MLP). By obtaining separate logits from each block’s output, we can leverage richer block-level information for more powerful adversarial optimization—without retraining or altering the Transformer blocks themselves.

intermediate block  $f_j$  to the classification head  $g$  (instead of solely the last block  $f_n$ ), we can extract critical *layer-specific* discriminative information for adversarial attack purposes, all without retraining the transformer blocks themselves.

Figure 2 shows the architecture of the modified model for the attack. This architecture is used to extract discriminative information across patches based on the classification layer of each transformer block. This is done by passing patches in each transformer block and updating them to retain important information and discarding irrelevant information. The final prediction for the updated patches is then used to calculate the adversarial loss function. As the network updates the patches in each transformer block, it distills crucial information, which allows the final prediction to retain the most discriminative information. This is a crucial step in our proposed method, as it allows us to compute the loss of input more effectively. The attack generates adversarial samples that consider classification and interpretation models using extracted information.

We note that while our current approach does not alter the existing weights, we expect that fine-tuning or retraining the model could further enhance attack transferability or interpretation quality.

**Attack Formulation.** The traditional attack formulation to generate perturbation is as follows:

$$\hat{x}^{(i+1)} = \Pi_{\mathcal{B}_\varepsilon(x)} \left( \hat{x}^{(i)} - \alpha \cdot \text{sign}(\nabla_{\hat{x}} \ell_{adv}(\hat{x}^{(i)})) \right), \quad (1)$$

where  $\Pi$ ,  $\alpha$ , and  $\mathcal{B}_\varepsilon(x)$  represent the projection operator, learning rate, and the norm ball respectively. The symbol  $\varepsilon$  in the attack formulation represents the size of the allowable perturbation and  $\ell_{adv}$  is the overall adversarial loss. Using Eq. 1 to generate adversarial samples based only on classification loss is not effective, as they can be easily detected by interpretation models (see Figure 1). In such cases, it is necessary to optimize the attack to generate perturbations that can mislead the interpretation models. To produce stealthy adversarial samples that fool the classification model and interpreter, we minimize the overall adversarial loss in terms of both classification loss  $\ell_{cls}$  and interpretation loss  $\ell_{int}$ :

$$\ell_{adv} = \min_{\hat{x}} \ell_{cls}(\mathcal{F}(\hat{x})) + \lambda \ell_{int}(\mathcal{G}(\hat{x}; \mathcal{F}), m) \quad (2)$$

Here,  $\ell_{cls}$  is the cross entropy classification loss,  $\ell_{int}$  is the difference of adversarial interpretation maps and benign interpretation maps, and the hyper-parameter  $\lambda$  balances  $\ell_{cls}$  and  $\ell_{int}$ .

We define  $\ell_{cls}$  as:

$$\ell_{cls} = -\frac{1}{n} \sum_{j=1}^n \log \left( \frac{e^{g_c(\mathcal{F}_j(\hat{x}))}}{\sum_{k=1}^{|C|} e^{g_k(\mathcal{F}_j(\hat{x}))}} \right) \quad (3)$$

Here,  $\mathcal{F}_j(\hat{x})$  denotes the output of the  $j$ -th transformer block processed by the adversarial sample  $\hat{x}$ , and  $g_c(\mathcal{F}_j(\hat{x}))$  represents the logit for the true class  $c$ , produced by the classification head based on the  $j$ -th block’s output. The function aims to maximize the misclassification at all levels of the transformer model.

We define  $\ell_{int}$  as:

$$\ell_{int} = \sum_{i=1}^m w_i \cdot (\mathcal{G}(\hat{x}; \mathcal{F})_i - m_i)^2 \quad (4)$$

Here,  $m_i$  and  $\mathcal{G}(\hat{x}; \mathcal{F})_i$  represent the importance scores of the  $i$ -th feature (or patch) in the benign and adversarial interpretation maps, respectively. The weight  $w_i$  could be introduced to prioritize features based on their relevance to the classification decision, although for simplicity, it could initially be set to 1 for all features, which implies equal importance. This formulation computes a weighted sum of squared differences across all features, encouraging the adversarial sample to maintain a similar interpretation to the benign sample.

The primary objective of AdViT, as encapsulated in Eq. 2, is to minimize the combined loss function that integrates the classification and interpretation losses of the adversarial input  $\hat{x}$ . This approach is optimized using the revised loss functions of  $\ell_{cls}$  and  $\ell_{int}$ , as shown in Eq. 3 and Eq. 4. These formulations leverage the inherent structure of transformer blocks to enhance the attack’s effectiveness, making

**Algorithm 1:** AdViT attack in black-box settings

---

**Data:** Source model  $\mathcal{F}$ , interpreter  $\mathcal{G}$ , input  $x$ , original category  $c$ , perturbation threshold  $\epsilon$ , mutation rate  $mr$ , crossover rate  $cr$ , target model  $\mathcal{F}'$ , population size  $n$ , generation  $G$ ,

**Result:** Adversarial sample  $\hat{x}$

```

1  $x' = \text{our\_attack}(\mathcal{F}, \mathcal{G}, x, n)$ 
2  $pop = \text{init\_population}(x, x', \epsilon)$ 
3 for  $g \leftarrow 1$  to  $G$  do
4    $p_1, p_2 = \text{random\_select}(pop)$ 
5    $v_1, v_2 = \text{get\_fitness}(\mathcal{F}', x, p_1, p_2)$ 
6    $loser, winner = \text{sort\_by\_fitness}(p_1, p_2, v_1, v_2)$ 
7    $child = \text{crossover}(cr, loser, winner)$ 
8    $child = \text{mutation}(mr, child)$ 
9   if  $f(child) \neq c$  then
10    |  $\text{return } child$ 
11  end
12   $pop = \text{update\_population}(pop, child)$ 
13 end

```

---

it applicable to a wide range of transformer-based models. By emphasizing the need for misclassification through  $\ell_{cls}$  while simultaneously ensuring stealth through  $\ell_{int}$ , our method achieves a sophisticated balance, with the aim of minimizing the combined loss.

### B. AdViT Optimization for Black-box Settings

Investigating the applicability of AdViT in a black-box scenario, we employ a modified mutation-based MGA [16] to generate adversarial examples against black-box models. Using adversarial samples generated against a white-box model  $\mathcal{F}$  as the initial population, the MGA evolves the population to generate an adversarial sample that can fool the black-box model  $\mathcal{F}'$  and its interpreter  $\mathcal{G}'$ . AdViT in the black-box scenario is described in **Algorithm 1**. The attack consists of genetic algorithm operators: *initialization* (line 1-2), *selection* (line 4-6), *crossover* (line 7), *mutation* (line 8), and *population update* (line 12).

**Initialization:** We generate adversarial samples by AdViT in white-box settings and provide them as the initial population for MGA (*i.e.*,  $\Psi : \{\psi_1, \psi_2, \dots, \psi_n\}$ , where  $n$  is the size of the population). We selected five as the population size, which provides the best trade-off between attack effectiveness and time complexity, based on our experiments.

**Fitness function:** The function evaluates the quality of the samples in the population and helps to improve their evolution toward the optimal population. In our attack, we evaluate each individual in the population by applying a loss function (*i.e.*, relative cross entropy) as the fitness function, which is based on the classification confidence and perturbation size [13]. Loss values reflect the fitness scores of the samples in the population where a higher fitness score is desired to achieve the attack. If a sample from the initial population is successful, the attack ends as the criterion is met (*e.g.*, when the transferability of white-box attacks is high).

TABLE I: Parameter configuration for the attack using perturbation generation (PGD) and genetic algorithm.

