Defending Against High-Intensity Adversarial Perturbations in Deep Neural Networks: A Robust Swin Transformer Approach

Quang Le, *University of Ottawa*
Francois Chan, *Royal Military College*
Jianbing Ni, *Queen's University*
Scott Yam, *Queen's University*
Ning Lu, *Queen's University*

***Abstract*—High-intensity adversarial perturbations present significant challenges to the reliability of deep learning models, underscoring the urgent need for robust and adaptive defense mechanisms. These perturbations notably impact the feature extraction process within models. In this paper, we propose the Robust Swin Transformer (RST), an end-to-end trainable model that employs a Double-Branch (DB) attention mechanism to effectively extract both robust and non-robust features across various representation levels. To enhance resilience against adversarial attacks, we implement a tailored combination of robust and non-robust loss functions, demonstrating that non-robust features correlate with multiple fake classes, which can be optimized by adjusting the non-robust loss. During inference, predictions are made by fusing representation features from all model stages, striking a balance between accuracy and robustness. Extensive experiments on CIFAR-10, SVHN, and MSTAR datasets show that RST outperforms the standard Swin Transformer with adversarial training and other advanced methods, achieving up to a 9.86% accuracy improvement against PGD attacks, particularly under high-intensity perturbations.**

I. Introduction

Deep Neural Networks (DNNs) have been widely adopted in various fields, particularly in image classification tasks. However, the vulnerability of DNNs to adversarial attacks, which exploit weaknesses in their operation and impact their accuracy, is a significant concern. Various methods have been employed to defend against attacks and enhance the robustness of DNNs, including data preprocessing, generalization, and adversarial training [1]. Nevertheless, the utilization of adversarial methods for defence presents certain challenges.

One significant challenge in addressing adversarial attacks is their diversity, which varies based on the algorithms employed and the attacker's knowledge of the target model [2]. These attacks can be classified as white-box, where the attacker has full access to the model's parameters, or black-box, where knowledge is limited [3]. Consequently, distinct attack generators utilize different initial parameters and requirements. Among the critical shared parameters, perturbation intensity, defined through norm-based formulations, is particularly influential [4]. Even minor pixel-level perturbations can propagate to higher-level features, severely degrading model performance [5]. In this paper, we define robust features as those that distinctly characterize a class, while non-robust features comprise background information that is more susceptible to adversarial perturbations. The proposed Robust Swin Transformer (RST) effectively classifies these two feature groups, enabling more robust decision-making in the presence of various attacks, including both white-box and black-box settings.

Among conventional adversarial methods, adversarial training is one of the most effective defences. However, it is sensitive to adversarial diversity and requires careful balancing of robustness and accuracy. Additionally, internal factors such as neural layer components and architectural design significantly influence the robustness of DNNs. To tackle these challenges, the RST integrates a novel Double-Branch (DB) attention mechanism, enabling the extraction of both robust and non-robust features across multiple representation levels. To further enhance resilience against adversarial attacks, a combination of loss functions is employed, ensuring a clear distinction between robust and non-robust components. A comprehensive analysis of features at various stages of the model bolsters overall robustness against high-intensity perturbation attacks. Experiments under various adversarial attack scenarios demonstrate that the proposed method mitigates the trade-off between robustness and accuracy more effectively than the traditional adversarial training method. This paper presents an end-to-end trainable transformer-based model demonstrating significant robustness against adversarial threats. The contributions of this research are summarized as follows:

- Design the RST model with a DB attention mechanism, enabling the extraction of robust and non-robust features across multiple levels of representation. This model achieves robustness against various adversarial attacks.
- Formulate a composite loss function integrating robust, non-robust, and reconstruction losses. The analysis shows that optimizing the non-robust loss, which correlates with multiple fake classes, enhances overall efficacy. During inference, we examine the fusion of outputs to achieve an optimal balance between robustness and accuracy.
- Experimental evaluations are conducted on the image classification task using three datasets, CIFAR-10, SVHN, and MSTAR. The proposed method's performance is assessed against a range of white-box and black-box attacks. Results consistently highlight the superiority of the proposed approach, demonstrating exceptional performance and robust resilience in most scenarios.

