# Building Trustworthy Vision Transformers

**Areeb Gani**
CPSC 5710
Yale College (2027)
areeb.gani@yale.edu

**Rohan Phanse**
CPSC 5710
Yale College (2027)
rohan.phanse@yale.edu

**Vishak Srikanth**
CPSC 5710
Yale College (2027)
vishak.srikanth@yale.edu

## Abstract

Vision transformers are growing in popularity due to their powerful architecture and extensive applicability. In mission-critical systems, ensuring robustness to adversarial inputs becomes crucial for upholding safety and preventing catastrophic failures. In this project, we analyze the robustness properties of well-known vision transformers by evaluating the performance of several defense strategies with regards to an adversarial dataset. Our results demonstrate how different architectures respond to defensive paradigms, informing future work in the space on how to best make these vision transformers more robust.

## 1 Introduction

### 1.1 Problem Definition

Computer vision is one of the most fundamental use cases of artificial intelligence, with important applications across a variety of domains from healthcare to transportation to manufacturing. Because computer vision systems are so widespread, it is crucial to ensure safety in these visual systems and robustness against adversarial inputs.

One of the biggest innovations in this space has been the introduction of transformers, which have recently gained ground in the vision scene as well. Vision transformers (ViT) are useful with the advent of more data, since these architectures rely on fewer assumptions and can learn from large training datasets [13]. Thus, it is extremely important to evaluate the robustness of these models to ensure safety and trust in real-world applications, which often suffer from data drift.

In this project, we study how large, pretrained Vision Transformers respond to gradient-based and patch-based adversarial attacks on images drawn from ImageNet-1K, and how different defense strategies affect both vulnerability to these attacks and accuracy on clean images. Our adversarial dataset is constructed using canonical perturbation attacks such as FGSM, as well as ViT-specific patch-based attacks, and we use it to evaluate DINOv2-based Vision Transformers and closely related variants. Building on this setup, we analyze three complementary defense paradigms: feature-level defenses such as robustness tokens and parameter-level defenses such as adversarial training, which we evaluate primarily against gradient-based attacks, and input-level patch-based transformations, which we evaluate specifically against patch-based attacks, in order to evaluate the extent to which these defenses are effective and what tradeoffs they induce in clean-image accuracy.

### 1.2 Relevance to Trustworthy Deep Learning

Vision Transformers are now widely deployed in settings where model failures carry real consequences, so understanding their behavior under adversarial perturbations is central to building trustworthy systems. Even small, engineered changes to an input can cause large shifts in model predictions, and existing work shows that ViTs can fail in ways that are qualitatively different from CNNs. Our project studies this reliability question directly: we construct a large adversarial dataset,

Preprint. Under review.

evaluate several ViT architectures under gradient- and patch-based attacks, and compare the robustness gains from various defense families. By examining how these defense strategies behave across model scales and attack strengths, we aim to clarify which approaches meaningfully improve ViT robustness and where current methods still break down.

### 1.3 Datasets

In this project, we used the ImageNet-1k dataset[1], a subset of ImageNet [3] used in the ILSVRC 2012 challenge [11]. ImageNet-1k is an image classification dataset that spans 1,000 object classes and contains 1,281,167 training images, 50,000 validation images and 100,000 test images.

We conducted an exploratory data analysis on ImageNet-1K by analyzing the first 10,000 samples in the training dataset, which covers all 1,000 classes of ImageNet-1k (Appendix 7.3). We also constructed custom adversarial testing sets as detailed in Section 4.1.

## 2 Related Work

Our robustness evaluation follows standard adversarial benchmarks. FGSM [6] and PGD [9] remain the canonical gradient-based attacks used to probe model vulnerability, and PGD-based adversarial training [9] serves as the standard baseline defense. These methods provide the core attack and defense protocols we use to evaluate and compare robustness across Vision Transformer architectures in our study.

Vision Transformers (ViTs) represent a paradigm shift in computer vision, applying the self-attention mechanism from natural language processing to image analysis by treating images as sequences of patches rather than spatial grids. As the computer vision community embraced these architectures following their introduction by Dosovitskiy et al. [5], researchers began investigating how these fundamentally different models respond to adversarial attacks. Aldahdooh et al. [1] conducted the first systematic robustness comparison between Vision Transformers and CNNs, revealing that ViTs demonstrate superior resilience under various norm-bounded attacks and adaptive attacks, though preprocessing defenses like blurring and JPEG compression are less effective for ViTs than for CNNs. This finding motivated our focus on Vision Transformers rather than traditional CNNs, as it suggested that transformer architectures possess unique robustness properties worthy of deeper investigation. However, the emergence of large-scale, self-supervised Vision Transformers like DINOv3 [12] has created new challenges that existing robustness research has not adequately addressed. DINOv3's 7-billion parameter scale and self-supervised training methodology represent a significant departure from the smaller, supervised models previously studied, creating a critical gap in our understanding of modern ViT robustness.