Algorithm	Parameter	Values
Perturbation generation	# iterations	20
	$\ell_{adv}$ coefficient ( $\lambda$ )	10
	max. search step size $\alpha_{max}$	0.08
	Perturbation threshold ( $\epsilon$ )	0.031
Genetic Algorithm	Mutation rate ( $mr$ )	1e-4
	Crossover rate ( $cr$ )	0.7
	Population size ( $n$ )	5

**Selection:** Samples with high fitness scores have a higher chance of passing along their features to the next generation. [9]. To maintain diversity and high interpretability (passed from the initial generation), we randomly select two samples from a population, one (winner) with a high fitness score and another (loser) with a relatively lower score, to pass on to the crossover phase.

**Crossover:** The new offspring (adversarial sample) is generated by transferring the genetic data of the winner  $\psi_{winner}$  and the loser  $\psi_{loser}$  with the predefined crossover rate  $cr$ :  $\psi_{child} = \psi_{winner} * S_{cr} + \psi_{loser} * (1 - S_{cr})$ , where  $S_{cr}$  is a mask matrix with the values of 1 and 0.  $S_{cr} = 1$  *rand*(0,1) <  $cr$  *otherwise* 0, where *rand*(0,1) generates uniformly distributed numbers between 0 and 1. In the experiment, we use a crossover rate of 0.7.

**Mutation:** The process further diversifies the population through another binary mask:  $\psi_{child} = -\psi_{child} * S_{mr} + \psi_{child} * (1 - S_{mr})$ , where  $S_{mr}$  is a mask matrix based on the mutation rate  $mr$ .

We use a mutation rate of 1e-4.

**Population update:** For continuous evolution, the population is updated by keeping the winners and replacing the losers with the new generation (*i.e.*, the mutated offspring).

The proposed modified genetic algorithm differs from the traditional ones by factoring in both interpretability and output classification when creating offspring. Specifically, we found that traditional GA-generated adversarial examples often fail to fool interpreter models, as they only consider children with high fitness scores (*i.e.*, based on classification output), resulting in a lack of diversity among future generations. Our proposed approach uses a strategy that considers children with low fitness scores (*i.e.*, losers) to generate adversarial examples to promote a higher degree of diversity and better control over the interpretation. This allows for a more effective attack on interpretability in addition to classification.

## IV. EXPERIMENTS

We evaluate AdViT attack on different vision transformer models and interpreters. Our analysis aims to answer the following questions: ① *How effective is it to attack vision transformer models and their coupled interpreters?* ② *Are the adversarial examples transferable across various vision transformer models?* ③ *Is it practical to attack the vision transformer models in black-box settings?* ④ *Is it possible to attack real-world vision transformer models?* For the reproducibility of our experiments, our code, data, and models are available at (<https://github.com/InfoLab-SKKU/AdViT>). For



comparison, existing attacks were implemented under our experimental environment and evaluated on the same dataset.

### A. Experimental Settings

**Datasets.** For our experiments, we use 1,000 images from the validation set of ImageNet that are classified correctly with a confidence higher than 70% by the target ViT model.

**ViT Models.** In white-box settings, we target DeiT-B, DeiT-S, DeiT-T [32], Swin-B, Swin-L [21], T2T-ViT-7, T2T-ViT-10 [38], ViT-B and ViT-L [14] models. In black-box settings, we use the same white-box models as surrogate models to attack ViT family models (ViT-B and ViT-L) [14]. In the realistic black-box scenario, we also demonstrate the effectiveness of AdViT against real-world APIs of four ViT models: ViT-B by Google [14], SWIN-T by Microsoft [21], MIT-B3 by Nvidia [37], and Vision-perceiver-learned by DeepMind [19].

**Interpreters.** We employ two interpreters: Transformer Interpreter [12] and IA-RED<sup>2</sup> [26]. Those interpreters utilize different characteristics of the model to generate interpretations.

**Metrics.** The evaluation metrics are divided into classification and interpretation metrics. Classification metrics include attack success rate and misclassification confidence (*i.e.*, adversarial confidence).

Metrics used for interpreters are qualitative comparison and IoU score.

In addition, we adopt another metric (*noise rate*) to measure the perturbation size. The description of each metric is as follows.

- **Attack success rate:** It calculates the ratio of successful attack cases to the total attack cases.
- **Misclassification confidence:** We measure the probability (confidence score) of an adversarial sample assigned by the target model. We calculate the average probability of adversarial samples being successfully misclassified.
- **Qualitative comparison:** This method is used to verify whether the interpretation results are perceptually similar. Every interpretation map is manually checked to see if it is identical to its benign interpretation map or if the interpretation is reliable.
- **IoU score** (Intersection-over-Union): This metric is used to quantify the similarity between two arbitrary shapes. It encodes the shape properties of interpretation maps, *e.g.*, *height*, *width*, and *location* into region properties and calculates the intersection areas between the predictions and the ground truths. It is widely employed to evaluate object detection, segmentation, and tracking:

$$\text{IoU}(m) = |O(m) \cap O(m_o)| / |O(m) \cup O(m_o)|,$$

where,  $m$  is the attribution map of samples when the universal perturbation is added and  $m_o$  is the attribution map of samples without any perturbation. In our case, we compare an adversarial interpretation map with the benign interpretation map based on (shapes, positions, and areas), for which the metric can be applied.

- **Noise rate:** Perturbation amount is calculated using the structural similarity index (SSIM) [35]. SSIM measures the

similarity score, and we find the non-similarity portion using that score (*i.e.*,  $\text{noise\_rate} = 1 - \text{SSIM}$ ).

The table containing the parameter values for the experimental settings is presented in **Table I**.

### B. AdViT: White-box Settings

This section explores the effectiveness of AdViT against DeiT-B, DeiT-S, DeiT-T, Swin-B, Swin-L, T2T-ViT-7, T2T-ViT-10, ViT-B, and ViT-L, using two popular ViT interpreters: the transformer interpreter and IA-RED<sup>2</sup>. In addition to comparing AdViT with existing interpreter-based adversarial methods (ADV<sup>2</sup> [42] and AdvEdge [1]), we also consider non-interpreter-based baselines, PGD [22] and ATViT [24].

**Table II** shows that our proposed approach successfully misleads all tested models with a 100% success rate and achieved high misclassification confidence scores. For attacks using the transformer interpreter, ADV<sup>2</sup> and AdvEdge obtain average misclassification confidences of approximately 0.62 and 0.63 on the DeiT variants, 0.59 and 0.60 on Swin models, 0.57 and 0.60 on T2T-ViT variants, and 0.61 and 0.59 on ViT models. In contrast, AdViT consistently outperforms them, reaching 0.72, 0.97, 0.93, and 0.97 on DeiT, Swin, T2T-ViT, and ViT models respectively. Under the IA-RED<sup>2</sup> interpreter, ADV<sup>2</sup> and AdvEdge struggle to exceed a misclassification confidence of about 0.43, while AdViT achieves a minimum of 0.36 and generally around 0.60 — 0.90.

The results of PGD and ATViT attacks are provided outside the interpreter-based comparison columns on **Table II** as they do not use interpreters to craft their perturbations. Including these non-interpreter-based methods as baselines clarifies that using interpretative information during adversarial generation leads to additional improvements. By extending and refining the fundamental concepts behind PGD and ATViT, AdViT surpasses these non-interpreter-based methods and significantly advances beyond other interpreter-guided attacks.

Moreover, our findings suggest that the IA-RED<sup>2</sup> interpreter is more robust than the transformer interpreter, as evidenced by the generally lower adversarial confidence scores.

To further analyze the impact of AdViT on model interpretation, we visualize the interpretation maps of benign and adversarial samples in **Figure 3**. These maps appear nearly identical, indicating that AdViT preserves critical interpretative features while fooling the classifiers. The IoU test results presented in **Figure 4** confirm this observation: AdViT achieves IoU scores exceeding 0.8 across all tested transformer-based models for both interpreters.

In contrast, PGD, ATViT, ADV<sup>2</sup>, and AdvEdge produce significantly lower IoU values, indicating less accurate interpretation maps.

### C. AdViT: Black-box Settings

Adopting a transferability-based approach (*e.g.*, [29], [5]), we improve transferability using the MGA algorithm. We investigate the performance using two settings: ① typical transferability and ② improved transferability via MGA.