II. Related Works

A. Adversarial Attacks

Adversarial attacks intentionally exploit vulnerabilities in DNNs by introducing subtle perturbations that lead to misclassifications. These attacks are quantified using norms to measure perturbation magnitudes and fall into three main categories: gradient-based attacks, such as the Fast Gradient Sign Method (FGSM) [6], which use gradients to generate adversarial samples; constrained optimization-based attacks, like Projected Gradient Descent (PGD) [6, 7], which treat adversarial generation as an optimization problem; and gradient-free (black-box) attacks, utilizing methods like random search or evolutionary algorithms, making them model-agnostic, as seen in Square Attack [8] and Pixle [9].

B. Adversarial Training

Significant research has focused on adversarial defence techniques to enhance the robustness of DNNs against attacks. A commonly used method is standard adversarial training, including approaches like PGD [6], TRADES [10], and MART [11]. PGD training uses iterative perturbations based on gradients, while TRADES balances adversarial robustness and natural accuracy through regularization. MART employs margin-based loss functions to improve resilience. Additionally, advanced loss functions, such as the Latent Feature Relation Consistency (LFCR) approach, ensure consistency among latent features [12]. While these methods focus on modifying the loss function, they do not leverage the underlying neural architecture. As a result, the feature extraction architecture and output format remain unchanged, rendering these methods heavily dependent on the adversarial training samples.

C. Feature Extraction as Adversarial Defense

Advanced defence mechanisms beyond standard adversarial training have integrated feature extraction techniques [13]. DNNs are trained to extract robust global and local features, as non-robust features significantly contribute to adversarial examples [14]. A distillation approach based on the Information Bottleneck framework addresses both robust and non-robust features [15]. Zhang et al. proposed hierarchical feature alignment for robust learning from clean and adversarial samples, while Wang et al. demonstrated a method to segregate robust and non-robust features. Cao et al. introduced the Feature Pyramid Network (FePN) to enhance robust feature learning; however, it requires additional storage for robust images and lacks experimental validation against high-intensity perturbations. Kim et al. presented the Feature Separation and Recalibration (FSR) method, which recalibrates attention to emphasize resilient characteristics [16]. While this approach is similar to ours, it extracts non-robust features based solely on a single fake class. In practice, background and outlier information in images are inherently uncertain, potentially leading to multiple fake classes. We assert that utilizing appropriate groupings of fake classes can enhance non-robust feature extraction and improve overall performance.

III. Methods

The RST model utilizes a hierarchical structure similar to the Swin Transformer (ST) for multi-scale feature extraction [17], as detailed in Section III-A. The original architecture, however, supports only a single information flow, hindering the differentiation between robust and non-robust features. To address this, we introduced the RST block with DB attention for simultaneous extraction of both information flows, as detailed in Section III-B. This yields robust and non-robust outputs, along with their combined reconstruction output. A tailored loss function guides training to produce a final robust prediction, as shown in Sections III-C and III-D.

A. Hierarchical and Multiple-stage Structure

As shown in Fig. 1, the main structure consists of N stages, beginning with a patch embedding block that partitions the input, followed by ST blocks [17], an RST block, a classification network, and a reconstruction network. This structure enhances defensive capabilities by integrating global and local features for comprehensive learning and distillation. The RST model extracts robust and non-robust features in pairs, ensuring that robust features preserve essential information for accurate predictions, while non-robust features are linked to redundant information, such as outliers and background.

In the initial stage, input data is duplicated into two states after passing through the embedding block, each containing robust and non-robust information. One state undergoes ST blocks to produce the initial robust feature, while the other serves as the initial non-robust feature. These features are then processed by the RST block to generate a new pair, resulting in both robust and non-robust outputs. This process is iterated through subsequent stages, with robust features retaining the critical information necessary for decision-making. Inspired by [16], which emphasizes the value of non-robust features in model predictions, a final pair of features is combined using a reconstruction network to generate the reconstruction feature.

