Adversarial Robustness in Vision Transformers (ViTs) for Object Recognition

Authors
Xueshi (Cassie) Kang, Suim Park, Hoi Mei (Kelly) Tong
Duke University

Abstract

Adversarial robustness has become a critical focus in neural networks, particularly with the emergence of Vision Transformers (ViTs) as a dominant architecture in computer vision tasks. The adversarial robustness of ViTs (ViT-B-32), traditional Convolutional Neural Networks (CNNs) experimented with ResNet56, and hybrid ensemble models (ViT-L-16 combined with BiT-M-101x3) is investigated against white-box and black-box attack methods. Transferability tests reveal vulnerabilities shared across architectures. ViT exhibits stronger defense against complex attacks, while CNNs display greater robustness to simpler adversarial perturbations. This project provides actionable insights for designing robust neural networks and contributes to the broader understanding of adversarial defense mechanisms.

1 Introduction

In recent years, attention-based architectures such as Vision Transformers (ViTs) have revolutionized computer vision tasks, achieving state-of-the-art performance in image classification. These models often outperform or match the capabilities of traditional Convolutional Neural Networks (CNNs), which have been the dominant approach for decades. However, while CNNs have been extensively studied for their vulnerabilities to adversarial attacks and the corresponding defenses, the robustness of ViTs under such attacks remains under-explored. This gap raises critical questions regarding their applicability in security-sensitive environments.

Adversarial attacks, which involve crafting small but malicious perturbations to input data, can cause neural networks to misclassify inputs with high confidence. Such attacks pose significant risks to real-world applications, including autonomous systems, healthcare diagnostics, and security systems. While CNNs have been extensively evaluated for adversarial robustness, ViTs, given their fundamentally different architectural principles, may exhibit distinct behaviors under similar attack scenarios.

This project seeks to bridge the gap in understanding the adversarial robustness of ViTs compared to CNNs. Specifically, we analyze the performance of ResNet56, ViT-B-32, and ensemble models (ViT-L-16 combined with BiT-M-R101x3) against various white-box and black-box adversarial attack methods. The study includes a comprehensive evaluation of transferability—how adversarial examples generated by one model can fool another—and aims to identify unique vulnerabilities and strengths across architectures.

Through exploring attack scenarios and assessing individual as well as ensemble defenses, this project adds valuable perspectives to the discussion on adversarial robustness in modern neural networks and offers practical guidance for developing resilient models.

2 Related Works

Adversarial robustness in neural networks has been extensively studied in both natural language processing (NLP) and computer vision. In NLP, the Transformer architecture has demonstrated notable robustness to adversarial attacks due to its reliance on self-attention mechanisms. Studies such as those by Mahmood et al. [2] explore how Transformers can maintain higher resilience against small perturbations compared to recurrent architectures like LSTMs.

In the computer vision domain, Vision Transformers (ViTs) have emerged as strong competitors to Convolutional Neural Networks (CNNs) due to their superior performance on image classification tasks. Dosovitskiy et al. [1] introduced ViTs, demonstrating their ability to achieve state-of-the-art accuracy by leveraging attention mechanisms to capture global image context. However, as highlighted by recent studies, ViTs exhibit distinct vulnerabilities to adversarial attacks compared to CNNs. Mahmood et al. [2] provided an in-depth evaluation of ViTs' adversarial robustness, revealing comparable weaknesses to CNNs under white-box attacks and suggesting that architectural differences significantly influence their behavior. Similarly, Takahashi et al. [3] examined the application of ViTs in medical imaging, identifying their ability to generalize across complex datasets while emphasizing the need for robust defenses against adversarial threats in critical applications.

Despite significant advances, the robustness of hybrid models combining ViTs and CNNs remains underexplored. This study seeks to address this gap by systematically evaluating and comparing the adversarial robustness of ViTs, CNNs, and their ensembles under various attack scenarios, providing actionable insights into their security and applicability in adversarial environments.