**Attack Transferability.** We used DeiT, Swin, T2T-ViT, and ViT-based models as a source to generate adversarial samples

TABLE II: White-box scenario: Comparison of misclassification confidence against various ViT-based models and two interpreters (Transformer Interpreter and IA-RED<sup>2</sup>). The gray-shaded columns (PGD and ATViT) represent non-interpreter baseline attacks, while other methods (ADV<sup>2</sup>, AdvEdge, and AdvViT) use interpretative information in crafting perturbations.

			Transformer Interpreter			IA-RED <sup>2</sup>		
	PGD	ATViT	ADV <sup>2</sup>	AdvEdge	Ours (AdvViT)	ADV <sup>2</sup>	AdvEdge	Ours (AdvViT)
DeiT-B	0.59±0.20	0.49±0.21	0.62±0.22	0.64±0.22	<b>0.78±0.18</b>	0.19±0.24	0.21±0.24	<b>0.45±0.20</b>
DeiT-S	0.58±0.19	0.51±0.23	0.63±0.22	0.64±0.22	<b>0.65±0.19</b>	0.17±0.22	0.17±0.23	<b>0.36±0.20</b>
DeiT-T	0.56±0.19	0.50±0.22	0.61±0.22	0.62±0.22	<b>0.72±0.18</b>	0.18±0.21	0.20±0.21	<b>0.43±0.19</b>
Swin-B	0.60±0.21	0.53±0.23	0.60±0.20	0.60±0.20	<b>0.95±0.10</b>	0.25±0.20	0.25±0.20	<b>0.59±0.12</b>
Swin-L	0.59±0.22	0.49±0.24	0.58±0.21	0.60±0.21	<b>0.98±0.08</b>	0.29±0.21	0.31±0.21	<b>0.62±0.10</b>
T2T-ViT-7	0.58±0.15	0.55±0.19	0.59±0.17	0.61±0.18	<b>0.92±0.11</b>	0.26±0.18	0.26±0.18	<b>0.60±0.14</b>
T2T-ViT-10	0.56±0.18	0.54±0.20	0.55±0.19	0.59±0.19	<b>0.94±0.16</b>	0.20±0.19	0.23±0.19	<b>0.59±0.17</b>
ViT-B	0.55±0.19	0.56±0.15	0.60±0.13	0.60±0.13	<b>0.97±0.04</b>	0.38±0.14	0.42±0.14	<b>0.95±0.13</b>
ViT-L	0.55±0.20	0.54±0.16	0.61±0.12	0.58±0.12	<b>0.96±0.05</b>	0.40±0.13	0.43±0.13	<b>0.93±0.05</b>

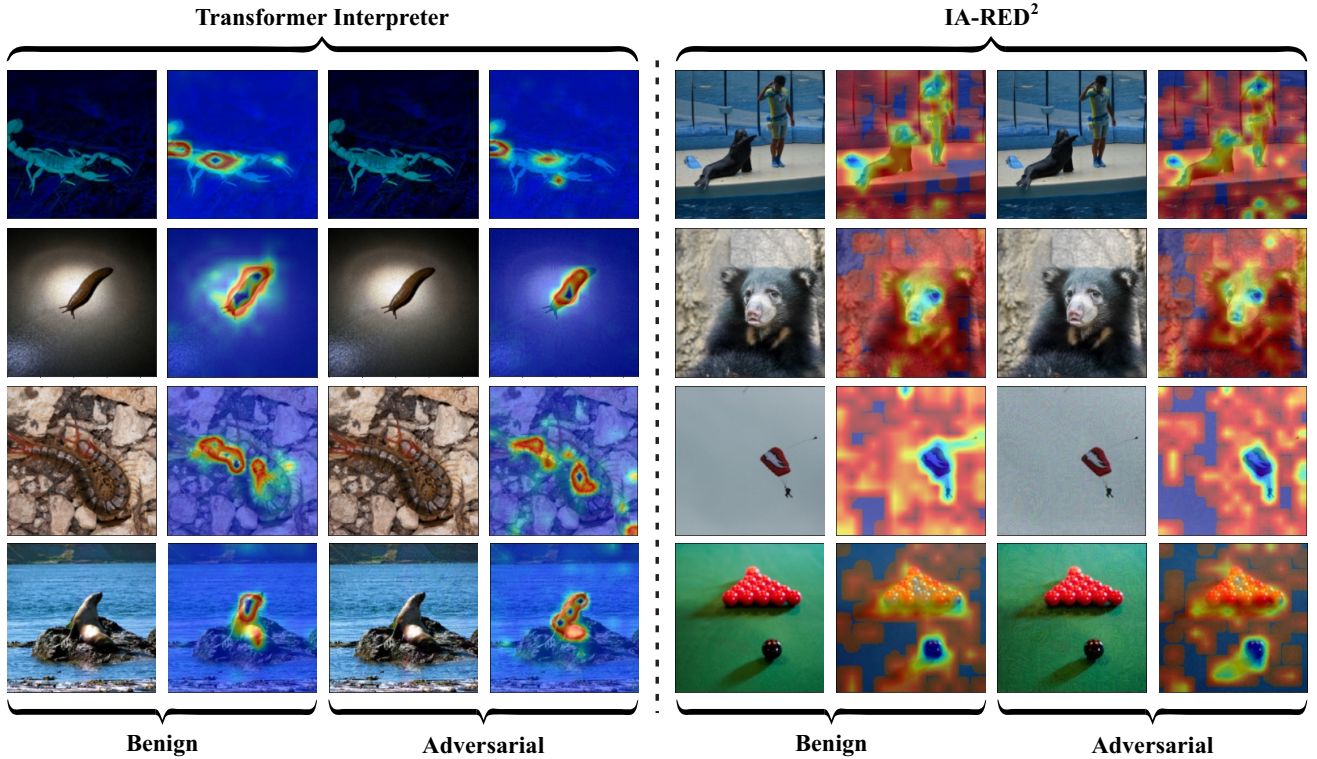


Fig. 3: Attribution maps of benign and adversarial samples generated by AdvViT using two interpreters.

against each other. Table III shows the attack success rate and misclassification confidence. When we used the transformer interpreter in our study, we found different results for each transformer family. For the DeiT family, the highest success rate of the attack was 0.78 (average 0.65 with a variation of  $\pm 0.18$ ) and the lowest was 0.16 (average 0.41 with a variation of  $\pm 0.18$ ). For the Swin family, the highest was 0.70 (average 0.71 with a variation of  $\pm 0.19$ ) and the lowest was 0.20 (average 0.42 with a variation of  $\pm 0.15$ ). In the T2T-ViT family, the highest rate was 0.84 (average 0.54 with a variation of  $\pm 0.15$ ) and the lowest was 0.15 (average 0.54 with a variation of  $\pm 0.19$ ). For the ViT family, the highest was 0.86 (average 0.84 with a variation of  $\pm 0.10$ ) and the lowest was 0.24 (average 0.45 with a variation of  $\pm 0.13$ ). When we used the IA-RED interpreter, the results were different. For the DeiT family, the highest success rate was 0.78 (average 0.49 with a variation of  $\pm 0.19$ ) and the lowest was 0.16 (average 0.33 with

a variation of  $\pm 0.17$ ). For the Swin family, the highest was 0.76 (average 0.47 with a variation of  $\pm 0.19$ ) and the lowest was 0.22 (average 0.31 with a variation of  $\pm 0.17$ ). In the T2T-ViT family, the highest rate was 0.81 (average 0.33 with a variation of  $\pm 0.14$ ) and the lowest was 0.16 (average 0.47 with a variation of  $\pm 0.19$ ). For the ViT family, the highest was 0.80 (average 0.75 with a variation of  $\pm 0.11$ ) and the lowest was 0.26 (average 0.43 with a variation of  $\pm 0.16$ ).