B. Double-Branch Attention

The RST block, unlike the ST block, features two branches with a bottleneck at the DB attention stage (see Fig. 2). This design allows the reconstruction of robust and non-robust states. The RST block uses a general attention mechanism to learn global dependencies and extract features. Its main operational steps are as follows:

In the initial stage, a "Mapping QKV" block projects a pair of robust and non-robust states, producing four tensors: query $\boldsymbol{q}$, key $\boldsymbol{k}$, value $\boldsymbol{v}$, and robust mask $\boldsymbol{r}$.

$$\boldsymbol{q}_{i-1}, \boldsymbol{v}_{i-1}, \boldsymbol{k}_{i-1}, \boldsymbol{r}_{i-1} \quad = \quad \boldsymbol{W}_m * \{f^r_{i-1}, f^{nr}_{i-1}\}. \quad (1)$$

In (1), i denotes the stage order, and $\boldsymbol{W}_m$ represents the linear layer parameters. The robust mask is activated by a sigmoid function to constrain its values between 0 and 1. The key

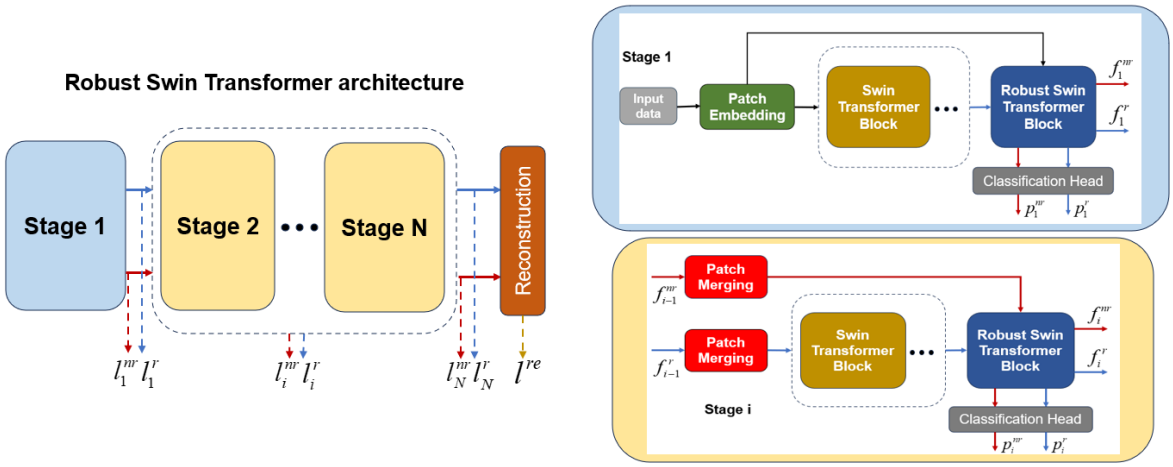

Fig. 1: RST architecture.

tensor and robust mask are then processed through the "Robust Masking" block to produce robust and non-robust keys.

$$\boldsymbol{k}_{i-1}^{r} = \sigma(\boldsymbol{r}_{i-1}) \circ \boldsymbol{k}_{i-1} \quad (2)$$

$$\boldsymbol{k}_{i-1}^{nr} = (1 - \sigma(\boldsymbol{r}_{i-1})) \circ \boldsymbol{k}_{i-1}. \quad (3)$$

In (2), σ is the sigmoid function, and $\circ$ denotes the Hadamard product. The key tensor captures features of both robust and non-robust dimensions. The Hadamard product with the robust mask generates two keys, enabling the production of robust state f_i^r and non-robust state f_i^{nr}.

$$\text{Attn}(\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}) = \text{softmax}\left(\frac{\text{cosine}(\boldsymbol{q}, \boldsymbol{k})}{\tau}\right). \quad (4)$$

$$f_i^a = \text{Attn}(\boldsymbol{q}_{i-1}, \boldsymbol{k}_{i-1}^a, \boldsymbol{v}_{i-1}). \quad (5)$$

In (4), "Attn" refers to the primary attention mechanism in a transformer model, with a representing either r or nr. Most attention mechanisms utilize $\boldsymbol{q}$, $\boldsymbol{k}$, and $\boldsymbol{v}$, enabling the proposed DB operation to be easily adapted for extracting robust and non-robust features in other transformers.