2.1 Models

ResNet56

The ResNet56 model used in this work is implemented through a custom PyTorch class of ResNet V2 architecture sourced from Yerlan Idelbayev. The architecture consists of three main stages of convolutional layers interleaved with residual blocks, followed by a global average pooling layer and a fully connected layer for classification. The ResNet56 model is initialized using this custom implementation tailored for input images of size 32 by 32 and number of classes equaling 10, matching the CIFAR-10 dataset format.

A standard supervised learning approached training and fine-tuning process is conducted manually for this initiated ResNet56. The Cross Entropy Loss function is used for computing loss, optimized with the Stochastic Gradient Descent (SGD) optimizer. The learning rate is set to an initial value of 0.01 to ensure convergence over the 20 training epochs. The model demonstrates improved accuracy on both training and validation dataset at the end of training. The accuracy tested on clean image dataset is 77.25% for ResNet56, which can be used as the benchmark for later comparison.

Vision Transformer Base with a Patch Size of 32 (ViT-B-32)

The Vision Transformer Base with a patch size of 32 (ViT-B-32) model utilized in this work is based on a pre-trained Vision Transformer architecture provided by the `timm` library. This model employs a patch size of 32 for processing input images and is initialized with pre-trained weights tailored for image classification tasks. To adapt the model for the CIFAR-10 dataset, the final classification layer is adjusted to accommodate 10 output classes, corresponding to the dataset's class format. The input images are resized to 224 by 224 to meet the model's requirements for pre-trained initialization.

For training, a standard supervised learning approach is applied using the Cross Entropy Loss function and the Adam optimizer with a learning rate of 0.001 to achieve convergence. The training process is conducted for 20 epochs, during which the model's parameters are fine-tuned to enhance its performance on the CIFAR-10 dataset. At the end of training, the ViT-B-32 model achieved an accuracy of 81.18% on the test dataset comprising clean images, setting a benchmark for evaluating robustness against adversarial attacks and assessing transferability between models.

Ensemble Model

The ensemble model combines the Vision Transformer Large with a Patch Size of 16 (ViT-L-16) and BiT-M-101x3 to leverage their complementary strengths: ViT-L-16 excels in the extraction of global features, while BiT-M-101x3 captures local patterns. Both models are fine-tuned on the CIFAR-10 dataset with resized 224 by 224 images and adapted to 10-class output.

The ensemble model employs an averaging method to combine predictions from the two networks. Specifically, the softmax probabilities generated by each model are averaged to produce the final output probabilities. This approach ensures that both models contribute equally to the decision-making process, blending the ViT-16-L's ability to generalize across spatial regions with the BiT-M-101x3's sensitivity to fine-grained local features. By using probability averaging, the ensemble reduces the risk of over-reliance on one model's biases or weaknesses, leading to more balanced and robust predictions.

Each model in the ensemble is trained independently for 5 epochs using the Cross Entropy Loss function and the Adam optimizer, with a learning rate of 0.0001. The ensemble model demonstrates a clean accuracy of 93.34% which, while slightly lower than the ViT-L-16 model's 95.08% accuracy, surpasses the BiT-M-101x3 model's 92.44% accuracy. The ensemble approach leverages the complementary strengths of the two architectures, balancing their performance and enhancing robustness to adversarial attacks.

2.2 White-Box Attacks

Fast Gradient Sign Method (FGSM)

In this work, FGSM was utilized to craft untargeted adversarial examples through applying small, carefully scaled perturbations that remain visually similar to the original input while maximizing the probability of misclassification. Adversarial attacks are generated with a predefined ε value and valid input ranges.

Momentum Iterative Method (MIM)

The Momentum Iterative Method (MIM) strengthens adversarial attacks by introducing a momentum term in the gradient updates. This technique improves adversarial examples over multiple iterations, merging gradients from previous and current steps to guide the attack away from local optima and achieve more successful outcomes. By applying this iterative process, MIM enhances the quality and effectiveness of adversarial perturbations while respecting input constraints, making it a dependable approach for evaluating model vulnerabilities.

