Understanding Robustness of Vision Transformers

Arya Sinha
2020498

BTP report submitted in partial fulfillment of the requirements
for the Degree of B.Tech. in Computer Science & Applied Mathematics
on 29 April 2024

BTP Track: Research

BTP Advisor
Dr A V Subramanyam

Indraprastha Institute of Information Technology
New Delhi

Student's Declaration

I hereby declare that the work presented in the report entitled **"Understanding Robustness of Vision Transformers"** submitted by me for the complete fulfillment of the requirements for the degree of *Bachelor of Technology* in *Computer Science & Applied Mathematics* at Indraprastha Institute of Information Technology, Delhi, is an authentic record of my work carried out under guidance of **Dr A V Subramanyam**. Due acknowledgements have been given in the report to all material used. This work has not been submitted anywhere else for the reward of any other degree.

..... **Place & Date: Delhi, 29 April 2024**
Arya Sinha

Certificate

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Place & Date: Delhi, 29 April 2024
Dr A V Subramanyam

Abstract

In this study, we explore attention-based networks, focusing on recent advancements in Vision Transformers (ViT) that surpass conventional Convolutional Neural Networks (CNNs) in various vision tasks. We investigate differences in robustness between ViTs and CNNs, analyzing vulnerabilities to adversarial samples and proposing modifications to enhance robustness.

We test ViT's robustness by altering model inputs, such as providing entire images, and expand our experiments to the CIFAR-100 dataset for a more comprehensive assessment. We introduce modifications to the loss term within the adversarial training paradigm, building upon the foundation of the TRADES loss. These enhancements deepen our understanding of ViT's capabilities and limitations across different scenarios.

Keywords: Adversarial Robustness, Adversarial Training, Computer Vision, Vision Transformer

Acknowledgments

I express my sincere appreciation to my advisor, Dr A V Subramanyam, for his continued support, guidance and patience during the project. I am grateful for the privilege of learning under his mentorship.

Contents

Chapter 1

Introduction

1.1 Motivation

Robust machine learning models play a pivotal role in ensuring the reliability and effectiveness of AI systems across diverse applications. Firstly, their importance lies in their ability to maintain high performance in the face of varied and unforeseen conditions, contributing to the overall stability of the model. In real-world scenarios, data may exhibit noise, outliers, or unexpected patterns, and robust models demonstrate resilience against such challenges, enhancing their generalization capabilities. Perturbations range from natural variations to adversarial perturbations.

Robustness against natural variations can be evaluated on the following benchmarks: ImageNet-C, ImageNet-R and ImageNet-A. It was seen that when trained on sufficient data, the accuracy of ViT models outperformed ResNets, and scaled better with model size [1].

Adversarial perturbations are small, intentionally crafted modifications applied to input data with the goal of fooling a machine learning model to produce incorrect or unexpected outputs. These perturbations are designed to be imperceptible to the human eye. Adversarial perturbations can be generated through various techniques, such as gradient-based methods or optimization algorithms, aiming to find subtle changes that have a maximal impact on the model's output.
ViTs have superior adversarial robustness against small perturbations in the input pixel space [12]. But, they do not provide much additional security over Big Transfer Models or conventional CNNs once perturbation size is increased [9]. ResNet models are more robust to the simpler FGSM attack than their ViT counterparts, but this advantage disappears for the more successful PGD attacks [1].

Ensuring adversarial robustness is crucial in applications to maintain security and trust. Robust models that can withstand adversarial attacks are needed for the for the practical deployment of machine learning models.

1.2 Research Problem

In their work, Debenedetti et al. introduced a novel training approach for Vision Transformers (ViTs), demonstrating enhanced adversarial robustness across diverse scales, architectures, datasets, and threat models [3]. Building upon this, the Robust Vision Transformer (RVT) [10] strategically combined robust components identified through analysis, achieving remarkable performance and robustness in image classification tasks.

Despite the advancements, Convolutional Neural Networks (CNNs) continue to dominate the robustbench benchmark. Intriguingly, even ViTs without adversarial training exhibit superior robustness compared to traditional CNNs when subjected to natural perturbations [3]. This observation gives an insight into the potential of ViTs for robust training, motivating our pursuit of enhancing the adversarial robustness of ViTs.