Furthermore, we investigate the attack transferability against model interpretation using the IoU test between benign and adversarial attribution maps. Figure 5 shows the IoU score of the attack on transformer models with two interpreters. As displayed, the performance is significantly high in both interpreters over 0.80.

**Improving Transferability via MGA.** Table IV shows that AdvViT significantly outperforms the existing Square attack [6] in the black-box scenario. AdvViT achieves a 100% success rate

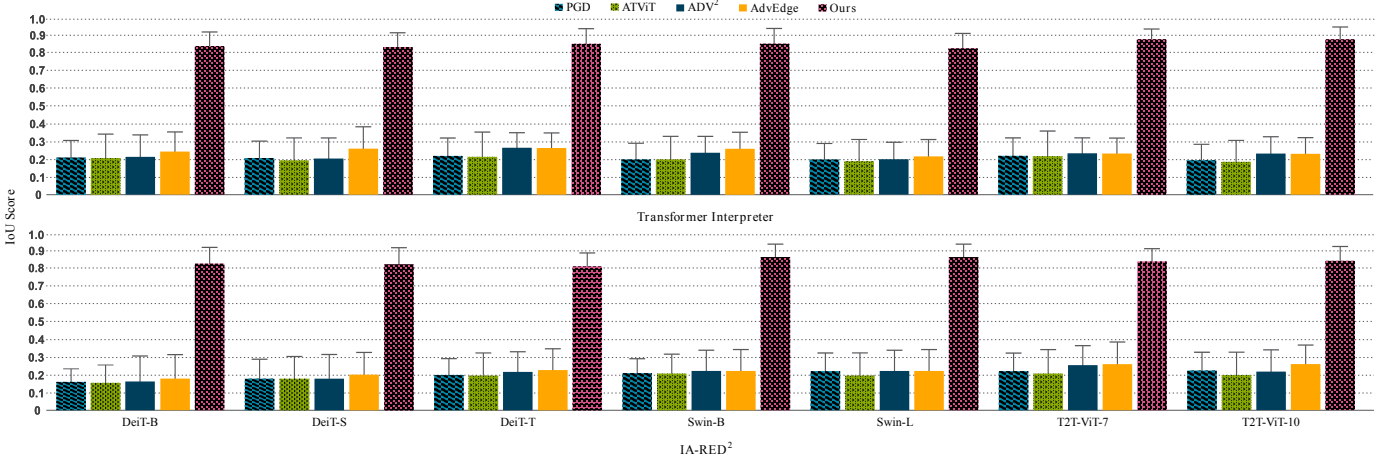


Fig. 4: White-box scenario: IoU scores of adversarial interpretation maps generated by AdViT and existing attacks.

TABLE III: Attack transferability of transformer models to generate adversarial samples against each other. The results are reported as attack success rate (misclassification confidence  $\pm$  standard deviation).

	Models	DeiT-B	DeiT-S	DeiT-T	Swin-B	Swin-L	T2T-ViT-7	T2T-ViT-10	ViT-B	ViT-L
Transformer Interpreter	DeiT-B		0.71 (0.62 $\pm$ 0.19)	0.78 (0.65 $\pm$ 0.18)	0.31 (0.53 $\pm$ 0.19)	0.20 (0.51 $\pm$ 0.18)	0.40 (0.55 $\pm$ 0.13)	0.41 (0.58 $\pm$ 0.16)	0.40 (0.94 $\pm$ 0.18)	0.30 (0.94 $\pm$ 0.19)
	DeiT-S	0.64 (0.56 $\pm$ 0.19)		0.70 (0.59 $\pm$ 0.18)	0.28 (0.48 $\pm$ 0.19)	0.18 (0.46 $\pm$ 0.18)	0.33 (0.50 $\pm$ 0.13)	0.36 (0.52 $\pm$ 0.16)	0.42 (0.94 $\pm$ 0.18)	0.32 (0.95 $\pm$ 0.17)
	DeiT-T	0.62 (0.52 $\pm$ 0.18)	0.57 (0.49 $\pm$ 0.19)		0.24 (0.42 $\pm$ 0.19)	0.16 (0.41 $\pm$ 0.18)	0.32 (0.44 $\pm$ 0.13)	0.33 (0.46 $\pm$ 0.16)	0.39 (0.93 $\pm$ 0.18)	0.28 (0.95 $\pm$ 0.19)
	Swin-B	0.25 (0.45 $\pm$ 0.14)	0.26 (0.39 $\pm$ 0.10)	0.28 (0.51 $\pm$ 0.14)		0.57 (0.61 $\pm$ 0.18)	0.37 (0.35 $\pm$ 0.12)	0.38 (0.38 $\pm$ 0.14)	0.25 (0.55 $\pm$ 0.18)	0.23 (0.51 $\pm$ 0.18)
	Swin-L	0.22 (0.46 $\pm$ 0.15)	0.20 (0.42 $\pm$ 0.15)	0.25 (0.44 $\pm$ 0.16)	0.70 (0.71 $\pm$ 0.19)		0.31 (0.30 $\pm$ 0.12)	0.27 (0.35 $\pm$ 0.15)	0.21 (0.55 $\pm$ 0.20)	0.20 (0.56 $\pm$ 0.19)
	T2T-ViT-7	0.19 (0.51 $\pm$ 0.18)	0.20 (0.44 $\pm$ 0.16)	0.22 (0.46 $\pm$ 0.16)	0.19 (0.55 $\pm$ 0.19)	0.20 (0.50 $\pm$ 0.11)		0.77 (0.45 $\pm$ 0.17)	0.15 (0.54 $\pm$ 0.19)	0.17 (0.50 $\pm$ 0.18)
	T2T-ViT-10	0.20 (0.52 $\pm$ 0.16)	0.23 (0.48 $\pm$ 0.18)	0.23 (0.54 $\pm$ 0.18)	0.26 (0.57 $\pm$ 0.20)	0.25 (0.55 $\pm$ 0.18)	0.84 (0.54 $\pm$ 0.15)		0.21 (0.57 $\pm$ 0.18)	0.20 (0.51 $\pm$ 0.17)
	ViT-B	0.33 (0.58 $\pm$ 0.18)	0.50 (0.62 $\pm$ 0.20)	0.57 (0.61 $\pm$ 0.19)	0.25 (0.49 $\pm$ 0.18)	0.21 (0.45 $\pm$ 0.13)	0.29 (0.36 $\pm$ 0.16)	0.36 (0.38 $\pm$ 0.16)		0.81 (0.83 $\pm$ 0.10)
	ViT-L	0.36 (0.60 $\pm$ 0.19)	0.55 (0.70 $\pm$ 0.20)	0.58 (0.68 $\pm$ 0.19)	0.30 (0.47 $\pm$ 0.20)	0.29 (0.48 $\pm$ 0.19)	0.35 (0.41 $\pm$ 0.15)	0.31 (0.44 $\pm$ 0.15)	0.86 (0.84 $\pm$ 0.10)	
IA-RED <sup>2</sup>	DeiT-B		0.73 (0.44 $\pm$ 0.20)	0.78 (0.49 $\pm$ 0.19)	0.32 (0.53 $\pm$ 0.20)	0.20 (0.41 $\pm$ 0.17)	0.36 (0.34 $\pm$ 0.20)	0.36 (0.31 $\pm$ 0.16)	0.44 (0.61 $\pm$ 0.20)	0.42 (0.62 $\pm$ 0.20)
	DeiT-S	0.65 (0.39 $\pm$ 0.20)		0.70 (0.44 $\pm$ 0.19)	0.29 (0.48 $\pm$ 0.2)	0.18 (0.37 $\pm$ 0.17)	0.32 (0.31 $\pm$ 0.2)	0.32 (0.28 $\pm$ 0.16)	0.39 (0.61 $\pm$ 0.16)	0.34 (0.62 $\pm$ 0.19)
	DeiT-T	0.62 (0.39 $\pm$ 0.2)	0.58 (0.35 $\pm$ 0.2)		0.26 (0.42 $\pm$ 0.2)	0.16 (0.33 $\pm$ 0.17)	0.28 (0.27 $\pm$ 0.2)	0.29 (0.25 $\pm$ 0.16)	0.33 (0.64 $\pm$ 0.16)	0.30 (0.63 $\pm$ 0.20)
	Swin-B	0.27 (0.41 $\pm$ 0.18)	0.27 (0.39 $\pm$ 0.16)	0.29 (0.40 $\pm$ 0.18)		0.71 (0.45 $\pm$ 0.18)	0.30 (0.33 $\pm$ 0.15)	0.31 (0.30 $\pm$ 0.16)	0.26 (0.42 $\pm$ 0.18)	0.24 (0.38 $\pm$ 0.19)
	Swin-L	0.23 (0.36 $\pm$ 0.18)	0.22 (0.34 $\pm$ 0.19)	0.23 (0.38 $\pm$ 0.18)	0.76 (0.47 $\pm$ 0.19)		0.30 (0.33 $\pm$ 0.17)	0.26 (0.30 $\pm$ 0.16)	0.23 (0.37 $\pm$ 0.19)	0.22 (0.31 $\pm$ 0.17)
	T2T-ViT-7	0.25 (0.54 $\pm$ 0.19)	0.35 (0.47 $\pm$ 0.15)	0.38 (0.45 $\pm$ 0.15)	0.33 (0.51 $\pm$ 0.19)	0.26 (0.50 $\pm$ 0.16)		0.78 (0.41 $\pm$ 0.17)	0.28 (0.51 $\pm$ 0.19)	0.25 (0.53 $\pm$ 0.19)
	T2T-ViT-10	0.20 (0.42 $\pm$ 0.18)	0.23 (0.41 $\pm$ 0.17)	0.25 (0.44 $\pm$ 0.19)	0.23 (0.53 $\pm$ 0.20)	0.16 (0.50 $\pm$ 0.17)	0.81 (0.33 $\pm$ 0.14)		0.16 (0.53 $\pm$ 0.19)	0.16 (0.47 $\pm$ 0.19)
	ViT-B	0.29 (0.47 $\pm$ 0.19)	0.51 (0.50 $\pm$ 0.20)	0.54 (0.47 $\pm$ 0.20)	0.26 (0.47 $\pm$ 0.19)	0.19 (0.46 $\pm$ 0.12)	0.33 (0.39 $\pm$ 0.18)	0.28 (0.47 $\pm$ 0.19)		0.78 (0.76 $\pm$ 0.12)
	ViT-L	0.32 (0.46 $\pm$ 0.17)	0.48 (0.46 $\pm$ 0.19)	0.49 (0.49 $\pm$ 0.19)	0.26 (0.43 $\pm$ 0.16)	0.23 (0.37 $\pm$ 0.19)	0.31 (0.40 $\pm$ 0.20)	0.30 (0.38 $\pm$ 0.17)	0.80 (0.75 $\pm$ 0.11)	