Projected Gradient Descent (PGD)

Projected Gradient Descent (PGD) employs an iterative approach to generate adversarial examples, progressively enhancing their effectiveness by applying multiple small perturbations. Unlike single-step attacks, this method recalculates gradients at each iteration to refine the perturbations and ensure they remain within a constrained range. For both ResNet56 and ViT-B-32 models, PGD was used to probe their robustness by introducing controlled adversarial modifications.

Auto Projected Gradient Descent (APGD)

Auto-PGD (APGD) enhances the iterative PGD attack by introducing adaptive step size adjustments, improving the generation of adversarial examples with greater precision and efficiency. This project employed APGD to assess the robustness of both ResNet56 and the ViT-B-32 models through its dynamic modification of perturbation steps and fine-tuned optimization to craft more effective adversarial inputs.

Backward Pass Differentiable Approximation (BPDA)

The Backward Pass Differentiable Approximation (BPDA) is employed in this project to handle non-differentiable layers. BPDA approximates gradients during the backward pass by replacing non-differentiable components of the model with differentiable surrogates. This approach allows for efficient computation of adversarial perturbations even when gradient obfuscation techniques, such as shattered gradients, are used as defenses. BPDA is particularly effective in exposing vulnerabilities in models that rely on these gradient-masking defenses.

Carlini and Wagner Attack (C&W)

The C&W attack was applied to specifically tailor for ResNet56 and ViT-B-32 models to analyze their resilience against subtle, well-optimized adversarial perturbations. This method prioritized precision by minimizing the L_2 norm of modifications, ensuring that the perturbations remained imperceptible while achieving misclassification.

Self-Attention Gradient Attack (SAGA)

The Self-Attention Gradient Attack (SAGA) is an advanced adversarial technique designed to exploit model vulnerabilities by leveraging gradient-based perturbations. SAGA is specifically applied to the ResNet56 model through iteratively computing gradients and adjusting perturbations to maximize the misclassification likelihood while keeping the changes imperceptible. The method incorporates self-attention mechanisms for transformers, but in the context of ResNet56, it directly utilizes gradient signals to craft adversarial attack and focus on critical image features through precise gradient calculations.

2.3 Black-Box Attacks (RayS)

RayS implemented on both models begins by introducing initial random perturbations to the input samples. These perturbations are refined iteratively by querying the model's predictions for perturbed inputs in positive and negative directions. Updates are made to the perturbations based on misclassification feedback until the model misclassifies the input or the query budget is exhausted. This process generates adversarial examples while ensuring that the perturbations remain imperceptible and bounded by the L_∞ norm.

2.4 Transferability Test

Transferability measurement is important in evaluating the ability of adversarial examples, crafted to fool one model, to mislead other models. This highlights shared vulnerabilities across different architectures, while exposes the ability of the adversarial attacks to generalize. Understanding transferability supports designing more robust defense mechanisms which could be used across platforms.

To calculate non-targeted transferability for generated adversarial attacks, the following definitions specified by Kaleel Mahmood [2] are followed: An attack $\mathscr{A}_{C_i}$ is applied to classifier C_i, which correctly identifies the input (x,y), to generate an adversarial example x_{adv}:

$$x_{\text{adv}} = \mathscr{A}_{C_i}(x,y) \tag{1}$$

This crafts adversarial examples that are designed to fool classifier $\mathscr{A}_{C_i}$. The adversarial example x_{adv} is considered as transferred to the new classifier C_j if:

$$\forall j = 1,\ldots,n \quad [\{C_j(x) = y\} \wedge \{C_j(x_{\text{adv}}) \neq y\}] \tag{2}$$