Our research aims to enhance the adversarial robustness of Vision Transformers (ViTs). We pursue this goal by investigating architectural variations, intending to develop Vision Transformers that are more robust. Through these efforts, we aspire to make a meaningful contribution to the ongoing development of adversarial robustness in deep learning models.

Chapter 2

Literature Survey

2.1 Vision Transformer

Vision Transformers (ViT) [4] are based on the transformer architecture, which was originally proposed for natural language processing (NLP) [14]. They have recently excelled in benchmarks for various computer vision applications while requiring lesser computational resources. Unlike CNNs, which operate on pixel arrays, ViT operates on patches. This model breaks down an image into fixed-size patches, embeds each patch, adds positional embedding and sends this as input to the transformer encoder.

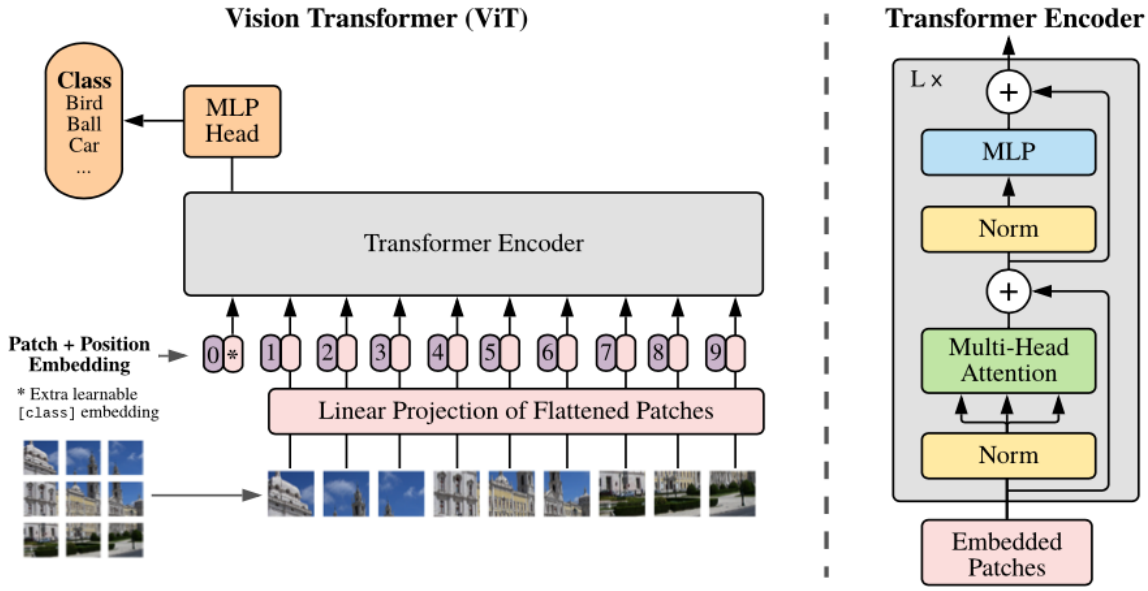

Figure 2.1: Vision Transformer Architecture

2.2 Adversarial Attacks

Adversarial perturbations are tiny changes deliberately made to input data to trick a machine learning model into giving wrong or surprising results. These changes are so small that people can't easily notice them.

A white-box attacker can see the model's parameters (structure and learned weights). The attacker can use the model's gradients to make an adversarial example directly.

A black-box attacker cannot see the model's parameters or structure, but can query the model many times, or use their own fake model to estimate the gradients and make adversarial examples.

2.3 Adversarial Training

Adversarial training is a defense method against adversarial examples that trains the model with perturbed data. It solves the following min-max optimization problem [13]:

$$\hat{\theta} = \min_{\theta} \sum_{i} \max_{\delta \in \Delta} \mathcal{L}(f_\theta(x_i + \delta), y_i)$$

Here, f_θ is a network with parameters θ, (x_i, y_i) is a training example, $\mathcal{L}$ is the loss function, and Δ is the perturbation set.

There exists a trade-off between accuracy and robustness. While adversarially training, it is important to preserve clean accuracy while increasing robust accuracy.