when transferring adversarial samples from DeiT-B to ViT-B under both interpreters, compared to 87% and 89% for the Square attack. AdViT is also more query-efficient, requiring as few as 152 queries for a 100% success rate on ViT-B, while the Square attack needs 405 queries. Misclassification confidence for AdViT range from 0.64 to 0.74 with the Transformer Interpreter and 0.50 to 0.74 with IA-RED<sup>2</sup>, comparable to Square’s 0.78 but achieved with higher success and efficiency.

Notably, ViT-L is the most robust model against AdViT, showing slightly lower success rates and requiring more queries. These results demonstrate AdViT’s effectiveness and efficiency over the Square attack.

We evaluate the similarity between adversarial and benign interpretation maps by calculating the IoU score. Figure 6

shows the performance of the proposed AdViT attack using the MGA algorithm for transferability. Despite the additional noise introduced by the generative algorithm, adversarial interpretation maps remain nearly indistinguishable from their benign counterparts, achieving an IoU of approximately 0.80 in both interpreters. In comparison, the Square attack produces significantly lower IoU scores, around 0.40.

#### D. AdViT: Real-world Black-box Scenario

In this experiment, we explore the performance of the proposed attack in real-world scenarios against four different models: ViT-B, SWIN-T, MIT-B, and VISION-P. We conduct the experiment in two transferability settings (see [Subsection IV-C](#)). We implement AdViT using the DeiT-B model as



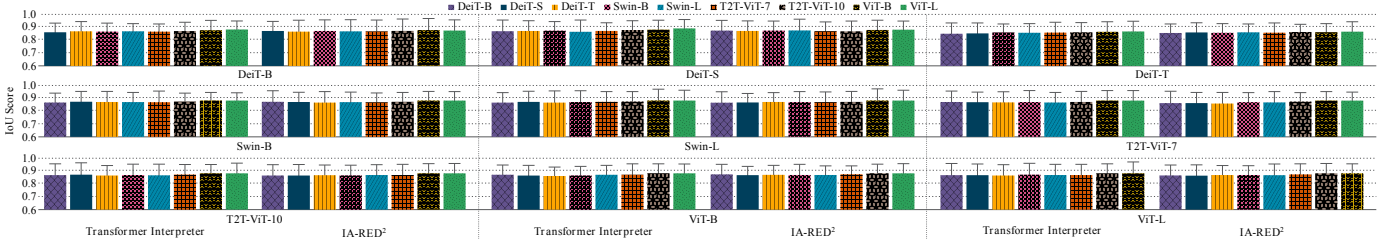


Fig. 5: Black-box scenario: IoU scores of adversarial interpretation maps generated by AdViT using typical transferability and the existing attack.

TABLE IV: Black-box scenario: Comparison of MGA-based AdViT and Square attack [6] in terms of success rate, queries, and confidence. Results for the Square attack are duplicated across both interpreters, as it does not rely on interpreters.

Attack	Source Model	Target Model	Transformer Interpreter			IA-RED <sup>2</sup>		
			Success Rate	Avg. Queries	Misclassification Confidence	Success Rate	Avg. Queries	Misclassification Confidence
AdViT	DeiT-B	ViT-B	1.00	150	0.76±0.20	1.00	161	0.54±0.20
	DeiT-S		0.90	155	0.72±0.20	0.90	158	0.51±0.20
	DeiT-T		0.90	180	0.68±0.20	0.89	179	0.51±0.20
	Swin-B		1.00	120	0.70±0.14	0.98	129	0.69±0.15
	Swin-L		0.96	128	0.71±0.14	0.92	137	0.67±0.15
	T2T-ViT-7		0.94	161	0.69±0.14	0.93	173	0.70±0.16
	T2T-ViT-10		0.94	172	0.66±0.19	0.95	184	0.68±0.19
	DeiT-B	ViT-L	0.95	162	0.76±0.20	0.97	165	0.57±0.20
	DeiT-S		0.87	177	0.70±0.20	0.92	185	0.55±0.20
	DeiT-T		0.82	188	0.69±0.20	0.86	189	0.50±0.20
	Swin-B		0.99	134	0.66±0.19	0.96	154	0.67±0.19
	Swin-L		0.93	149	0.67±0.19	0.94	156	0.67±0.19
	T2T-ViT-7		0.90	189	0.70±0.18	0.91	183	0.74±0.19
	T2T-ViT-10		0.91	195	0.64±0.19	0.91	197	0.65±0.19
Square		ViT-B	0.87	405	0.81±0.16	0.87	405	0.81±0.16
		ViT-L	0.89	420	0.78±0.14	0.89	420	0.78±0.14

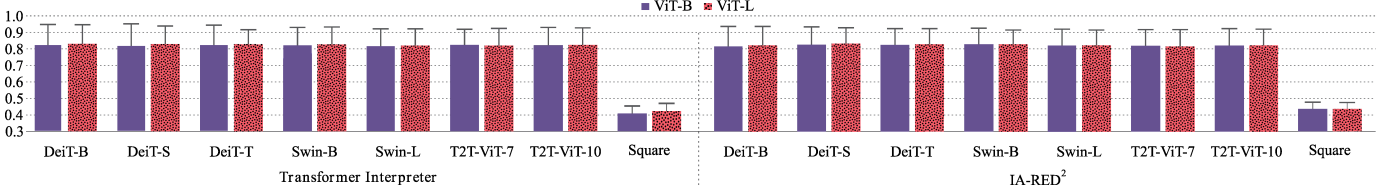


Fig. 6: Black-box scenario: IoU scores of adversarial interpretation maps generated by AdViT using transferability via MGA and Square attack.