This condition ensures that the transferability metric only considers clean inputs correctly classified by both models and measures the success of $\mathscr{A}_{C_i}$ in misleading the target model. With 2 classifiers C_i and C_j, and a set of m input correctly identified by both, transferability from C_i to C_j is defined as:

$$t_{i,j} = \frac{1}{m}\sum_{k=1}^{m} \begin{cases} 1, & \text{if } C_j\big(\mathscr{A}_{C_i}(x_k,y_k)\big) \neq y_k, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

This matrix quantifies the shared vulnerability of the two models as higher transferability indicates that both models are vulnerable to the same attacks. The final value in percentage provides the average transfer success rate of the type of attack from source model C_i to target model C_j. In other words, this is defined as number of adversarial examples misclassified by the model.

To meet these conditions, we first filtered clean examples with necessary resizing to ensure both models correctly classify them. Next, 1000 randomly selected adversarial examples (generated by the source model using various attacks methods) are resized and tested on the target model. The test measures the transfer success rate and records the maximum over 10 epochs to account for variability.

3 Experiment

3.1 Model Robust Accuracy

The clean accuracy and robustness of ResNet56 and ViT-B-32 models are examined against a comprehensive set of white-box and black-box adversarial attacks. As outlined in Table 1, the ViT-B-32 model outperforms ResNet56 in clean accuracy, achieving 81.18% compared to ResNet56's 77.25%. This suggests that ViT-B-32 is more effective in classifying unperturbed images. However, when exposed to adversarial attacks, significant differences in robustness emerge. Under most white-box attacks, such as FGSM, MIM, PGD, APGD, and BPDA, ViT-B-32 shows lower robust accuracy than ResNet56, with accuracies dropping as low as 0.37% under PGD and 0.36% under BPDA. These results indicate that ViT-B-32 is more vulnerable to adversarial perturbations, possibly due to its reliance on attention mechanisms, which may exacerbate its sensitivity to crafted perturbations.

Table 1: Model Robust Accuracy to White-box and Black-box Attacks

Types of Attacks	**ResNet56 (%)**	**ViT-B-32 (%)**
Clean	77.25	81.18
FGSM	18.57	17.44
MIM	5.89	2.14
PGD	2.57	0.37
APGD	5.09	4.80
BPDA	0.44	0.36
C&W	9.53	26.48
SAGA	1.67	-
RayS	35.00	43.70

On the other hand, ResNet56 demonstrates higher robustness to these basic attacks, with accuracies of 18.57% under FGSM and 5.89% under MIM. However, it struggles significantly against modern and more advanced attacks like C&W and RayS, achieving only 9.53% and 35.00%, respectively. In contrast, ViT-B-32 performs better under C&W (26.48%) and RayS (43.70%), highlighting its ability to handle these sophisticated attacks more effectively. These findings reveal a trade-off between clean accuracy and adversarial robustness. ViT-B-32's higher clean accuracy makes it well-suited for scenarios without adversarial interference, but its susceptibility to simpler attacks suggests that additional defenses, such as adversarial training, are necessary. In contrast, ResNet56 offers better overall robustness to basic attacks but requires enhanced mechanisms to address vulnerabilities to modern attack strategies.

3.2 Transferability Test

A set of experiments are conducted to test transferability across the selected attacks. Table 2 highlights the transferability of adversarial examples generated by ResNet56 and ViT-B-32 when tested on each other.

Table 2: Transferability Test on White-box and Black-box Attacks.