Recent work has begun developing both ViT-specific attack and defense mechanisms that exploit the unique architectural properties of transformers. On the attack side, Liu et al. [8] demonstrated that Vision Transformers are vulnerable to attention-based adversarial patch attacks, where small patches covering only 1-3% of the input image can degrade model accuracy to 0% by manipulating attention mechanisms. Cools et al. [2] further showed that adversarial patches designed for CNNs transfer effectively to ViTs, with attack success rates ranging from 40.04% to 99.97% across different ViT architectures. Kashefi et al. [7] categorize ViT explainability methods into attention-based, pruning-based, and inherently explainable approaches. On the defense side, Pulfer et al. [10] introduced Robustness Tokens, a method that fine-tunes additional learnable tokens rather than the entire model, achieving significant robustness improvements with minimal computational overhead. Similarly, Doan et al. [4] pioneered patch processing defenses specifically for ViTs, demonstrating that the patch-based input representation can be leveraged through techniques like PatchDrop and PatchShuffle, which exploit the differential response of clean and backdoored inputs to patch manipulations before positional encoding. These transformer-native approaches represent a paradigm shift from traditional adversarial methods, offering both new attack vectors and computationally efficient defense alternatives particularly suited for large-scale models. Our project builds directly upon these innovations by systematically evaluating robustness mechanisms across different model scales and architectures, with particular emphasis on understanding their effectiveness when applied to modern large-scale transformers like DINOv3.

[1] https://huggingface.co/datasets/ILSVRC/imagenet-1k

# 3 Methods

## 3.1 Architecture and Implementation

Our system is organized around three main components: adversarial data generation, model defenses, and a centralized evaluation pipeline (Figure 1). We start from large pre-trained Vision Transformer models (e.g., DINOv2) trained on ImageNet. We assume white-box access to these models, which allows us to backpropagate through them and compute gradients for attack generation.

**Adversarial Data Generation.** Clean ImageNet samples are first passed through a modular attack suite consisting of gradient-based attacks and patch-based attacks. Each attack is implemented as a drop-in module that operates on batches of images and returns perturbed images constrained within an $\ell_\infty$ or $\ell_2$ ball (depending on the attack). To handle the memory constraints of working with ImageNet-scale data, we adopt a shard-based pipeline: we stream in shards of the dataset, generate adversarial examples for each shard on the cluster GPUs, upload the resulting adversarial batches to storage, and then free the shard from memory. The outputs these attacks comprised the adversarial dataset.

**Patch Based Attacks.** We implement a suite of localized patch-based attacks tailored to the patch-wise input structure of Vision Transformers, inspired respectively by adversarial token attacks, patch perturbation attacks, and the Patch-Fool framework. Using pretrained ViT-family models from Hugging Face, including a standard ViT, a distilled DeiT, and a DINOv2 checkpoint, we define a shared configuration that specifies the relative patch area, number of gradient steps, and step size, and use helper routines to compute a square patch size and sample patch locations on each correctly classified ImageNet-1K image. Given a batch of normalized pixel tensors and labels, we apply three related attack routines. The token attack learns an explicit adversarial patch tensor for each image: at every step it copies this patch into a sampled box on the image, runs the model, and updates the patch by taking sign-gradient steps on the cross-entropy loss before finally writing the optimized patch back into the image. The patch-perturbation attack instead treats the entire image as the optimization variable but applies a binary mask so that projected-gradient updates are confined to a single randomly chosen patch region while the rest of the image remains unchanged. The Patch-Fool-style attack also optimizes in pixel space, but at each iteration it scans a grid of non-overlapping patches, identifies the patch with the largest gradient norm, and applies a masked sign-gradient update only on that most sensitive patch. In all three cases, we clamp the perturbed images back to a valid pixel range after each update, producing localized adversarial examples in which only a small contiguous region has been modified. As a reference baseline for these experiments, we always evaluate each model on the corresponding clean, unpatched images from the same ImageNet-1K subset before applying any patch attacks.

**Defenses.** On top of the pre-trained ViTs, we implement three conceptually distinct defenses, as described below:

- **Robustness tokens:** a feature-level defense where we augment the ViT input sequence with learned robustness tokens. These tokens are trained to absorb or attenuate adversarial perturbations in the representation space, while keeping the original backbone weights largely intact.
- **Adversarial training:** a parameter-level defense where we fine-tune the ViT on adversarial examples. Our implementation focuses on FGSM to solve the inner-max problem, with an $\epsilon \approx 0.03$.
- **Image transformations (incl. patch-based defenses):** an input-level defense where we apply deterministic or stochastic transformations (e.g., Gaussian blurring, JPEG compression, and patch-based masking) to the input image prior to feeding it into the ViT. For global transformations, we apply Gaussian blur and JPEG compression by denormalizing model inputs to uint8 RGB images, running either a fixed-kernel Gaussian blur or a lower-quality JPEG encode–decode step, and then renormalizing the results back into the model's input space. To target localized adversarial behavior, we introduce patch masking defenses: for each image, we compute a patch size from a chosen patch ratio, sample a random patch box using the same routine as in our patch attacks, and either overwrite that region with

zeros in normalized space or replace it with a blurred version obtained by denormalizing, applying a small Gaussian blur, and renormalizing the patch. Together, these defenses allow us to compare how global image transformations and localized patch processing affect the robustness of ViT, DeiT, and DINOv2 models to the patch-based attacks described above.

Each defense produces a defended model variant, all of which share the same underlying architecture but differ in how they process or adapt to adversarial inputs. Note that we treat patch-based defenses separately, since that does not generate a new model (it is simply a pre-processing step).

**Evaluation Pipeline.** The defended and baseline models are evaluated on the same concatenated adversarial dataset, as well as on a held-out set of clean images. The evaluation module computes standard performance metrics (top-1 accuracy) alongside robustness-oriented diagnostics (e.g., attack success rate by $\epsilon$) and sensitivity analyses. For explainability, we extract attention maps and related internal signals from the ViTs to study how different defenses alter the model's focus under attack. This centralized, standardized evaluation pipeline is key to making the comparison between defenses trustworthy and reproducible.