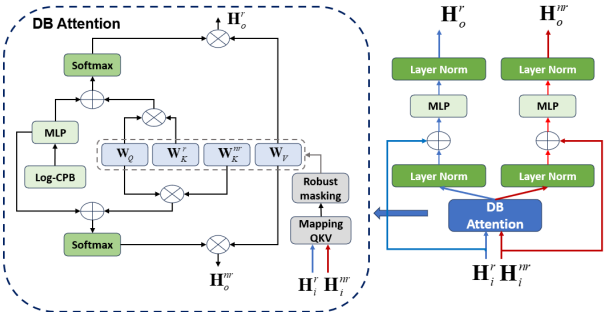

Fig. 2: RST block and DB attention structure.

C. Model Outputs and Loss Functions

As shown in Fig. 1, the model generates a pair of robust and non-robust features at each stage, which are then passed through the reconstruction network to produce the reconstructed feature. Each stage also processes the features with a classification head to output probabilities for all classes.

$$\begin{aligned} f_i^r, f_i^{nr} &= \boldsymbol{\theta}_i(f_{i-1}^r, f_{i-1}^{nr}) \\ f^{re} &= \boldsymbol{\beta}(f_N^r, f_N^{nr}) = f_N^r + \tanh(\boldsymbol{W}_\beta\{f_N^r, f_N^{nr}\}) \circ f_N^{nr} \\ \boldsymbol{p}_i^r &= \boldsymbol{\gamma}_i(f_i^r), \boldsymbol{p}_i^{nr} = \boldsymbol{\gamma}_i(f_i^{nr}), \boldsymbol{p}^{re} = \boldsymbol{\gamma}_N(f^{re}). \end{aligned} \quad (6)$$

In (6), i indicates the stage order, $\boldsymbol{p}_i$ denotes output probabilities, and $\boldsymbol{\theta}$, $\boldsymbol{\gamma}$, and $\boldsymbol{\beta}$ correspond to feature extraction, the classification head for stage i, and the reconstruction block, respectively. The classification head comprises linear layers, ReLU activations, and a Softmax layer for class probabilities, while the reconstruction block uses a gating technique with weights $\boldsymbol{W}_\beta$ linked to a linear layer for data reconstruction.

To effectively acquire robust and non-robust features, our approach utilizes a combination of loss functions: robust loss, non-robust loss, and reconstruction loss. The robust loss is computed using cross-entropy functions that consider the ground truth class of the input.

$$l^r = \frac{\sum_{i=1}^N w_i \cdot l_i^r}{\sum_{i=1}^N w_i} = \frac{\sum_{i=1}^N w_i \cdot \text{CE}\left(\boldsymbol{p}_i^r, \boldsymbol{p}_{gt}^r\right)}{\sum_{i=1}^N w_i}. \quad (7)$$

In (7), w_i denotes the weight for stage i, $\boldsymbol{p}_i^r$ represents the probability vector for the K classes, and $\boldsymbol{p}_{gt}^r$ indicates the ground truth probabilities, set to 1 for the true class and 0 for others. This setup ensures that extracted robust features focus on the true class, enhancing the model's discriminative ability.

Previous research has focused on extracting robust features [13] or learning non-robust features from a single high-probability fake class [16]. However, non-robust features can be influenced by the image background, confusing the model and causing misclassifications, especially with unstable multiple classes. To address this, we show that using multiple fake classes enhances feature extraction. Specifically, we introduce the hyperparameter k, which determines the number of fake classes $\boldsymbol{c}_i^{fake}$ for the non-robust features.

$$\boldsymbol{c}_i^{fake} = \underset{c_{true} \notin \boldsymbol{c}_i^{fake}}{\text{argtopk}} \left(\boldsymbol{p}_i^{nr}\right) = \underset{c_{true} \notin \boldsymbol{c}_i^{fake}}{\text{argtopk}} \left(\gamma(f_i^{nr})\right). \quad (8)$$