Types of Attacks	**ResNet56 Generated Tested on ViT-B-32 (%)**	**ViT-B-32 Generated Tested on ResNet56 (%)**
FGSM	58.43	81.47
MIM	58.85	67.55
PGD	58.34	68.78
APGD	70.44	72.64
RayS	55.00	64.80

ViT-B-32 appears more robust against adversarial examples generated by ResNet56 compared to ResNet56's robustness against ViT-B-32-generated attacks. This is reflected in the overall higher transfer success rates for ResNet56 generated attacks tested on ResNet56. These attacks have transfer success rates ranging from 64.80% (RayS) to 81.47% (FGSM). This result indicates that vision transformer might contribute to improving adversarial attacks transferability while it is also more robust against attacks generated by CNN architectures. This might be caused by ViT nature of relying on attention-based feature extraction which consider entire input sequence and captures global contexts. This contributes to ViT's efficiency and scalability which might contribute to generalizing well across architectures. In contrast, CNN architectures are more effective at capturing local patterns such as textures and shapes of images [3]. This may cause it to lack the ability of generating effectively transferable adversarial attacks.

The FGSM attacks generated by ViT-B-32 exhibit a remarkably high transfer success rate when tested on ResNet56. Despite being crafted using a fundamentally different architecture, these attacks are nearly as effective in deceiving ResNet56 as FGSM attacks generated directly by ResNet56 itself (achieving an accuracy of 18.57%). The simplicity of the single-step FGSM attack contributes to this success, as it creates less model-specific perturbations and instead produces more generalized adversarial patterns. Among the various attack methods tested, APGD has the highest transfer success rate on average for both model, suggesting its powerfulness in fooling different architectures. This is likely due to the high precision and iterative optimization used in APGD. Conversely, RayS has the lowest transfer success rate (55% on ViT-B-32 and 64.8% on ResNet56). This is also foreshadowed by the high model robust accuracy results from previous section, suggesting that the implemented RayS attack is not powerful enough at fooling ResNet56 and ViT-B-32. The low transfer success rate of RayS might be caused by its black-box attack nature which is not having access to the model's internal parameters or gradients.

4 Conclusion

In conclusion, the experiments conducted offer valuable insights into the robustness of CNNs and vision transformers (ViTs) in adversarial attack scenarios. While the CNN-based ResNet56 demonstrates greater resilience to basic attack methods, the ViT-B-32 model excels against more advanced attacks, such as C&W. These findings highlight the importance of choosing the appropriate architecture based on the specific threat model. ResNet56's robustness against simpler attacks makes it a strong candidate for systems facing less sophisticated adversarial techniques. Conversely, ViT-B-32's higher clean accuracy and better performance against advanced attacks position it as a preferable choice for environments requiring both high classification accuracy and defenses against complex adversarial threats.

The transferability tests further underscore the differences in attack resilience and adversarial generation between the two architectures. Adversarial examples generated by ViT-B-32 are more transferable to ResNet56 than the reverse, indicating that the attention-based global feature extraction in ViTs creates more generalizable adversarial patterns. Vision transformer also have stronger robustness against CNN generated attacks. In real-world scenarios, this suggests that ViTs may be better suited for security defense systems that require robustness against a wide range of attacks, particularly those generated by diverse model architectures.

References

[1] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2021). URL: `https://arxiv.org/abs/2010.11929`.

[2] Kaleel Mahmood, Rigel Mahmood, and Marten van Dijk. "On the Robustness of Vision Transformers to Adversarial Examples". In: *arXiv preprint arXiv:2104.02610* (2021). URL: `https://arxiv.org/abs/2104.02610`.

[3] S. Takahashi et al. "Comparison of Vision Transformers and Convolutional Neural Networks in Medical Image Analysis: A Systematic Review". In: *Journal of Medical Systems* 48.1 (2024), p. 84. DOI: `10.1007/s10916-024-02105-8`. URL: `https://doi.org/10.1007/s10916-024-02105-8`.

A Adversarial Attack Methods

This section explains the 7 white-box attack and 1 black-box attack methods we used along with mathematical concepts. Parameters used for generating attacks using each method are summarized at the end of the section in Table 3.