2.4 TRADES

A new defense method named TRADES [16] (Tradeoff-inspired Adversarial Defense via Surrogate-loss minimization) is developed in this work. This method is designed to balance the trade-off by using a regularized surrogate loss. It combines the standard classification loss (often the cross-entropy loss) with a robust loss. The robust loss penalizes the model for producing significantly different outputs when presented with slightly perturbed inputs (adversarial examples) optimizing both for accuracy on natural examples and robustness against adversarial examples.

The TRADES method was empirically validated on real-world datasets and showcased substantial improvements in robustness without significantly sacrificing accuracy. Notably, it won the NeurIPS 2018 Adversarial Vision Challenge, outperforming other entries significantly in terms of mean L2 perturbation distance.

Chapter 3

Methodology

3.1 Training Procedure Modifications

3.1.1 EMA Training

An Exponential Moving Average (EMA) model is concurrently created alongside the original model being trained. The EMA model maintains a smoothed version of the model's learned weights. During evaluation, it's the EMA model, not the original one, that is assessed.
The learning rate scheduler operates based on the EMA evaluation metrics and the best model checkpoints are saved according to the EMA evaluation as well, ensuring that the model's performance is optimized based on the smoothed EMA metrics.
This approach is introduced to enhance training stability and potentially improve generalization.

3.2 Architectural Modifications

3.2.1 EMA Modification in Attention Mechanism

This modification replaces the traditional QK' (Query-Key) attention with an EMA-based approach.

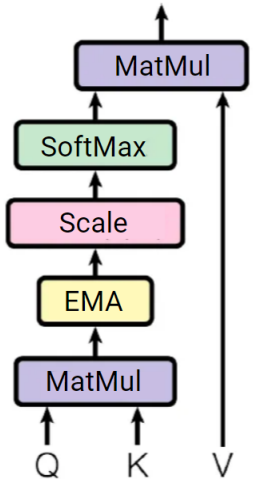

Figure 3.1: EMA layer added to Scaled-Dot Product Attention

This is done in the following way:

$$EMA_{QK'}(0) = \text{QK'}(0)$$
$$EMA_{QK'}(\text{t}) = (1 - \alpha) \times EMA_{QK'}(t-1) + \alpha \times \text{QK}'(t), \quad \text{for } \text{t} \geq 1$$

Here, α represents the EMA weight decay.

By averaging the weights over time, EMA can smooth out the effects of short-term fluctuations in the training data, leading to more stable and reliable model performance. It can make the model more robust to outliers and noise in the data. However, this approach demands additional memory.

3.2.2 Cross Attention Module

We integrate cross-attention in the self-attention mechanism. Each head not only processes its own query, key, and value but also integrates information from other heads.

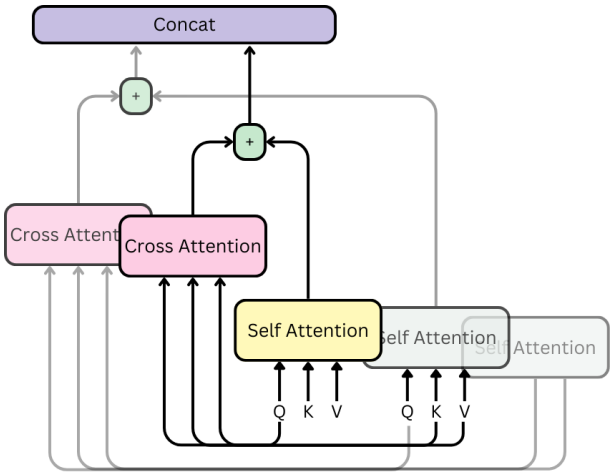

Figure 3.2: Addition of CA module to SA module

A cross attention module is built between query weights of each head and key and value weights of the next head:

Let self attention heads be denoted by h_1, h_2, ..., h_8 and input by X.
For head h_i, cross attention module is built between XW_{q_i}, $XW_{k_{i+1}}$, $XW_{v_{i+1}}$.
The output of this module is then added to the output of the self-attention for head h_i.

This process is uniformly applied across all heads and transformer blocks.

This structure allows for a richer and more diverse feature extraction as each head not only focuses on different aspects of the input but also integrates features from other heads.

3.3 Input Modifications

We explored modifying the input provided to the model. One method involved adding an entire image as an extra patch to enrich the model's understanding.