In (8), $\boldsymbol{c}_i^{fake}$ denotes the subset of k fake classes with the highest probabilities at stage i, obtained via the "argtopk" operation, where $k \leqslant K$. Meanwhile, c^{true} represents the true class. The ground truth probability for the non-robust output based on the classes in $\boldsymbol{c}_i^{fake}$ can be defined as follows:

$$\begin{aligned} \boldsymbol{p}_{gt,i}^{nr} &= \{p_{gt,i,j}^{nr}\}_{j=1}^{K} \\ p_{gt,i,j}^{nr} &= \begin{cases} \frac{1}{k} & \text{if } j \in \boldsymbol{c}_i^{fake} \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{9}$$

By assigning equal values to all k fake classes, the model is encouraged to learn highly uncertain non-robust features, leveraging Shannon's entropy principles [18]. This approach fosters ambiguity and randomness in the non-robust feature representation, aligning with information theory. Consequently, the non-robust loss can be derived as follows:

$$l^{nr} = \frac{\sum_{i=1}^{N} w_i \cdot l_i^{nr}}{\sum_{i=1}^{N} w_i} = \frac{\sum_{i=1}^{N} w_i \cdot \mathrm{KL}\left(\boldsymbol{p}_i^{nr}, \boldsymbol{p}_{gt,i}^{nr}\right)}{\sum_{i=1}^{N} w_i} \tag{10}$$

where "KL" refers to the Kullback-Leibler divergence. To guide the learning phase of the proposed method, we apply a combination of robust, non-robust, and reconstruction losses.

$$l = l^{r} + l^{nr} + l^{re}. \tag{11}$$

D. Fusion Inference

During inference, we evaluate the integration of robust, non-robust, and final reconstruction outputs as probabilities for K classes. This enables us to assess the trade-off between robustness and accuracy in our method. Analyzing this output fusion offers insights into the balance between the model's resilience to adversarial attacks and its predictive accuracy. The final results are summarized as follows:

$$\begin{aligned} \boldsymbol{p}_{out} &= [(\boldsymbol{p}^{r}) - \boldsymbol{p}^{nr}] + \boldsymbol{p}^{re} \\ &= \frac{\left[\left(\sum_{i=1}^{N} w_i \boldsymbol{p}_i^{r}\right) - \sum_{i=1}^{N} w_i \boldsymbol{p}_i^{nr}\right]}{\sum_{i=1}^{N} w_i} + \boldsymbol{p}^{re}. \end{aligned} \tag{12}$$

IV. Experiments

A. Model Configuration

Key hyperparameters, such as the number of stages, layers, window and kernel size are predetermined based on the ST tiny version [17]. The main distinction is replacing the last ST block in each stage with the RST block, enabling the extraction of robust and non-robust features.

RST configuration: The number of stage $N = 4$; the numbers of layers for stages $L = (2, 2, 6, 2)$; the hidden channels $C = 96$; the weights for stages $W = (0.5, 0.8, 1.1, 1.5)$.

Learning hyperparameters: the maximum number of epochs $T = 300$, the batch size $B = 64$, the optimizer Adam (learning rate $l = 0.0001$, decay rate $d = 0.99$).

B. Datasets

MSTAR is a radar dataset, collected in 1998 and sponsored by DARPA and AFRL, comprising 6874 SAR images of 10 military vehicle types. **CIFAR-10** contains images categorized into ten distinct classes. Each image is 32×32 pixels in size, featuring three colour channels (RGB) with pixel intensities. **SVHN** consists of over 600000 colour images of house numbers captured from Google Street View.

C. Hyperparameter Tuning

We focus on optimizing the number of fake classes for non-robust feature extraction while keeping parameters fixed. By training the model with various hyperparameter combinations, we aim to identify the configuration that maximizes performance. Accuracy results across different fake class sets are detailed in Section IV-A. Training uses adversarial examples with a perturbation intensity of $\epsilon = 16/255$, while validation includes clean data and PGD examples with $\epsilon = 8/255$ and $16/255$. Experiments explore the impact of different fake class numbers, as shown in Fig. 3, where ki denotes a set of (i, i, i, i) for each stage. We defined two fake class sets: setA $= (7, 5, 3, 1)$ and setB $= (1, 3, 5, 7)$.