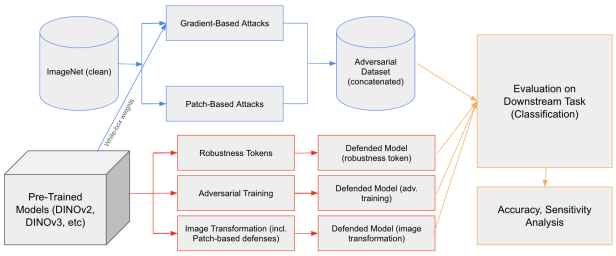


Figure 1: Workflow showing how our adversarial dataset is generated, how our models are defended, and our evaluation pipeline.

# 4 Results

## 4.1 Dataset Generation

To construct our adversarial dataset, we developed modular code that supports gradient-based and patch-based attacks [2]. The main challenge was GPU memory: the cluster GPUs cannot simultaneously store large portions of ImageNet, generate adversarial examples, and buffer them for upload. To address this, we implemented a shard-based pipeline. We download a shard of ImageNet, run our attacks, upload the adversarial examples once we reach a threshold, then delete the shard and proceed to the next one. This yields an efficient, streaming-style pipeline for adversarial data generation.

We also incorporated several optimizations. Our dataset includes FGSM examples for a range of $\epsilon$ values ($\epsilon \in \{0.1, 0.3, 0.5\}$). For some values of $\epsilon$, an adversarial example may fail to flip the model's prediction; in such cases, we do *not* include the example in the adversarial dataset, since it does not meaningfully test robustness. Similarly, we discard base images that are already misclassified, as our focus is on how the model adapts to genuinely adversarial perturbations rather than to clean errors.

Our final dataset is hosted on HuggingFace[3], which makes it easy for group members to run parallel experiments with different model defenses. In the current version (generated with 100k examples), we achieve an overall attack success rate of 91.2%. Interestingly, under our pre-filtering conditions, the success rate *decreases* as we increase $\epsilon$. This is counterintuitive relative to standard FGSM behavior, but there are a few plausible explanations. First, DINOv2 may be inherently robust as a Vision Transformer, so larger perturbations at higher $\epsilon$ values become visually obvious and are more easily rejected by the model. Second, our clamping procedure may cause larger gradient steps to saturate pixel values, effectively reducing the impact of the perturbation compared to smaller, more nuanced $\epsilon$ steps.

[2]Preliminary data loading code: https://github.com/areebg9/cpsc4710-final/tree/fgsm
[3]Huggingface repository: https://huggingface.co/cpsc-5710-final-vit-robustness

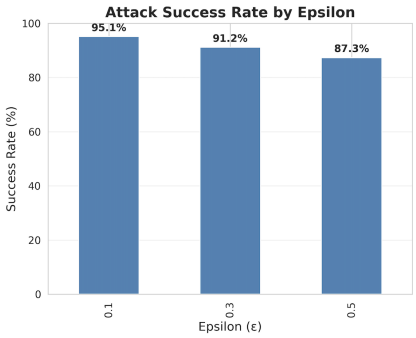


Figure 2: Attack success rate by epsilon ($\epsilon$). Note that our success rate reduces as $\epsilon$ value increases.

Below, we show an example where an attack with a smaller $\epsilon = 0.1$ successfully flips the prediction, while a larger $\epsilon = 0.5$ does not.

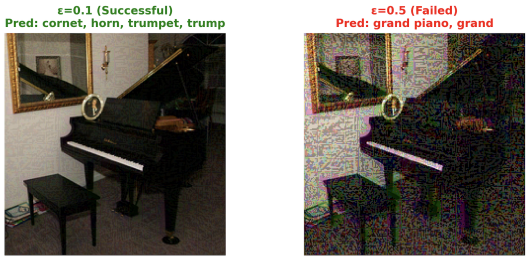


Figure 3: Example of $\epsilon$-scaled attacks. The larger $\epsilon$ introduces more visible noise, potentially making the perturbation easier for the model to reject as adversarial.

### 4.2 Adversarial Training Results

Our first defense is using adversarial training. We conduct an ablation study for adversarial training on FGSM-generated examples by comparing three configurations: a non-robust baseline (no adversarial training), an adversarially trained model where we unfreeze only the last 4 transformer blocks, and a fully unfrozen adversarially trained model. In the "Unfreeze (4 blocks)" setting, we update 7.9M out of 22.8M parameters ($\sim$34.6%), whereas in "Unfreeze (all)" we fine-tune the entire ViT. In this experiment, we finetune on top of `facebook/dinov2-small-imagenet1k-1-layer`(Table 1).

| Attack | Baseline | Unfreeze (4 blocks) | Unfreeze (all) |
|---|---|---|---|
| ID | 80.2 | 68.8 | 69.1 |
| FGSM | 3.3 | 20.6 | 45.8 |
| PGD | 0.0 | 0.8 | 43.0 |

Table 1: Adversarial training results for DINOv2. Numbers represent percent accuracy scores.

The baseline model is effectively non-robust; FGSM accuracy drops to 80.2% on clean (ID) data to 3.3%, and PGD is strong enough to drive accuracy to 0%. Adversarially training while unfreezing only the last 4 blocks buys some robustness (FGSM accuracy improves to 20.6%, PGD to 0.8%), but the model remains extremely fragile and looses general accuracy. This suggests that updating a thin slice of the network (at least, the final layers of this ViT) is not enough to substantially reshape the decision boundary.