The motivation behind including an additional patch was to provide the model with supplementary information or enhance its ability to learn more generalizable features.

We experimented with two main strategies for including the additional patch:

1. Vanilla Approach: The input image was first downsized to a smaller resolution. A patch was extracted from this downscaled image. The total number of patch tokens was calculated. The extracted patch was concatenated with the output of the XCA block in the XCiT model.

2. Dimension Re-factoring: The model's internal hyperparameters corresponding to number of patches along height and width were adjusted to accommodate the patch size.

This was done for two patch size, 4 and 16.

3.4 Loss Modifications

3.4.1 Modifications within TRADES loss

We explore two potential modifications to the KL divergence component of the TRADES loss function, aiming to enhance model robustness against adversarial attacks.

The original TRADES formulation typically computes the KL divergence between a clean data point's prediction (c) and its corresponding adversarial example's prediction (c'). This encourages the model to produce similar outputs for the clean and adversarial versions of the same data point.

$$\text{Original TRADES Loss: } L_{TRADES} = L_{natural}(c, y) + \lambda L_{robust}(c, c')$$

1. Pairwise KL with Class-Specific Adversarials

   This modification proposes calculating KL divergence between a clean data point and all adversarial examples belonging to the same class as the clean data point's adversarial counterpart.

   This modification aims to potentially encourage the model to produce predictions that are more robust not just to its own adversarial example, but also to adversarial examples within its own predicted class. This could lead to improved model discrimination under adversarial attacks.

   We have the following formulation for this:

$$L_I = \sum_{i=1}^{B} \sum_{j=1}^{B} \mathrm{KL}(c_{i,j}, \widetilde{c'_{i,j}}) \cdot I[y_i == y_j]$$

2. Mean KL Divergence with Class-Level Aggregation

   This approach suggests calculating the KL divergence between a class's clean prediction (c_i) and its adversarial counterpart (c'_i), for each class (i). Additionally, it proposes two strategies for further calculations:

   (a) We compute KL divergence between predictions of different classes. Then, it subtracts these inter-class KL divergences from the sum of intra-class KL divergences. This might incentivize the model to maintain distinct representations for different classes even under adversarial perturbations.

   We have the following formulation for this:

$$L_{II} = \sum_{i=1}^{C} \mathrm{KL}(c_i, c'_i) - \sum_{i=1}^{C} \sum_{j=i, j\neq i}^{C} \mathrm{KL}(c_i, c_j)$$

(b) Similar to the first strategy, it calculates intra-class KL divergences. However, it subtracts the KL divergence between a class's clean prediction (c_i) and the average adversarial prediction across all other classes ($\overline{c'}_i$). This might push the clean predictions away from the average behavior of other classes under adversarial attacks, potentially leading to improved class separation.

We have the following formulation for this:

$$L_{III} = \sum_{i=1}^{C} \mathrm{KL}(c_i, c'_i) - \sum_{i=1}^{C} \mathrm{KL}(c_i, \overline{c'}_i)$$

3.4.2 Cluster Loss

We explore a novel strategy to enhance the robustness of Vision Transformers (ViTs) by introducing a new loss function called "Cluster Loss". This loss function leverages Kullback-Leibler (KL) divergence to encourage the model to produce predictions that cluster around pre-defined anchor points for each class, promoting resilience against adversarial attacks.

Algorithm 1 Cluster Loss Computation

Input: Training data (x, y), number of classes C, model f

Preprocessing:

1. Compute mean feature vector μ_c for each class $c \in \{1, \ldots, C\}$ from the training data.
2. Sample k anchor feature vectors $z_{c,k}$ for each class c from the training data. These k data points in each class are closest to the corresponding mean vector (μ_c).
3. For each anchor $z_{c,k}$, find the corresponding data point $x_{c,k}$ in the training data.