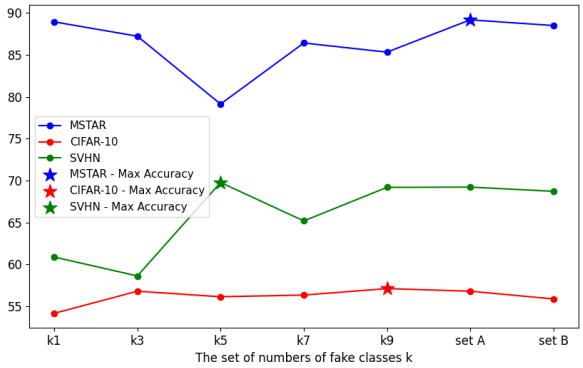

Fig. 3: Analysis of Accuracy Across Different Sets of Numbers of Fake Classes of Non-Robust Features.

Fig. 3 illustrates performance variations across different k sets for the MSTAR, SVHN, and CIFAR-10 datasets, highlighting the significance of hyperparameter tuning. Performance fluctuations are more pronounced in MSTAR and SVHN than in CIFAR-10, suggesting a strong link between the number of fake classes and the extraction of non-robust features. The configuration setA $= (7, 5, 3, 1)$ achieves the highest performance for MSTAR, with optimal values of $k = 5$ for CIFAR-10 and $k = 9$ for SVHN. The effectiveness of setA indicates that fewer fake classes enhance the extraction of higher-level non-robust features. These insights will inform our selection of fake classes in future experiments.

D. Trade-off between Robustness and Accuracy

Fusion Inference. In the context of adversarial methods, accuracy refers to the evaluation metric when testing the model with clean data, while robustness values are obtained from evaluating the model's performance on adversarial examples.

To investigate this trade-off, we assess fusion inference with various types. Table I highlights the relationship between

TABLE I: Accuracy results of MSTAR dataset with different inference types.

RST $\boldsymbol{p}_{out}$	Clean Data	PGD$_{\epsilon=8/255}$	PGD$_{\epsilon=32/255}$
$\boldsymbol{p}^{re}$	99.39	95.01	69.07
$\boldsymbol{p}^{r}$	98.39	96.32	76.42
$\boldsymbol{p}^{r} - \boldsymbol{p}^{nr}$	97.91	96.78	78.02
$\boldsymbol{p}^{re} + \boldsymbol{p}^{r} - \boldsymbol{p}^{nr}$	99.29	96.53	76.92

accuracy and robustness, shaped by the model's features. The reconstruction output achieves the highest accuracy on clean data but lacks robustness against adversarial examples. Combining robust and non-robust features enhances resilience, while merging all features strikes a balance, yielding the second-best metrics for accuracy and robustness. We will use this inference approach in future experiments.

TABLE II: Accuracy results of MSTAR dataset with models trained with different PGD samples perturbation intensity.

RST $\boldsymbol{p}_{out}$	Clean Data	PGD$_{\epsilon=8/255}$	PGD$_{\epsilon=32/255}$
RST ($\epsilon = 8/255$)	99.31	97.23	73.45
ST + AT ($\epsilon = 8/255$)	98.55	96.78	45.12
RST ($\epsilon = 16/255$)	99.29	96.53	76.92
ST + AT ($\epsilon = 16/255$)	98.24	93.61	58.06
RST ($\epsilon = 32/255$)	99.11	95.12	79.26
ST + AT ($\epsilon = 32/255$)	95.82	91.28	65.74

Training perturbation intensity: We analyze the relationship between the perturbation intensity of training samples and the model's performance by training the RST using PGD samples generated with different perturbation intensities. To further assess the trade-off between accuracy and robustness, we compare the proposed method with the ST model utilizing adversarial training. Table II shows the trade-off between accuracy and robustness with perturbation intensity. Lower intensity boosts accuracy but reduces robustness, while higher intensity enhances robustness at the cost of accuracy. The RST model outperforms conventional methods, achieving a superior balance with less sensitivity to perturbations.

E. Adversarial Diversity

Different perturbation intensities. We analyze adversarial diversity by examining the effects of varying perturbation intensities and types of attacks. First, we evaluate the RST model trained with PGD examples at a perturbation intensity of $\epsilon = 16/255$. We then compare its performance with PGD examples generated at different ϵ values. For a comprehensive assessment, we also implement ST with adversarial training and ST with FSR for comparative analysis.