Fully unfreezing the backbome during FGSM-based adversarial training leads to much larger gains. FGSM accuracy increases from 3.3% to 45.8%, and PGD accuracy from 0.0% to 43.0%, while clean accuracy only drops from 80.2% to 69.1%. Intuitively, training on FGSM examples encourages the model to smooth its decision boundary in feature space along the directions of largest loss gradients.

Since PGD is a stronger, multi-step version of FGSM that follows similar gradient directions, this local smoothing transfers to PGD as well, so robustness improves on both attacks even though we only train on FGSM.

Overall, these results show that (i) without adversarial training, small and targeted perturbations can almost completely break a high-accuracy DINOv2 classifier, and (ii) achieving non-trivial robustness requires updating at least a substantial fraction of the backbone parameters, not just a small classification head or a few blocks, even at the cost of a modest drop in clean accuracy.

### 4.3 Robustness Token Results

We conducted a series of robustness token training and evaluation runs for DINOv2 and DINOv3 models[4]. These runs were made by modifying the code from the original study to conform to our experiment parameters. Further training details such as loss curves and hyperparameters are included in Appendix 7.2.

We conducted an ablation study to investigate three factors: 1) the effect of the number of robustness tokens, 2) comparisons between DINOv2 models of various sizes, and 3) the benefit of using a custom linear head.

To study the impact of the number of robustness tokens, we trained DINOv2-ViT-Base/14 with 1, 5, 10, and 20 robustness tokens, respectively. To assess how robustness token effectiveness varies with model scale, we also trained DINOv2-ViT-Small/14 and DINOv2-ViT-Large/14 models with 10 robustness tokens. The corresponding training loss curves are presented in Figure 7.

The total training loss is defined by Pulfer at al. [10] as $\mathcal{L} = \mathcal{L}_{inv} + \mathcal{L}_{adv}$, where the invariance loss $\mathcal{L}_{inv}$ is the cosine similarity between the embeddings for the original and robust model on clean inputs, and the adversarial loss $\mathcal{L}_{adv}$ is the cosine similarity between the original model's embeddings for clean inputs and the robustness-token model's embeddings for adversarial inputs generated with Projected Gradient Descent (PGD). During training, the robustness tokens are optimized to maximize the loss (up to 2) to improve the backbone's invariance and robustness to adversarial inputs. As shown in Figure 7, DINOv2 models across various sizes (ViT-Small, ViT-Base, Vit-Large) converged to similarly high final loss values (1.76–1.82).

After training the robustness tokens, we trained a custom linear classification head for each backbone using 100k images from ImageNet-1k over 3 epochs, with robustness tokens present as register tokens. For comparison, we also trained "base" linear heads on the corresponding backbone models without robustness tokens, which serve as non-robust baselines. Because the robustness-token training objective explicitly encourages embedding similarity with the base model, we hypothesized that using the "base" linear head would remain effective, although a custom head should still yield the best performance. Accordingly, we evaluated both the base head and custom head for backbones with robustness tokens across all model sizes.

As shown in Table 2, robustness tokens consistently improve feature robustness and adversarial accuracy across all model sizes while largely preserving clean accuracy. Varying the number of robustness tokens has little effect on robustness, with robust classification accuracy remaining similar (42.3-46.4) among tested DINOv2-ViT-Base/14 models. Robustness and accuracy appears to scale with model size, with DINOv2-ViT-Large/14 achieving the strongest results overall. Finally, while custom linear heads consistently outperform base heads in robust accuracy, the gap is small, suggesting that robustness-token training preserves compatibility with the original embedding space.

### 4.4 Patch-Based Attacks and Defenses Results

We evaluate three gradient-based patch attacks (token, patch-perturbation, and `patch_fool`) together with image and patch-level preprocessing defenses on pretrained ViT, DeiT, and DINOv2 models. For each model, we first construct a set of images that are correctly classified under clean inputs with no defenses, so that subsequent accuracy values directly reflect robustness rather than baseline errors. Starting from these correctly classified, clean images for each model, we generate adversarial examples for each attack at a fixed patch ratio and number of gradient steps, and re-evaluate the same

[4]Our fork of Pulfer et al's [10] robustness token repository: https://github.com/rohanphanse/robustness-tokens

| Model | # Tokens | Feature Robustness | | Classification | |
|---|---|---|---|---|---|
| | | Cosine Similarity | MSE | Robust Acc. | Clean Acc. |
| DINOv2-ViTS 14 | 0 | 0.08 | 9.77 | 0.3 | 75.6 |
| DINOv2-ViTS 14-base-head | 10 | **0.93** | 0.63 | 26.3 | 73.7 |
| DINOv2-ViTS/14 | 10 | **0.93** | 0.63 | 30.2 | 74.3 |
| DINOv2-ViTB/14 | 0 | 0.04 | 5.06 | 0.8 | 80.1 |
| DINOv2-ViTB/14 | 1 | 0.89 | 0.53 | 42.3 | 79.3 |
| DINOv2-ViTB/14 | 5 | 0.91 | 0.43 | 46.4 | 79.7 |
| DINOv2-ViTB/14-base-head | 10 | 0.92 | 0.41 | 42.5 | 79.7 |
| DINOv2-ViTB/14 | 10 | 0.92 | 0.41 | 45.2 | 79.2 |
| DINOv2-ViTB/14 | 20 | 0.92 | 0.38 | 46.6 | 79.6 |
| DINOv2-ViTL/14 | 0 | 0.06 | 3.77 | 1.5 | **82.5** |
| DINOv2-ViTL/14-base-head | 10 | 0.90 | **0.34** | 58.7 | 82.3 |
| DINOv2-ViTL/14 | 10 | 0.90 | **0.34** | **60.4** | 82.4 |