Forward Pass:

1. For a data point x_i and its label y_i:
   (a) Obtain the model prediction $p_i = \text{softmax}(f(x_i))$.
   (b) Compute similarity vector $s_i = [s_{i,c} | c \in \{1, \ldots, C\}]$:

$$s_{i,c} = p_i \cdot \text{softmax}(f(x_{c,k}))$$

   for all anchor points k of class c.
2. Repeat steps above for the corresponding adversarial example x_i^{adv}.
3. Compute the KL divergence loss between similarity vectors s_i and s_i^{adv}:

$$L_{cluster} = \text{KL}(s_i || s_i^{adv})$$

Output: Cluster loss $L_{cluster}$

Experimentation Strategies

Several variations of the Cluster Loss were explored:

1. Vanilla Cluster Loss: The basic implementation with statically defined mean and anchor features throughout training.
2. Updating per Epoch: Anchor features are updated each epoch to reflect the evolving class distribution learned by the model.
3. Multi-Sampled Cluster Loss: Utilizes multiple anchor points per class for a more robust representation of the class cluster.
4. Temperature regulation: Softmax predictions of anchor labels are computed with a temperature parameter, promoting learning of broader class representations.

5. Cluster Logits: Employs logits directly instead of softmax for anchor labels, potentially capturing finer-grained information.

Chapter 4

Experimental Setup

4.0.1 Baseline

To establish a baseline for comparison, we included two ViT variants:

1. Vanilla ViT: We utilized the standard Vision Transformer model with a base architecture of ViT-B-16 as one of our baselines. This baseline represents the conventional ViT model without any specific adversarial training or robustness enhancements.

2. Robustly Trained XCiT: As another baseline, we employed the XCiT-S-12 model that has been robustly trained [3]. This model serves as a benchmark for robustness in the ViT family and helps us assess the effectiveness of our proposed adversarial training method.

4.0.2 Dataset

Most tests were conducted on CIFAR-10 for faster results. CIFAR-100 was also utilized for a broader evaluation of model performance.

4.0.3 Evaluation

The evaluations were performed using two widely recognized adversarial attack techniques:

1. PGD: Projected Gradient Descent is a white-box attack that iteratively perturbs the input data in the direction that maximizes the loss function, while constraining the perturbations to be within a certain epsilon bound. The PGD attack is considered to be more powerful than other attacks like FGSM (Fast Gradient Sign Method) because it performs multiple iterations and takes small steps to find the optimal perturbation [8].

   We employed the Projected Gradient Descent (PGD) attack with 10 steps and an ϵ= 8/255.

2. APGD: We also utilized the Auto Projected Gradient Descent attack. The step size is adapted based on the optimization progress, and once reduced, the maximization restarts from the best point found so far. This algorithm is different from usual PGD in terms of the choice of step size across iterations and the adaptive reduction of the step size [2].

   APGD-CE stands for Auto-PGD with Cross-Entropy loss. Here, the loss function used is Cross-Entropy loss.

3. Auto Attack: AutoAttack is an ensemble of diverse parameter-free attacks. It combines different attacks, including APGD-CE, APGD-DLR (Auto-PGD with Targeted-DLR loss), FAB, and Square Attack. [2]

Chapter 5

Results

5.0.1 Architecture and Training Modifications

To strengthen the generalizability of our observations, we conducted experiments on the CIFAR-100 dataset. Here are the results of experimental modifications on the XCiT-S-12 model. Here, robust accuracy is evaluated on APGD-CE attack.

Table 5.1: Experiments on CIFAR100

Modifications	Clean Accuracy	Robust Accuarcy
Benign	67.34%	37.05%
EMA Training	**70.52**%	**37.55**%
EMA Modification	70.16%	36%
Cross attention	65.91%	32.92%

The experimental modifications yielded a promising 0.5% increase in robust accuracy, showcasing progress.

The results on CIFAR-100 mirrored the trends we observed on CIFAR-10, suggesting that EMA training shows promising results.

As we can see, EMA training and EMA modification show promising improvements.

5.0.2 Input Modifications

Following are the results of performing input modifcations on the XCiT-S-12 using CIFAR10 dataset. Robust accuracy is evaluated using APGD-CE attack.

Our initial attempts at incorporating an additional image patch involved resizing the input image to a very small scale (4x4 pixels) and extracting a 4x4 patch. This is because ViT considers patches of size 4x4 for CIFAR10 dataset.

While this approach aimed to capture a specific detail, the significant downsizing likely resulted in a loss of crucial visual information necessary for accurate classification. Consequently, this method did not yield positive results.