Fig. 4 shows that the RST model achieves the highest accuracy with clean data and exhibits superior robustness. As perturbation intensity increases, its shallower slopes indicate reduced sensitivity compared to other models. These results highlight the RST's resilience against adversarial attacks while maintaining accuracy across perturbation levels.

Different adversarial attacks: To evaluate adversarial diversity, we assess the method in white-box and black-box settings,

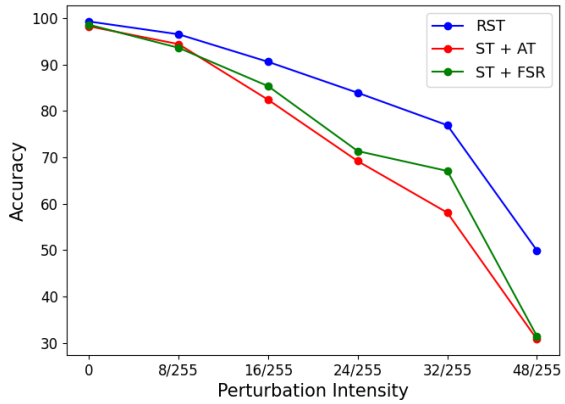

Fig. 4: Accuracy results of the proposed methods and the baseline under PGD attack with varying perturbation intensities.

using the l_∞ norm for perturbation intensities. All models are trained with PGD samples at $\epsilon = 16/255$ and tested with various attacks at different ϵ values. In white-box settings, we implement PGD, FGSM, and VMIFGSM attacks with a step size of $4/255$ for 10 iterations. In black-box settings, we use square and pixel attacks, allowing up to 500 queries.

In conclusion, Table III shows the RST method's superior performance compared to the baseline, achieving better results across various datasets and adversarial scenarios. In the MSTAR dataset, RST attains the highest accuracy with clean data, outperforming competitors in 7 of 8 cases against PGD ($\epsilon = 32/255$), VMIFGSM ($\epsilon = 32/255$), and Square ($\epsilon = 32/255$) attacks. It also excels in the CIFAR-10 dataset, particularly in black-box settings, and demonstrates strong robustness on the SVHN dataset against high-intensity perturbation attacks ($\epsilon = 32/255$).

V. Conclusion

In this paper, we proposed the Robust Swin Transformer (RST), a novel transformer model that employs a Double-Branch (DB) attention mechanism to extract both robust and non-robust features. Experimental results demonstrate that the RST achieves an excellent balance between accuracy and robustness, exhibiting strong performance against a variety of white-box and black-box attacks. However, challenges persist in adapting this approach to other transformer models and in managing the additional computational costs associated with equipping the RST with the necessary weights. In future work, we aim to design a general feature distillation framework that can be integrated into a wide range of classification models, not limited to transformer-based architectures, to enhance robust feature extraction.

References

[1] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, and X. Yi, "A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability," *Computer Science Review*, vol. 37, p. 100270, 2020.

TABLE III: The accuracy results of the proposed method and previous methods against different datasets.