Table 2: Results from our ablation study, modifying the number of trainable robustness tokens and the type of classification head (base vs. custom). Attacks were generated with PGD upon the base non-robust model. Evaluation was performed upon a test set of 5,005 images from ImageNet-1k. Cosine similarity and mean squared error (MSE) are computed between the base (non-robust) model's embeddings on clean inputs and the tested's embeddings model on adversarial inputs.

model on the attacked images under four defenses: none, Gaussian blur, JPEG compression, patch masking with blur, and patch masking with zero fill.

#### 4.4.1 Metrics and Evaluation

For each model–attack–defense combination, we report the accuracy and the induced attack success rate (ASR), defined on the filtered set as ASR $= 1 -$ accuracy. The clean, undefended configuration on this filtered set achieves an accuracy of $1.00$ for all three models because we constructed the set of images such that the clean, undefended model classifies all of these images correctly. Therefore, any reduction in accuracy directly measures robustness degradation due to the presence of an attack or images being missclassified due to a defense transformation. Entries in Table 4 are sorted by model, then attack, then defense in a fixed order (attacks: Clean, token, patch_perturbation, `patch_fool` and defenses: None, blur, compress, patch_mask_blur, patch_mask_zero).

The effects of various attack and defense combinations on model accuracy and ASR are shown in Figures 4 and 5 respectively.

#### 4.4.2 Results

Across all three models, `patch_fool` is the strongest attack: with no defenses, accuracy drops from $1.000$ on the clean-correct set to $0.576$ on DeiT, $0.340$ on DINOv2, and $0.292$ on ViT (ASR $0.424$, $0.660$, and $0.708$), substantially worse than the token and patch-perturbation attacks in the same setting. Standard input-level defenses are highly effective at countering these attacks: Gaussian blur and JPEG compression consistently recover a large portion of the lost accuracy, especially for `patch_fool` (for example, on DINOv2, `patch_fool` accuracy increases from $0.340$ to $0.780$ with blur and $0.916$ with compression, and on ViT from $0.292$ to $0.616$ and $0.632$), while still maintaining high clean accuracy around $0.95$–$0.97$. In contrast, the local patch masking defenses leave clean accuracy almost unchanged but provide only modest robustness gains against `patch_fool` (e.g., DINOv2 remains near $0.35$ accuracy and ViT below $0.35$ under masking), indicating that simple global transformations such as blur and compression are more effective than blind patch masking for mitigating gradient-based patch attacks on Vision Transformers.

These trends align with how attacks and defenses interact with Vision Transformer self-attention. The `patch_fool` attack explicitly optimizes a localized patch to hijack attention, creating a small but highly salient region that dominates global self-attention and overwrites evidence from the rest of the image, whereas token and patch-perturbation attacks produce more diffuse, less attention-aligned perturbations that the model can partially ignore. Gaussian blur and JPEG compression act as global low-pass filters that smear or denoise the high-frequency patterns introduced by such localized patches, reducing their ability to monopolize attention and shifting weight back to the clean context,

while patch masking is weaker because it removes only a single random patch and often misses the true adversarial region.

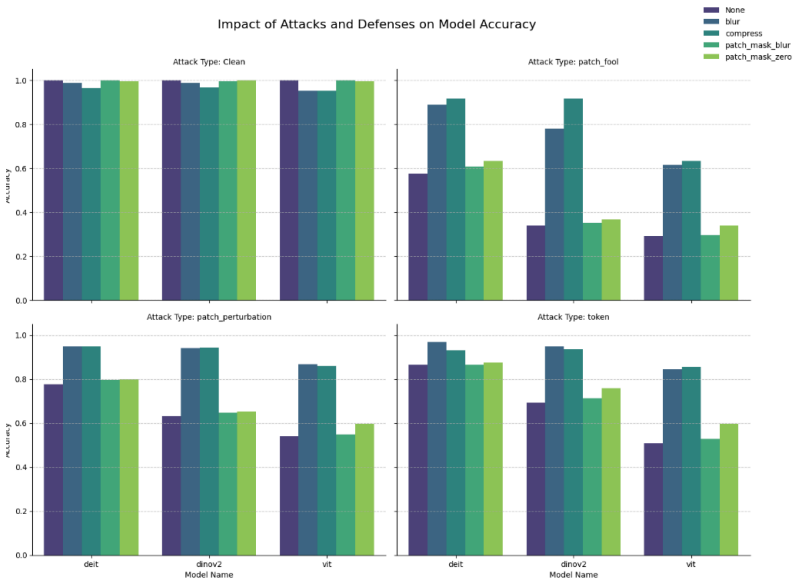


Figure 4: Impact of Attacks and Defenses on Model Accuracy


Figure 5: Impact of Attacks and Defenses on ASR

### 4.5 Overall Evaluation

Across all models, we treat the undefended DINOv2 model as our baseline: accuracy stays near 30% for every $\epsilon$ value, reflecting almost no adversarial robustness. Both adversarially trained variants perform well at small perturbation strengths (above 80% at $\epsilon = 0.1$), but their performance drops sharply as $\epsilon$ increases, falling into the mid–30% range by $\epsilon = 0.5$. Notably, unfreezing only the last four blocks behaves very similarly to full-model fine-tuning—slightly worse at low $\epsilon$, slightly better at higher $\epsilon$, suggesting that partial fine-tuning already captures most of the gains for this FGSM-only attack setting.