Fast Gradient Sign Method (FGSM)

Fast Gradient Sign Method (FGSM) is a straight forward method that leverages the gradient of the loss function with respect to the input to craft adversarial samples. It generates a perturbation by adding a scaled sign of the gradient:

$$x_{\text{adv}} = x + \varepsilon \cdot \text{sign}(\nabla_x L(x, y; \theta))$$

where ε is the perturbation budget, L is the loss function, and $\nabla_x L$ is the gradient of the loss with respect to x. It is also important to note that this is a single step attack. The adversary only backpropagates on the model once to obtain the gradient of the loss function and then applies this directly to x.

Momentum Iterative Method (MIM)

The Momentum Iterative Method (MIM) improves upon FGSM by incorporating a momentum term that helps the adversary escape poor local maxima during optimization. The attack updates the adversarial example iteratively as:

$$g_{t+1} = \mu \cdot g_t + \frac{\nabla_x L(x_t, y)}{\|\nabla_x L(x_t, y)\|_1}$$

$$x_{t+1} = x_t + \alpha \cdot \text{sign}(g_{t+1})$$

where g_t represents the accumulated gradient at iteration t, μ is the decay factor for momentum, and α is the step size. By using momentum, MIM smooths the optimization path and increases the likelihood of finding stronger adversarial perturbations.

Projected Gradient Descent (PGD)

Projected Gradient Descent (PGD) is an iterative extension of the FGSM algorithm designed to find the smallest perturbation within a defined boundary that maximizes the model's loss. The method begins by initializing a random perturbation within a ball of radius d centered at the original input x. At each iteration, a gradient step is performed in the direction that maximizes the loss, and the resulting perturbation is projected back into the ball if it exceeds the boundary.

The k-step PGD algorithm starts with $x_0 = x$, and the perturbed image at the i^{th} step, x_i, is computed as::

$$x^i = P(x^{i-1} + \alpha \cdot \text{sign}(\nabla_x L(x^{i-1}; y; \theta)))$$

where P is a projection function that ensures the adversarial example remains within the $\varepsilon-$ball centered at x^{i-1}, and α represents the step size. The constraints on the projection are determined by the l_p norm.

Auto Projected Gradient Descent (APGD)

Auto Projected Gradient Descent (APGD) is an automated extension of Projected Gradient Descent (PGD) that dynamically adjusts the step size during adversarial attacks. The method splits the attack process into two phases: an exploration phase with larger step sizes to rapidly explore the search space and an exploitation phase with smaller step sizes to fine-tune the optimization. The step size is adjusted based on optimization progress and specific conditions that monitor the effectiveness of each iteration:

$$\sum_{i=w_{j-1}}^{w_j-1} \nVdash\{f(x^{(i+1)}) > f(x^{(i)})\} < \rho \cdot (w_j - w_{j-1}),$$

$$\eta^{(w_{j-1})} \equiv \eta^{(w_j)} \quad \text{and} \quad f_{\max}^{(w_{j-1})} \equiv f_{\max}^{(w_j)}$$

Here, w_j are checkpoints for adjusting the step size, ρ is a threshold for assessing progress, $\eta^{(w_j)}$ represents the step size at checkpoint w_j, and $f_{\max}^{(w_j)}$ denotes the highest objective value reached up to w_j. If these conditions are met, the step size is halved for subsequent iterations, allowing for precise control over the attack.

Backward Pass Differentiable Approximation (BPDA)

Backward Pass Differentiable Approximation (BPDA) is an attack designed to handle non-differentiable layers in neural networks by approximating gradients during the backward pass. It is effective against defenses relying on gradient obfuscation, including shattered gradients, stochastic gradients, and exploding/vanishing gradients.

For a network $f(x) = f_1 \circ f_2 \circ \ldots \circ f_j(x)$ with a non-differentiable layer $f_i(x)$, BPDA replaces $f_i(x)$ with a differentiable surrogate $g(x)$ during the backward pass. The gradient is computed as:

$$\nabla_x f(x) \approx \nabla_x (f_1 \circ \ldots \circ g \circ \ldots \circ f_j)(x)$$

Adversarial examples are then generated iteratively using a method similar to Projected Gradient Descent (PGD), ensuring perturbations remain within an ε-ball of the input. BPDA is particularly effective in bypassing defenses that mask gradients, exposing vulnerabilities in seemingly robust models.