Table 5.2: Image as Patch

Modifications	Clean Accuracy		Robust Accuracy
Benign	90.06%		58.62%
Vanilla approach	Patch size = 4	78.85%	40.31%
Dimension Re-factoring	Patch size = 4	76.65%	38.06%
	Patch size = 16	89.44%	54.43%
Ensemble EMA Training + Dimension Re-factoring	Patch size = 16	89.26%	55.92%

We then investigated using a larger image size of size 16x16 and a corresponding 16x16 patch. This improved upon the smaller patch but it did not outperform the baseline XCiT model, however, it highlights the importance of considering both the patch size and its integration within the architecture.

5.0.3 Loss Modifcations

Here, our goal was to see if adjustments to the loss function could improve performance against adversarial samples.

Modifying TRADES loss

The following table illustrates the result of modifying TRADES loss on the XCiT-S-12 using CIFAR10 dataset using PGD attack.

Table 5.3: Modifying TRADES loss

Modifications	β	Clean Accuracy	Robust Accuracy
Benign		90.06%	64.12%
L_I	6	91.71%	57.24%
	12	91.89%	57.11%
	3	91.06%	57.93%
L_{II}	2	90.91%	63.70%
L_{III}	2	90.90%	63.68%

Our experiments with modified TRADES loss did not yield significant improvements in model performance compared to the baseline XCiT trained with the standard TRADES loss. Interestingly, the modifications also did not lead to any degradation in performance. Upon further analysis, we discovered that the modified terms in the loss function often converged to near-zero values during the training process.

Cluster Loss

The cluster loss was evaluated in two configurations:

Combined with KL Divergence: Here, the cluster loss was incorporated alongside the KL divergence term from the TRADES loss function, aiming for a combination of robust loss and information divergence. Cluster Loss Alone: We also investigated the performance of the cluster loss as a standalone loss function, independent of the KL divergence term.

We experimented with different values for the weight (β) assigned to the cluster loss within the overall loss function. This involved plotting the ratio between various losses, such as natural loss, KL divergence loss, and the cluster loss itself.

Table 5.4: Cluster Loss

Modification	β	Clean Accuracy	AA RA	PGD RA	KL loss term present?
Benign		90.06%	56.14%	64.12%	
Vanilla Cluster Loss	**2**	**90.32%**	**55.51%**	**63.82%**	**Yes**
	2	90.17%	54.76%	62.43%	Yes
	6	91.54%		61.86%	No
	8	90.69%		61.59%	No
Updating per Epoch	6	90.71%		62.19%	No
Temperature regulation	6	94.25%		48.71%	No
Cluster Logits	6	89.61%		59.10%	No

Combining with KL Divergence led to a decrease of 0.63% in auto-attack accuracy compared to the baseline model, paired with an increase in clean accuracy by 0.26%. On the other hand, using the cluster loss alone yielded a more significant decrease in auto-attack accuracy of 1.38% compared to the baseline.

5.0.4 Comparison with SOTA

The evaluation involved comparing the model's performance with state-of-the-art robustly trained models using APGD-CE is given in Table 5.5.

Table 5.5: The notation "(b)" represents the baseline model's performance, * represents the model with EMA training, while ** represents the ensemble of the model with EMA training and EMA modification.

Paper/Model	Clean Accuracy	Robust Accuracy	Conference
Qin et al. [11]	86.28%	55.70%	NeurIPS19
Kim & Wang [6]	91.51%	56.64%	-
Hendrycks et al. [5]	87.11%	57.23%	ICML19
XCiT (b)	90.06%	58.62%	SaTML23
Wang & Zhang [15]	92.80%	59.09%	ICCV19
XCiT *	**90.39%**	**59.23%**	Ours
XCiT **	**90.40%**	**59.47%**	Ours

We also demonstarte our performance on Auto Attack as a benchmark.

Table 5.6: The notation "(b)" represents the baseline model's performance, * represents the model trained with cluster loss on CIFAR-10 checkpoint, while ** represents the model trained with cluster loss on ImageNet checkpoint.