The robustness-token models show a different pattern. While they start below adversarial training at $\epsilon = 0.1$, they degrade much more slowly as the perturbation grows. ViT-B with robustness tokens remains between 50–70% across all $\epsilon$, and ViT-L is consistently the strongest model at higher attack strengths, reaching 71.5% at $\epsilon = 0.3$ and 65.5% at $\epsilon = 0.5$. Within the robustness-token family, performance scales cleanly with backbone size (ViT-L $>$ ViT-B $>$ ViT-S). Overall, adversarial training gives the best robustness for small perturbations, whereas robustness tokens (especially on larger backbones) offer more stable performance as attack strength increases.

| FGSM $\epsilon$ | Baseline | Adv. Train (4 blocks) | Adv. Train (all) | RT ViT-S | RT ViT-B | RT ViT-L |
|---|---|---|---|---|---|---|
| 0.1 | 32.04 | 80.84 | 82.93 | 52.99 | 70.06 | 79.94 |
| 0.3 | 27.33 | 63.96 | 62.16 | 44.74 | 59.16 | 71.47 |
| 0.5 | 30.33 | 35.74 | 34.23 | 43.24 | 53.15 | 65.47 |

Table 3: FGSM accuracy (percent) across models and attack strengths on our adversarial dataset.

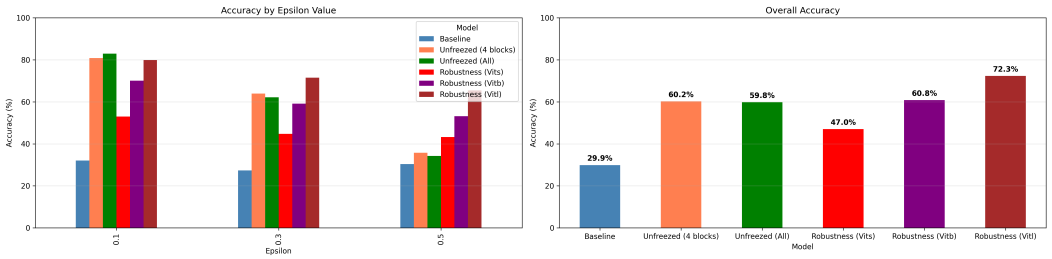


Figure 6: Example of $\epsilon$-scaled attacks. The larger $\epsilon$ introduces more visible noise, potentially making the perturbation easier for the model to reject as adversarial.

# 5 Conclusion

## 5.1 Summary and Findings

Overall, our experiments confirm that standard pre-trained ViTs are highly vulnerable to both gradient-based and patch-based attacks: the undefended DINOv2 baseline collapses to roughly 30% accuracy under FGSM across all $\epsilon$ values, and patch-based attach such as `patch_fool` can drive accuracy below 35% on ViT, DeiT, and DINOv2 despite perfect performance on the same images in the clean setting. Adversarial training does help, but only once a substantial fraction of the backbone is unfrozen: updating the last four transformer blocks already yields noticeable robustness gains, while fully unfreezing the model produces much larger improvements under FGSM and PGD, at the cost of a moderate drop in clean accuracy.

Robustness tokens offer a more parameter-efficient alternative. Across all tested DINOv2 sizes, adding tokens markedly improves robustness and feature invariance while keeping clean accuracy close to the original models, and performance scales cleanly with backbone size, with ViT-L consistently the strongest under stronger attacks. For patch-based threates, simple global transforms such as Gaussian blur and JPEG compression and surprisingly effective at recovering accuracy from `patch_fool`-style attacks across all architectures, whereas blind patch masking provides only limited gains and sometimes even hurts in specific settings. Taken together, these results suggest that (i) ViTs need explicit defenses to be reliable under adversarial perturbations, (ii) both adversarial training and robustness tokens can improve robustness but operate in different regimes (e.g. small v. larger perturbations), and (iii) inexpensive input-level defenses such as patch-based defenses can prove to be a strong complement to these strategies.

Our code can be found here: `https://github.com/areebg9/cpsc4710-final`. Our Hugging Face can be found here: `https://huggingface.co/cpsc-5710-final-vit-robustness`.

### 5.2 Challenges and Limitations

Currently, our dataset is limited to FGSM-based attacks. Furthermore, our adversarial training is performed with FGSM for the inner-max problem; this is more efficient in terms of time, but leveraging a stronger attack such as PGD in these cases could yield stronger results.

Our evaluations also focus primarily on accuracy. With more time for this project, we would hope to incorporated more fine-grained analyses such as attention-map inspection or layer-wise feature drift, which limits our understanding of *why* certain defenses work or fail.

Finally, some phenomena we observe (such as decreasing FGSM success rate at higher $\epsilon$ under our filtering rules) are not fully explained and highlight open questions about the interaction between model robustness and attack strength.

## 6 Individual Contributions

### 6.1 Areeb Gani

I worked on the construction of the adversarial dataset, including implementing the shard-based generation pipeline and running all FGSM attacks at scale. I also implemented and ran the adversarial training experiments and produced the corresponding robustness evaluations. Finally, I developed the centralized evaluation pipeline and used it to benchmark the defended models under consistent metrics and attack settings. Along with my teammates, I also contributed to the writing of the paper as well.