Carlini and Wagner Attack (C&W)

The Carlini and Wagner (C&W) attack generates adversarial examples by solving an optimization problem that minimizes the perturbation while ensuring misclassification. The objective is given as:

$$\min \frac{1}{2} \| \tanh(\omega) + 1 - x \|_2^2 + c \cdot f\left(\frac{1}{2}(\tanh(\omega) + 1)\right),$$

where ω is a latent variable ensuring valid input range, c is a trade-off constant, and $f(x')$ is the misclassification objective defined as:

$$f(x') = \max\left(\max_{i \neq t}\{Z(x')_i\} - Z(x')_t, -\kappa\right),$$

with $Z(x')$ being the logits, t the target class, and κ controlling confidence in misclassification. The adversarial example is computed as:

$$x_{\text{adv}} = \frac{1}{2}(\tanh(\omega) + 1).$$

This iterative attack effectively minimizes L_2-norm perturbations while achieving high-confidence misclassification, making it a benchmark for evaluating model robustness.

Self-Attention Gradient Attack (SAGA)

The Self-Attention Gradient Attack (SAGA) is a white-box attack designed to exploit vulnerabilities in ensembles of Vision Transformers (ViTs) and CNNs. Its goal is to craft adversarial examples that simultaneously mislead both types of models. In this study, SAGA is applied specifically to ResNet56, focusing on its robustness under adversarial conditions, with the mathematical framework preserved for ensembles involving ViTs.

The adversarial example is iteratively updated as:

$$x_{\text{adv}}^{(i+1)} = x_{\text{adv}}^{(i)} + \varepsilon_s \cdot \text{sign}\left(G_{\text{blend}}(x_{\text{adv}}^{(i)})\right),$$

where $x_{\text{adv}}^{(1)} = x$, ε_s is the step size, and $G_{\text{blend}}(x_{\text{adv}}^{(i)})$ is the blended gradient, defined as:

$$G_{\text{blend}}(x_{\text{adv}}^{(i)}) = \sum_{k \in K} \alpha_k \frac{\partial L_k}{\partial x_{\text{adv}}^{(i)}} + \sum_{v \in V} \alpha_v \phi_v \odot \frac{\partial L_v}{\partial x_{\text{adv}}^{(i)}},$$

where:

- K is the set of CNN models, and V is the set of ViTs,
- $\frac{\partial L_k}{\partial x^{(i)}_{\text{adv}}}$ and $\frac{\partial L_v}{\partial x^{(i)}_{\text{adv}}}$ are the gradients of the loss functions for the respective models,
- α_k and α_v are weighting factors balancing the contributions of CNNs and ViTs, and
- ϕ_v is the self-attention map for each ViT, defined as:

$$\phi_v = \prod_{l=1}^{n_l} \left(\sum_{i=1}^{n_h} \left(0.5 W_{l,i}^{(\text{att})} + 0.5 I \right) \right) \odot x,$$

where n_l is the number of attention layers, n_h is the number of attention heads, $W_{l,i}^{(\text{att})}$ is the attention weight matrix, I is the identity matrix, and x is the input image. This technique takes into account the attention flow from each layer of the transformer to the next layer, including the effect of skip connections. The attention values from the different attention heads within the same layer are averaged, and the attention values are recursively multiplied between different layers.

Black-box Attack (RayS)

RayS is a query-efficient black-box adversarial attack method designed to evaluate the robustness of machine learning models without requiring access to gradients or internal parameters. Unlike traditional gradient-based methods, RayS identifies adversarial perturbations by conducting binary queries to the target model in an iterative manner. The search process is optimized through a systematic refinement of the perturbation direction, reducing the number of queries needed to uncover adversarial examples. The objective is to determine whether a perturbed input $\hat{x}$ satisfies the adversarial condition $f(\hat{x}) \neq y$, where f is the model and y is the true label.