Paper/Model	Clean Accuracy	Robust Accuracy	Conference
Vanilla ViT [7]	99.03%	0%	ICLR 2021
Resnet-50	84.80%	41.56%	-
Hendrycks et al. [5]	87.11%	54.92%	ICML19
XCiT (b)	90.06%	56.14%	SaTML23
XCiT *	**90.32%**	**55.51%**	Ours
XCiT **	**90.17%**	**54.76%**	Ours

RobustBench, a comprehensive benchmarking platform, has emerged as a critical tool for systematically tracking the real progress in adversarial robustness. This platform rigorously evaluates the performance of various models under a wide range of adversarial conditions, ensuring that reported improvements are truly reflective of genuine progress.

Our work on adversarial robustness ranks close to the top submissions made on the transformer architecture, a testament to the effectiveness of our approach in this domain.

5.0.5 Conclusion

We observed that cluster loss did not lead to a consistent improvement in performance across all metrics compared to the baseline model.

However, it's important to note that the cluster loss alone achieved good results when compared to some state-of-the-art (SOTA) models in terms of adversarial robustness. This suggests the potential of this approach, particularly for tasks where achieving strong robustness is crucial.

While there have been notable improvements in performance, they are not substantial.

Overall, our exploration provided valuable insights into potential avenues for enhancing XCiT. While none of the specific approaches yielded a definitive performance boost, the findings could guide future research directions.

Bibliography

[1] Bhojanapalli, S., Chakrabarti, A., Glasner, D., Li, D., Unterthiner, T., and Veit, A. Understanding robustness of transformers for image classification. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (10 2021), IEEE. [Online; accessed 2023-11-29].

[2] Croce, F., and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. https://arxiv.org/abs/2003.01690, mar 3 2020.

[3] Debenedetti, E., Sehwag, V., and Mittal, P. A light recipe to train robust vision transformers. In *2023 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)* (2 2023), IEEE. [Online; accessed 2023-11-29].

[4] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. https://arxiv.org/abs/2010.11929, oct 22 2020.

[5] Hendrycks, D., Lee, K., and Mazeika, M. Using Pre-Training Can Improve Model Robustness and Uncertainty. https://proceedings.mlr.press/v97/hendrycks19a.html, may 24 2019.

[6] Kim, J., and Wang, X. Sensible adversarial learning. https://openreview.net/forum?id=rJlf_RVKwr.

[7] Kolesnikov, A., Dosovitskiy, A., Weissenborn, D., Heigold, G., Uszkoreit, J., Beyer, L., Minderer, M., Dehghani, M., Houlsby, N., Gelly, S., Unterthiner, T., and Zhai, X. An image is worth 16x16 words: Transformers for image recognition at scale.

[8] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards Deep Learning Models Resistant to Adversarial Attacks. https://arxiv.org/abs/1706.06083, jun 19 2017.

[9] Mahmood, K., Mahmood, R., and van Dijk, M. On the Robustness of Vision Transformers to Adversarial Examples. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)* (10 2021), IEEE. [Online; accessed 2023-11-29].

[10] Mao, X., Qi, G., Chen, Y., Li, X., Duan, R., Ye, S., He, Y., and Xue, H. Towards Robust Vision Transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (6 2022), IEEE. [Online; accessed 2023-11-29].

[11] Qin, C., Martens, J., Gowal, S., Krishnan, D., Dvijotham, K., Fawzi, A., De, S., Stanforth, R., and Kohli, P. Adversarial Robustness through Local Linearization. *Advances in Neural Information Processing Systems 32*.

[12] Shao, R., Shi, Z., Yi, J., Chen, P.-Y., and Hsieh, C.-J. On the adversarial robustness of vision transformers. https://arxiv.org/abs/2103.15670, mar 29 2021.

[13] Tao, Zhuozhuo Tu, J. Z. D. Theoretical Analysis of Adversarial Learning: A Minimax Approach. [Online; accessed 2023-11-29].

[14] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need. https://arxiv.org/abs/1706.03762, jun 12 2017.

[15] Wang, J., and Zhang, H. Bilateral Adversarial Training: Towards Fast Training of More Robust Models Against Adversarial Attacks. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (10 2019), IEEE. [Online; accessed 2023-12-01].

[16] Zhang, H., Yu, Y., Jiao, J., Xing, E., Ghaoui, L. E., and Jordan, M. Theoretically principled trade-off between robustness and accuracy, 09–15 Jun 2019.