### 6.2 Rohan Phanse

I worked on the robustness token experiments, updating Pulfer et al.'s [10] repository to improve support for DINOv2, add support for DINOv3, and write the code necessary to conduct all of the robustness token ablation studies and experiments in our project. I also contributed to the writing of the paper as well.

### 6.3 Vishak Srikanth

I designed the framework for dataset generation for the patch-based attacks and patch-based defenses portions, ran extensive experiments for the attack-defense combinations as well as discussions and conclusions sections and contributed to the writing of the paper.

## References


[1] Amin Aldahdooh, Wassim Hamidouche, and Olivier Deforges. Reveal of vision transformers robustness against adversarial attacks, 2021. URL `https://arxiv.org/abs/2106.03734`.

[2] Lucas Cools, Simon Geirnaert, and Tim Van hamme. Vision transformers: the threat of realistic adversarial patches. *arXiv preprint arXiv:2509.21084*, 2024.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[4] Khoa Doan, Yingjie Liu, Khalil Hameed, Shengshan Hu, Rui Zhang, Dawei Jin, Zhengming Chen, and Anh Nguyen. Defending backdoor attacks on vision transformer via patch processing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 498–506, 2023. doi: 10.1609/aaai.v37i1.25125.

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=YicbFdNTTy`.

[6] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015. URL https://arxiv.org/abs/1412.6572.

[7] Rojina Kashefi, Leili Barekatain, Mohammad Sabokrou, and Fatemeh Aghaeipoor. Explainability of vision transformers: A comprehensive review and new perspectives. *arXiv preprint arXiv:2311.06786*, 2023.

[8] Liang Liu, Yanan Guo, Youtao Zhang, and Jun Yang. Understanding and defending patched-based adversarial attacks for vision transformer. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pages 21631–21657. PMLR, 2023.

[9] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019. URL https://arxiv.org/abs/1706.06083.

[10] Brian Pulfer, Yury Belousov, and Slava Voloshynovskiy. Robustness tokens: Towards adversarial robustness of transformers. In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part LIX*, page 110–127, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-73201-0. doi: 10.1007/978-3-031-73202-7_7. URL https://doi.org/10.1007/978-3-031-73202-7_7.

[11] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015. URL https://arxiv.org/abs/1409.0575.

[12] Oriane Siméoni, Huy V. Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, Francisco Massa, Daniel Haziza, Luca Wehrstedt, Jianyuan Wang, Timothée Darcet, Théo Moutakanni, Leonel Sentana, Claire Roberts, Andrea Vedaldi, Jamie Tolan, John Brandt, Camille Couprie, Julien Mairal, Hervé Jégou, Patrick Labatut, and Piotr Bojanowski. Dinov3, 2025. URL https://arxiv.org/abs/2508.10104.

[13] Tobias van der Werff. CNN vs. Vision Transformer: A Practitioner's Guide to Selecting the Right Model — tobiasvanderwerff.com. https://tobiasvanderwerff.com/2024/05/15/cnn-vs-vit.html. [Accessed 22-10-2025].

# 7 Appendix

## 7.1 Reproducibility

Custom datasets and model weights for our project are provided in our Hugging Face repository: https://huggingface.co/cpsc-5710-final-vit-robustness.

### 7.1.1 Robustness Token Experiments

We provide our fork of Pulfer et al.'s [10] repository at https://github.com/rohanphanse/robustness-tokens. The training and evaluation results in Figure 7 and Table 2 can be reproduced by following the setup guide in `README.md` and then running `train.sh` and `eval.sh` respectively.

## 7.2 Additional Results

The training loss curves for DINOv2 and DINOv3 with robustness tokens are provided in Figure 7. DINOv2 models were trained with a batch size of 8, a learning rate of $10^{-3}$, and a maximum of 100 steps (see the configuration file used to train robustness tokens for DINOv2-ViTB/14 here). DINOv3 models were trained with a batch size of 8, a learning rate (LR) of $10^{-2}$ (with an LR-warmup of 50 steps for DINOv3-ViTB/16), and a maximum of 100 steps.


Figure 7: Training loss curves for DINOv2 and DINOv3 models trained with robustness tokens.

## 7.3 Dataset Analysis

We conducted an exploratory data analysis on ImageNet-1k by analyzing the first 10,000 samples in the training dataset. This subset covers all 1,000 classes of ImageNet-1k. The class frequency distribution is depicted in Figure 8. Specifically, the number of samples per class ranges from 3 to 22, with 10 samples per class on average and a standard deviation of 3.07. Examples from ImageNet-1K are visualized in Figure 9.

ImageNet-1K contains a highly diverse set of images varying in dimension, object scale, pose, background, lighting, and more, reflecting real-world variation. Specifically, the first 10,000 training images in ImageNet-1K have an average width and standard deviation of $473.2 \pm 201.6$ pixels, and an average height and standard deviation of $406.8 \pm 174.9$ pixels. This dataset poses complex challenges for prediction models, with non-trivial intra-class variability and inter-class similarity and a variety of image dimensions and resolutions covered.