TABLE V: Real-world scenario: attack success rate and misclassification confidence of AdViT using typical transferability using DeiT-B as source model.

	Transformer Interpreter			
	ViT-B	SWIN-T	MIT-B	VISION-P
Success Rate	0.37	0.47	0.42	0.22
Misclassification Confidence	0.51	0.37	0.30	0.27

TABLE VI: Real-world scenario: attack success rate, misclassification confidence, and average queries of AdViT using transferability via MGA and DeiT-B as source model.

	Transformer Interpreter			
	ViT-B	SWIN-T	MIT-B	VISION-P
Success Rate	0.85	1.00	0.91	0.73
Misclassification Confidence	0.30	0.26	0.22	0.24
Avg. Queries	205	192	199	218

the source model and transformer interpreter. Table V shows the attack success rate and misclassification confidence using a typical transferability setting.

The result shows that the SWIN-T model is the most susceptible, with an attack success rate of 0.47, while VISION-P is more robust, with the least attack success rate of 0.22 and a misclassification score of 0.27. The ViT-B model shows a high misclassification score of 0.51.

The results show a significant increase in attack success rate when using MGA to enhance transferability, as shown in Table VI.

For example, the attack success rate increases from 0.22 to 0.73 for the VISION-P model. The results show that the average queries are higher for the VISION-P model than for other models.

#### E. AdViT: Attacking Defensive ViT Models

This section evaluates how MGA-based AdViT performs when defense techniques are applied in black-box settings. This experiment explores the effectiveness of attack against three well-known pre-processing strategies(*i.e.*, R&P, bit-depth

TABLE VII: Success rate and average queries of the proposed attack against four defense techniques using different classifiers and interpreters testing on 500 images of ImageNet dataset. The attack is based on black-box settings with MGA transferability.

Interpreter	Source Model	Target Model	R&P		Bit-Depth Reduction		Median Smoothing		Adversarial Training	
			Success Rate	Avg. Queries	Success Rate	Avg. Queries	Success Rate	Avg. Queries	Success Rate	Avg. Queries
Transformer Interpreter	DeiT-B	ViT-B	0.95	139	0.91	128	0.93	121	0.97	146
		ViT-L	0.90	160	0.92	153	0.91	138	0.94	158
	DeiT-S	ViT-B	0.85	147	0.88	144	0.87	141	0.87	195
		ViT-L	0.84	188	0.86	200	0.84	174	0.85	207
	Swin-B	ViT-B	0.86	169	0.90	153	0.90	134	0.86	156
		ViT-L	0.82	187	0.85	158	0.87	158	0.84	164
	Swin-L	ViT-B	0.89	172	0.89	148	0.89	166	0.89	210
		ViT-L	0.87	207	0.87	204	0.90	195	0.88	211
	T2T-ViT-7	ViT-B	0.85	165	0.87	158	0.88	137	0.87	158
		ViT-L	0.82	185	0.88	154	0.91	157	0.87	167
IA-RED <sup>2</sup>	T2T-ViT-10	ViT-B	0.91	180	0.94	147	0.90	176	0.88	201
		ViT-L	0.92	205	0.92	190	0.85	192	0.85	206
	DeiT-B	ViT-B	0.95	110	0.97	115	0.94	120	0.88	150
		ViT-L	0.91	144	0.94	151	0.92	145	0.87	184
	DeiT-S	ViT-B	0.83	153	0.86	163	0.83	162	0.88	195
		ViT-L	0.81	176	0.84	182	0.80	177	0.85	210
	Swin-B	ViT-B	0.91	120	0.98	124	0.98	122	0.89	149
		ViT-L	0.86	150	0.94	152	0.90	156	0.85	171
	Swin-L	ViT-B	0.86	166	0.89	163	0.84	169	0.83	202
		ViT-L	0.80	188	0.84	186	0.82	189	0.84	210
	T2T-ViT-7	ViT-B	0.89	159	0.89	151	0.83	144	0.89	198
		ViT-L	0.91	139	0.94	158	0.95	154	0.88	206
	T2T-ViT-10	ViT-B	0.88	163	0.96	156	0.95	164	0.90	201
		ViT-L	0.82	179	0.91	162	0.85	175	0.86	203

reduction, median smoothing) and adversarial training defense using 500 images from the ImageNet dataset. For this experiment, we use DeiT, Swin, and T2T-ViT models as source models to generate adversarial samples targeting ViT-B and ViT-L models. We investigate the performance of two interpreters: transformer interpreter and IA-RED<sup>2</sup>.

Although defense techniques are applied, the attack success rate is still high, as shown in Table VII. For example, the success rate is between 0.82 and 0.95 for the transformer interpreter and between 0.80 and 0.95 in the IA-RED<sup>2</sup> interpreter when using R&P defense. Against adversarial training, the results show that in both interpreters, AdvIT achieves a high success rate ranging between 0.83 and 0.97. Another critical evaluation metric is the number of average queries required to attack the target model in the black-box setting. Against all defenses, the average number of queries is between 110 and 210, which is an outstanding result for black-box settings.

In terms of IoU scores, Figure 7 shows that even when a defense technique is applied, our proposed attack still maintains high IoU scores (*i.e.*, 0.80 against all scenarios).

#### F. Interpretation-based Adversarial Detection

The recent work [42] suggests using an ensemble of interpretation models to defend against interpretation-based attacks.

We test the detectability of the attack based on different interpretations. Using multiple interpretations of a single input, we build a multiple-interpreter-based detector that checks whether the input is adversarial or benign.

TABLE VIII: Performance of two types of ensemble detectors composed of two interpreters (*i.e.*, transformer interpreter and IA-RED<sup>2</sup>). The first ensemble detector is based on a 2-channels (*i.e.*, 2 interpretation maps) and the second ensemble detector is based on a 3-channels (*i.e.*, 3 interpretation maps).

Detector Type	Detection Success Rate
2-channel detector	0.75
3-channel detector	0.80

We generated interpretation maps of adversarial samples via two interpreters for our experiment. For example, adversarial samples and adversarial interpretation maps are generated based on the Transformer interpreter, and extra interpretation maps of those adversarial samples are produced using an IA-RED<sup>2</sup> interpreter. We repeated the same process by generating samples based on an IA-RED<sup>2</sup> interpreter and applying a Transformer interpreter as a secondary one. Since the generated attribution maps are based on single-channel, we stacked single-channel attribution maps from two interpreters to convert them into benign and adversarial two-channel data, respectively. 2,000 benign and 2,000 adversarial samples are produced for the experiment.