		MSTAR							
		White-Box					Black-Box		
Model	Clean Data	PGD $\epsilon = 8/255$	PGD $\epsilon = 32/255$	FGSM $\epsilon = 8/255$	FGSM $\epsilon = 32/255$	VMIFGSM $\epsilon = 32/255$	Square $\epsilon = 8/255$	Square $\epsilon = 32/255$	Pixle
RST	**99.29**	**96.53**	**76.92**	**97.08**	78.53	**70.23**	**98.54**	**69.99**	**98.79**
ST + AT	98.24	93.61	58.06	94.72	70.99	54.34	96.17	39.81	93.56
ST + FSR	98.59	94.41	67.06	95.52	**84.26**	68.12	96.53	59.32	95.35
ST + LFRC	97.98	92.51	59.37	93.61	75.21	53.64	96.43	53.19	94.21
ResNet101 + FSR [16]	98.79	95.72	49.52	96.07	62.19	44.84	97.58	58.42	92.45
ResNet101 + LFRC [12]	99.11	96.02	62.14	96.52	75.66	56.71	97.88	72.81	81.14
		CIFAR-10							
RST	**77.34**	**55.16**	**35.21**	**61.27**	**49.81**	**34.73**	**59.92**	**52.34**	**63.28**
ST + AT	59.43	48.41	30.68	53.28	42.75	30.29	29.24	22.06	27.12
ST + FSR	66.81	48.79	28.84	52.56	37.56	32.14	52.87	47.89	61.39
ST + LFRC	52.69	47.71	31.23	49.41	34.51	27.74	46.07	42.09	44.28
ResNet101 + FSR [16]	72.84	49.68	29.27	49.81	34.98	31.23	44.45	30.19	44.33
ResNet101 + LFRC [12]	62.86	48.16	20.42	45.39	31.76	29.34	28.52	12.69	36.17
		SVHN							
RST	92.92	**67.17**	**44.82**	**74.21**	**64.49**	**49.31**	70.90	**63.75**	**76.58**
ST + AT	72.87	66.15	41.71	71.61	58.48	44.44	69.26	55.62	71.28
ST + FSR	89.21	66.22	41.98	70.45	54.45	41.69	**72.16**	60.23	73.76
ST + LFRC	81.45	63.12	39.11	65.12	51.23	40.45	56.79	51.33	58.69
ResNet101 + FSR [16]	**93.85**	60.84	38.99	66.61	55.61	37.84	69.36	55.82	74.21
ResNet101 + LFRC [12]	83.31	56.72	36.11	62.66	52.61	35.12	57.53	38.06	68.89

[2] H. Kwon and J. Lee, "Diversity adversarial training against adversarial attack on deep neural networks," *Symmetry*, vol. 13, no. 3, p. 428, 2021.

[3] Y. Wang, T. Sun, S. Li, X. Yuan, W. Ni, E. Hossain, and H. V. Poor, "Adversarial attacks and defenses in machine learning-empowered communication systems and networks: A contemporary survey," *IEEE Communications Surveys & Tutorials*, 2023.

[4] L. Schwinn, R. Raab, A. Nguyen, D. Zanca, and B. Eskofier, "Exploring misclassifications of robust neural networks to enhance adversarial attacks," *Applied Intelligence*, vol. 53, no. 17, pp. 19 843–19 859, 2023.

[5] C. Schlarmann and M. Hein, "On the adversarial robustness of multi-modal foundation models," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 3677–3685.

[6] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[7] Y. Li, M. Cheng, C.-J. Hsieh, and T. C. Lee, "A review of adversarial attack and defense for classification methods," *The American Statistician*, vol. 76, no. 4, pp. 329–345, 2022.

[8] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, "Square attack: a query-efficient black-box adversarial attack via random search," in *European conference on computer vision*. Springer, 2020, pp. 484–501.

[9] J. Pomponi, S. Scardapane, and A. Uncini, "Pixle: a fast and effective black-box attack based on rearranging pixels," in *2022 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2022, pp. 1–7.

[10] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International conference on machine learning*. PMLR, 2019, pp. 7472–7482.

[11] Y. Wang, D. Zou, J. Yi, J. Bailey, X. Ma, and Q. Gu, "Improving adversarial robustness requires revisiting misclassified examples," in *International conference on learning representations*, 2019.

[12] X. Liu, H. Kuang, H. Liu, X. Lin, Y. Wu, and R. Ji, "Latent feature relation consistency for adversarial robustness," *arXiv preprint arXiv:2303.16697*, 2023.

[13] C. Xie, Y. Wu, L. v. d. Maaten, A. L. Yuille, and K. He, "Feature denoising for improving adversarial robustness," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 501–509.

[14] A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," *Advances in neural information processing systems*, vol. 32, 2019.

[15] J. Kim, B.-K. Lee, and Y. M. Ro, "Distilling robust and non-robust features in adversarial examples by information bottleneck," *Advances in Neural Information Processing Systems*, vol. 34, pp. 17 148–17 159, 2021.

[16] W. J. Kim, Y. Cho, J. Jung, and S.-E. Yoon, "Feature separation and recalibration for adversarial robustness," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 8183–8192.

[17] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong *et al.*, "Swin transformer v2: Scaling up capacity and resolution," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 12 009–12 019.

[18] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.