Table 4: Patch-based robustness of DeiT, DINOv2, and ViT on the 250-image clean-correct evaluation set. Accuracy is computed only over images that are correctly classified under clean inputs with no attacks and no defenses. ASR denotes attack success rate ASR $= 1 -$ accuracy.

| Model | Attack Type | Defense Type | Accuracy | ASR |
|---|---|---|---|---|
| deit | Clean | None | 1.000 | 0.000 |
| deit | Clean | blur | 0.988 | 0.012 |
| deit | Clean | compress | 0.964 | 0.036 |
| deit | Clean | patch_mask_blur | 1.000 | 0.000 |
| deit | Clean | patch_mask_zero | 0.996 | 0.004 |
| deit | patch_fool | None | 0.576 | 0.424 |
| deit | patch_fool | blur | 0.888 | 0.112 |
| deit | patch_fool | compress | 0.916 | 0.084 |
| deit | patch_fool | patch_mask_blur | 0.608 | 0.392 |
| deit | patch_fool | patch_mask_zero | 0.632 | 0.368 |
| deit | patch_perturbation | None | 0.776 | 0.224 |
| deit | patch_perturbation | blur | 0.948 | 0.052 |
| deit | patch_perturbation | compress | 0.948 | 0.052 |
| deit | patch_perturbation | patch_mask_blur | 0.796 | 0.204 |
| deit | patch_perturbation | patch_mask_zero | 0.800 | 0.200 |
| deit | token | None | 0.864 | 0.136 |
| deit | token | blur | 0.968 | 0.032 |
| deit | token | compress | 0.932 | 0.068 |
| deit | token | patch_mask_blur | 0.864 | 0.136 |
| deit | token | patch_mask_zero | 0.876 | 0.124 |
| dinov2 | Clean | None | 1.000 | 0.000 |
| dinov2 | Clean | blur | 0.988 | 0.012 |
| dinov2 | Clean | compress | 0.968 | 0.032 |
| dinov2 | Clean | patch_mask_blur | 0.996 | 0.004 |
| dinov2 | Clean | patch_mask_zero | 1.000 | 0.000 |
| dinov2 | patch_fool | None | 0.340 | 0.660 |
| dinov2 | patch_fool | blur | 0.780 | 0.220 |
| dinov2 | patch_fool | compress | 0.916 | 0.084 |
| dinov2 | patch_fool | patch_mask_blur | 0.352 | 0.648 |
| dinov2 | patch_fool | patch_mask_zero | 0.368 | 0.632 |
| dinov2 | patch_perturbation | None | 0.632 | 0.368 |
| dinov2 | patch_perturbation | blur | 0.940 | 0.060 |
| dinov2 | patch_perturbation | compress | 0.944 | 0.056 |
| dinov2 | patch_perturbation | patch_mask_blur | 0.648 | 0.352 |
| dinov2 | patch_perturbation | patch_mask_zero | 0.652 | 0.348 |
| dinov2 | token | None | 0.692 | 0.308 |
| dinov2 | token | blur | 0.948 | 0.052 |
| dinov2 | token | compress | 0.936 | 0.064 |
| dinov2 | token | patch_mask_blur | 0.712 | 0.288 |
| dinov2 | token | patch_mask_zero | 0.760 | 0.240 |
| vit | Clean | None | 1.000 | 0.000 |
| vit | Clean | blur | 0.952 | 0.048 |
| vit | Clean | compress | 0.952 | 0.048 |
| vit | Clean | patch_mask_blur | 1.000 | 0.000 |
| vit | Clean | patch_mask_zero | 0.996 | 0.004 |
| vit | patch_fool | None | 0.292 | 0.708 |
| vit | patch_fool | blur | 0.616 | 0.384 |
| vit | patch_fool | compress | 0.632 | 0.368 |
| vit | patch_fool | patch_mask_blur | 0.296 | 0.704 |
| vit | patch_fool | patch_mask_zero | 0.340 | 0.660 |
| vit | patch_perturbation | None | 0.540 | 0.460 |
| vit | patch_perturbation | blur | 0.868 | 0.132 |
| vit | patch_perturbation | compress | 0.860 | 0.140 |
| vit | patch_perturbation | patch_mask_blur | 0.548 | 0.452 |
| vit | patch_perturbation | patch_mask_zero | 0.596 | 0.404 |
| vit | token | None | 0.508 | 0.492 |
| vit | token | blur | 0.844 | 0.156 |
| vit | token | compress | 0.856 | 0.144 |
| vit | token | patch_mask_blur | 0.528 | 0.472 |
| vit | token | patch_mask_zero | 0.596 | 0.404 |

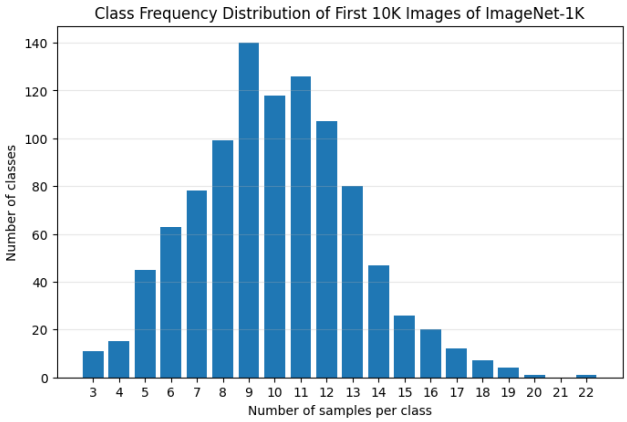


Figure 8: A histogram of the class frequency distribution of the first 10,000 training images of ImageNet-1k.


Figure 9: Visualizations of the first four training images in ImageNet-1k with their corresponding labels.