As the dataset size is small, we adopted the pre-trained CNN model EfficientNet-B7 [31] to extract feature vectors of a given input and a model called gradient boosting classifier as a final layer instead of the fully-connected layer. This approach is

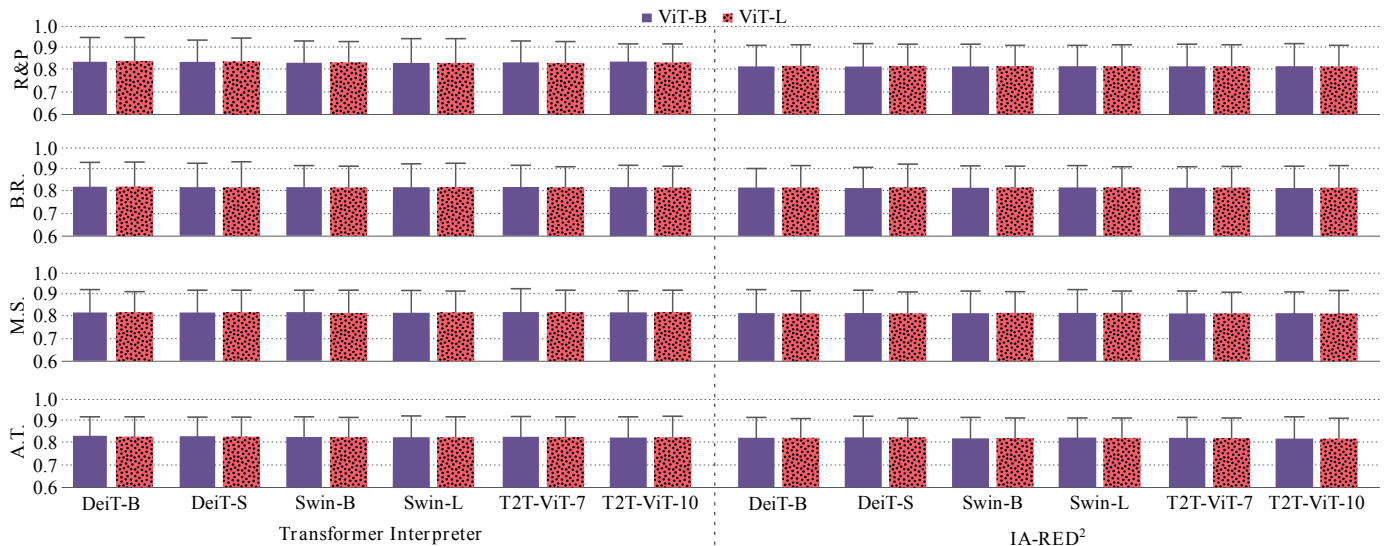


Fig. 7: IoU scores of adversarial interpretation maps generated by the proposed attack when defense techniques are applied. MGA algorithm is used to optimize the attack. A.T., M.S. and B.R. stand for Adversarial Training, Median Smoothing, and Bit-Depth Reduction respectively.

due to the high similarity of benign and adversarial attribution maps and the complexity of the process required to classify the samples. Generally, the EfficientNet models have better accuracy and efficiency than the existing CNNs, with a significant reduction in parameter size and FLOPs. The gradient boosting classifier consists of several weak learning models that form a stronger predictive model. Each attribution map is a one-channel image. To adjust the weights of the model and generate two-channel samples, we replaced the input and output layers of EfficientNet-B7. We used the multiplication of attribution maps extracted from two interpreters as the third channel for the second detector. Table VIII shows the results of the interpretation-based adversarial detector. Even though the dataset is small, the results are promising, which can be seen in the results of the 3-channel detector.

## V. RELATED WORK

**Interpretation-guided White-box Attacks.** Zhang et al. [42] conducted the first systematic security analysis for interpretable deep learning systems (IDLSeS), demonstrating their vulnerability to adversarial manipulation. They presented ADV<sup>2</sup>, a new class of attacks that generate adversarial inputs capable of deceiving DNN models and misleading their interpreters. Following this work, AdvEdge and AdvEdge+ [1], [3] were proposed to optimize the adversarial attack by adding perturbation to the edges in the regions highlighted by the interpretation map, allowing for more stealthy attacks. Their work has been extended by proposing a query-efficient black-box attack [4] that stealthily manipulates both predictions and interpretations of deep models without access to model internals. In another study, Zhang et al. [40] introduced the Interpretation Manipulation Framework (IMF), a data poisoning attack framework that can manipulate the interpretation results of the target inputs as intended by the adversary while preserving the prediction performance. Dombrowski et al. [17] demonstrated that saliency map interpreters (*i.e.*,

LRP, Grad-CAM) could easily be fooled by incorporating the interpretation results directly into the penalty term of the objective function.

**Interpretation-guided Black-box Attacks.** Most existing attacks against IDLSeS rely on white-box settings, which limit their practicality in real-world applications.

Zhan *et al.* [39] introduced a new methodology called Dual Black-box Adversarial Attack (DBAA) that produces adversarial samples to fool the classifier and have comparable interpretations to the benign. They focused on only a single class of interpreters (CAM, Grad-CAM) and CNN-based models. Baniecki and Biecek [10] proposed an algorithm that manipulates SHapley Additive Explanations (SHAP) interpreter based on the perturbation of tabular data. It employs genetic-based data perturbations to control SHAP for a model by minimizing the loss between the manipulated explanation and an arbitrarily selected target. Naseer *et al.* [24] studied improving adversarial transferability in a black-box setting, focusing on ViTs and showing that carefully crafted perturbations can fool models across architectural differences without direct access to their parameters. Our attack strategy builds upon this idea of transferability, leveraging the capacity of robust perturbations to remain effective across various models. proposed SingleADV, a target-specific adversarial attack designed to mislead both predictions and interpretation maps in a class-specific manner. Their method effectively crafts perturbations that suppress visual saliency for the target class while enhancing it for a chosen distractor, exposing vulnerabilities in interpretable deep learning systems. Abdurkhamidov et al. [2] proposed SingleADV, a target-specific adversarial attack designed to mislead both predictions and interpretation maps by crafting a universal perturbation for a class-specific category to be misclassified. Their method effectively crafts perturbations that suppress visual saliency for the target class, *i.e.*, maintaining precise and highly relevant interpretations.

**Transfer-based Attacks.** Aivodji *et al.* [5] examined the



capability of fairwashing attacks by analyzing the fidelity-unfairness trade-off. They demonstrated that fairwashed explanation models generalize beyond the legal group being sued (*i.e.*, beyond the data points being explained), suggesting they can rationalize future unfair decisions made on the basis of black-box models using fairwashed explanation models. Fu *et al.* [36] proposed strategies to improve the transferability of adversarial examples across different vision transformers (ViTs) by considering their patch-based inputs and self-attention mechanisms. Zhang *et al.* [41] proposed the Token Gradient Regularization (TGR) method, which reduces the variance of the back-propagated gradient in each internal block of ViTs in a token-wise manner. TGR utilizes the regularized gradient to generate adversarial samples, offering improved performance compared to state-of-the-art transfer-based attacks when attacking both ViTs and CNNs. The transferability of adversarial examples has also been studied in various transfer-based attack methods [30], [33], [18], [28].

## VI. CONCLUSION

This work examines the security of IDLSes based on vision transformer models. We present AdvIT, an interpretation-guided attack that generates adversarial inputs to mislead target transformer models and deceive their coupled interpreters.

Through comprehensive experiments, we demonstrate the effectiveness of AdvIT against a range of transformer classifiers and interpretation models in both white-box and black-box settings. We show that AdvIT maintains high transferability to target black-box models, especially when employing MGA to optimize the adversarial samples. We also explore the attack's effectiveness against the real-world APIs of four ViT models, ViT-B, SWIN-T, MIT-B, and VISION-P, highlighting the practical implications of our findings. Furthermore, we present the robustness of AdvIT against various defense mechanisms, including random resizing and padding (R&P), bit-depth reduction, median smoothing, and adversarial training. Although AdvIT demonstrates remarkable success against these defenses, we show that implementing an interpretation-based ensemble detector indicates a promising direction to harden the security of ViT-based IDLSes.

As AdvIT is the first attack targeting ViT models coupled with interpretation models, it paves the way for the development of potentially more powerful adversarial attacks. Our work also serves as a catalyst for researchers to create more effective defenses against attacks similar to AdvIT, fostering a more secure and robust environment for the deployment of ViT-based IDLSes in real-world applications.

## ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT)(No. 2021R1A2C1011198), (Institute for Information & communications Technology Planning & Evaluation) (IITP) grant funded by the Korea government (MSIT) under the ICT Creative Consilience Program (IITP-2021-2020-0-01821), AI Platform to Fully Adapt and Reflect Privacy-Policy Changes (No. 2022-0-00688), and Convergence security core talent training business (No.2022-0-01199).

## REFERENCES

- [1] Eldor Abdukhamidov, Mohammed Abuhamad, Firuz Juraev, Eric Chan-Tin, and Tamer AbuHmed. Advedge: Optimizing adversarial perturbations against interpretable deep learning. In *International Conference on Computational Data and Social Networks*, pages 93–105. Springer, 2021. 1, 6, 11
- [2] Eldor Abdukhamidov, Mohammed Abuhamad, George K Thiruvathukal, Hyounghick Kim, and Tamer Abuhmed. Singleadv: single-class target-specific attack against interpretable deep learning systems. *IEEE Transactions on Information Forensics and Security*, 19:5985–5998, 2024. 11
- [3] Eldor Abdukhamidov, Mohammed Abuhamad, Simon S. Woo, Eric Chan-Tin, and Tamer Abuhmed. Hardening interpretable deep learning systems: Investigating adversarial threats and defenses. *IEEE Transactions on Dependable and Secure Computing*, 21(4):3963–3976, 2024. 11
- [4] Eldor Abdukhamidov, Mohammed Abuhamad, Simon S Woo, Eric Chan-Tin, and Tamer Abuhmed. Stealthy query-efficient opaque attack against interpretable deep learning. *IEEE Transactions on Reliability*, 2025. 11
- [5] Ulrich Aïvodji, Hiromi Arai, Sébastien Gambs, and Satoshi Hara. Characterizing the risk of fairwashing. *Advances in Neural Information Processing Systems*, 34:14822–14834, 2021. 6, 11
- [6] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European conference on computer vision*, pages 484–501. Springer, 2020. 7, 9
- [7] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bénéttot, Siham Tabik, Alberto Barbado, Salvador García, Sergio Gil-López, Daniel Molina, Richard Benjamins, et al. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information fusion*, 58:82–115, 2020. 1
- [8] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS one*, 10(7):e0130140, 2015. 3
- [9] Thomas Back. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford university press, 1996. 5
- [10] Hubert Baniecki and Przemyslaw Biecek. Manipulating shap via adversarial data perturbations (student abstract). *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(11):12907–12908, Jun. 2022. 11
- [11] Srinadh Bhojanapalli, Ayan Chakrabarti, Daniel Glasner, Daliang Li, Thomas Unterthiner, and Andreas Veit. Understanding robustness of transformers for image classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10231–10241, 2021. 1
- [12] Hila Chefer, Shir Gur, and Lior Wolf. Transformer interpretability beyond attention visualization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 782–791, 2021. 2, 3, 6
- [13] Jinyin Chen, Mengmeng Su, Shijing Shen, Hui Xiong, and Haibin Zheng. Poba-ga: Perturbation optimized black-box adversarial attacks via genetic algorithm. *Computers & Security*, 85:89–106, 2019. 5
- [14] 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. *ICLR*, 2021. 1, 6
- [15] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 3681–3688, 2019. 1
- [16] Inman Harvey. The microbial genetic algorithm. In *European conference on artificial life*, pages 126–133. Springer, 2009. 5
- [17] Juyeon Heo, Sunghwan Joo, and Taesup Moon. Fooling neural network interpretations via adversarial model manipulation. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 11
- [18] Lifeng Huang, Chengying Gao, and Ning Liu. Erosion attack: Harnessing corruption to improve adversarial examples. *IEEE Transactions on Image Processing*, 32:4828–4841, 2023. 12

- [19] Andrew Jaegle, Sebastian Borgeaud, Jean-Baptiste Alayrac, Carl Doersch, Catalin Ionescu, David Ding, Skanda Koppula, Daniel Zoran, Andrew Brock, Evan Shelhamer, Olivier J. Hénaff, Matthew M. Botvinick, Andrew Zisserman, Oriol Vinyals, and João Carreira. Perceiver IO: A general architecture for structured inputs & outputs. *CoRR*, abs/2107.14795, 2021. 6
- [20] Tianpeng Liu, Jing Li, Jia Wu, Lefei Zhang, Jun Chang, Jun Wan, and Lezhi Lian. Tracking with saliency region transformer. *IEEE Transactions on Image Processing*, 33:285–296, 2024. 1
- [21] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 6
- [22] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 6
- [23] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern recognition*, 65:211–222, 2017. 3
- [24] Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Fahad Khan, and Fatih Porikli. On improving adversarial transferability of vision transformers. In *International Conference on Learning Representations*, 2022. 3, 6, 11
- [25] Maximilian Noppel and Christian Wressneger. Sok: Explainable machine learning in adversarial environments. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 21–21. IEEE Computer Society, 2023. 2
- [26] Bowen Pan, Rameswar Panda, Yifan Jiang, Zhangyang Wang, Rogerio Feris, and Aude Oliva. Ia-red2: Interpretability-aware redundancy reduction for vision transformers. *Advances in Neural Information Processing Systems*, 34:24898–24911, 2021. 2, 3, 6
- [27] Rulin Shao, Zhouxing Shi, Jinfeng Yi, Pin-Yu Chen, and Cho-Jui Hsieh. On the adversarial robustness of vision transformers. *arXiv preprint arXiv:2103.15670*, 2021. 1
- [28] Meng Shen, Changyue Li, Hao Yu, Qi Li, Liehuang Zhu, and Ke Xu. Decision-based query efficient adversarial attack via adaptive boundary learning. *IEEE Transactions on Dependable and Secure Computing*, 21(4):1740–1753, 2024. 12
- [29] David Solans, Battista Biggio, and Carlos Castillo. Poisoning attacks on algorithmic fairness. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 162–177. Springer, 2020. 6
- [30] Xuxiang Sun, Gong Cheng, Hongda Li, Lei Pei, and Junwei Han. On single-model transferable targeted attacks: A closer look at decision-level optimization. *IEEE Transactions on Image Processing*, 32:2972–2984, 2023. 12
- [31] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019. 10
- [32] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021. 6
- [33] Donghua Wang, Wen Yao, Tingsong Jiang, and Xiaoqian Chen. Improving transferability of universal adversarial perturbation with feature disruption. *IEEE Transactions on Image Processing*, 33:722–737, 2024. 12
- [34] Jingyuan Wang, Yufan Wu, Mingxuan Li, Xin Lin, Junjie Wu, and Chao Li. Interpretability is a kind of safety: An interpreter-based ensemble for adversary defense. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 15–24, 2020. 2
- [35] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 6
- [36] Zhipeng Wei, Jingjing Chen, Micah Goldblum, Zuxuan Wu, Tom Goldstein, Yu-Gang Jiang, and Larry S. Davis. Towards transferable adversarial attacks on image and video transformers. *IEEE Transactions on Image Processing*, 32:6346–6358, 2023. 12
- [37] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *CoRR*, abs/2105.15203, 2021. 6
- [38] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis E.H. Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 558–567, October 2021. 6
- [39] Yike Zhan, Baolin Zheng, Qian Wang, Ningping Mou, Binqing Guo, Qi Li, Chao Shen, and Cong Wang. Towards black-box adversarial attacks on interpretable deep learning systems. In *2022 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2022. 11
- [40] Hengtong Zhang, Jing Gao, and Lu Su. Data poisoning attacks against outcome interpretations of predictive models. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 2165–2173, 2021. 11
- [41] Jianping Zhang, Yizhan Huang, Weibin Wu, and Michael R. Lyu. Transferable adversarial attacks on vision transformers with token gradient regularization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16415–16424, June 2023. 12
- [42] Xinyang Zhang, Ningfei Wang, Hua Shen, Shouling Ji, Xiapu Luo, and Ting Wang. Interpretable deep learning under fire. In *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020. 1, 6, 10, 11