The optimization problem solved by RayS can be expressed as:

$$\hat{x} = \arg \min_{x+\delta \in \mathscr{B}_\varepsilon(x)} \nVdash [f(x+\delta) \neq y]$$

where $\mathscr{B}_\varepsilon(x)$ is the ε-ball centered around the input x, and δ represents the perturbation.

Here, $\mathscr{B}_\varepsilon(x)$ denotes the set of valid inputs within an ε-distance from x, ensuring that perturbations remain imperceptible. The term $\nVdash [f(x+\delta) \neq y]$ is an indicator function that evaluates to 1 when the model misclassifies the perturbed input and 0 otherwise. Through its efficient exploration of the perturbation space, RayS reduces the number of queries needed to achieve misclassification and proves to be a practical tool for black-box adversarial testing.

Table 3: Attack Parameters for White-box and Black-box Attacks

Types of Attacks	Parameters
FGSM	$\varepsilon = 0.031$
MIM	$\varepsilon = 0.031$, $\varepsilon_{step} = 0.00155$, steps = 10, decay factor = 1.0
PGD	$\varepsilon = 0.031$, $\varepsilon_{step} = 0.00155$, steps = 20
APGD	$\varepsilon = 0.031$, steps = 10, number of restarts = 1, $\rho = 0.75$, n^2 queries = 5000
BPDA	$\varepsilon = 0.031$, steps = 10, max iterations = 100, learning rate = 0.5
C&W	c = 1.0, steps = 100
SAGA	$\varepsilon = 0.031$, steps = 50, step size = 0.00062, coefficients = [1.0]
RayS	$\varepsilon = 0.031$, query budget = 100

B Additional Findings with Ensemble Model

In this section, we present the performance of the ensemble model, which combines ViT-L-16 and BiT-M-101x3, under three types of white-box adversarial attacks: FGSM, MIM, and PGD. Additionally, we analyze the transferability of adversarial examples generated by the ensemble model to two benchmark models, ViT-B-32 and ResNet56.

Table 4: The Accuracy of the ensemble Model (ViT-L-16 and BiT-M-101x3) under three White-box Attacks and Transferability of the Adversarial Examples to the Benchmark Models

Attacks Types	**Accuracy (%)**	**Transferability on ViT-B-32 (%)**	**Transferability on ResNet56 (%)**
FGSM	35.96	42.08	57.92
MIM	2.03	41.20	58.80
PGD	0.08	41.49	58.51

The findings highlight the vulnerability of the ensemble model to all three white-box attacks, with its robust accuracy dropping to as low as 0.08% under the PGD attack. This severe drop underscores the ensemble model's susceptibility to adversarial perturbations, particularly with stronger iterative attacks like MIM and PGD. Despite combining two distinct architectures, ViT-L-16 and BiT-M-101x3, the ensemble does not exhibit significant robustness improvements against adversarial attacks, suggesting that architectural diversity alone is insufficient to counter these threats.

Interestingly, FGSM, a single-step attack, allows the ensemble model to retain a relatively higher accuracy (35.96%) compared to the iterative methods, MIM (2.03%) and PGD (0.08%). This contrast highlights the efficacy of iterative attacks in degrading model performance, as they iteratively refine perturbations to maximize mis-classification. The superior effectiveness of iterative attacks further emphasizes the importance of developing more robust defense mechanisms tailored to ensemble models.

Notably, the adversarial examples generated by the ensemble model demonstrate significantly higher transferability to ResNet56 compared to ViT-B-32 across all attack types. For instance, PGD examples show a transferability rate of 58.51% on ResNet56 compared to 41.49% on ViT-B-32. This suggests that the ensemble model's attacks are more aligned with the vulnerabilities of ResNet architectures, potentially due to structural differences between vision transformers and convolutional networks.