

# Understanding The Robustness in Vision Transformers

Daquan Zhou<sup>1\*</sup> Zhiding Yu<sup>2</sup> Enze Xie<sup>3</sup>  
Chaowei Xiao<sup>2,4</sup> Anima Anandkumar<sup>2,5</sup> Jiashi Feng<sup>6†</sup> Jose M. Alvarez<sup>2</sup>

## Abstract

Recent studies show that Vision Transformers (ViTs) exhibit strong robustness against various corruptions. Although this property is partly attributed to the self-attention mechanism, there is still a lack of systematic understanding. In this paper, we examine the role of self-attention in learning robust representations. Our study is motivated by the intriguing properties of the emerging visual grouping in Vision Transformers, which indicates that self-attention may promote robustness through improved mid-level representations. We further propose a family of fully attentional networks (FANs) that strengthen this capability by incorporating an attentional channel processing design. We validate the design comprehensively on various hierarchical backbones. Our model achieves a state-of-the-art 87.1% accuracy and 35.8% mCE on ImageNet-1k and ImageNet-C with 76.8M parameters. We also demonstrate state-of-the-art accuracy and robustness in two downstream tasks: semantic segmentation and object detection. Code will be available at <https://github.com/NVlabs/FAN>.

## 1. Introduction

Recent advances in visual recognition are marked by the rise of Vision Transformers (ViTs) (Dosovitskiy et al., 2020) as state-of-the-art models. Unlike ConvNets (LeCun et al., 1989; Krizhevsky et al., 2012) that use a “sliding window” strategy to process visual inputs, the initial ViTs feature a design that mimics the Transformers in natural language processing - An input image is first divided into a sequence of patches (tokens), followed by self-attention (SA) (Vaswani

\*Work done during an internship at NVIDIA.

†Work partially done during the affiliation with NUS.

<sup>1</sup>National University of Singapore <sup>2</sup>NVIDIA <sup>3</sup>The University of Hong Kong <sup>4</sup>ASU <sup>5</sup>Caltech <sup>6</sup>ByteDance. Correspondence to: Zhiding Yu <zhidingy@nvidia.com>.

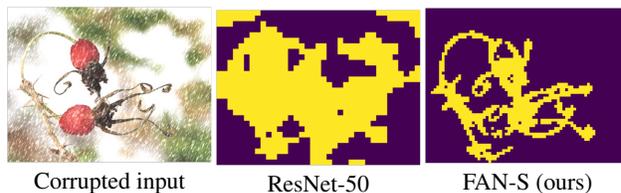
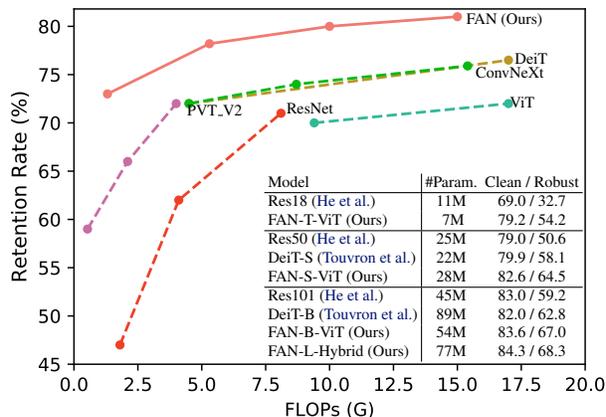
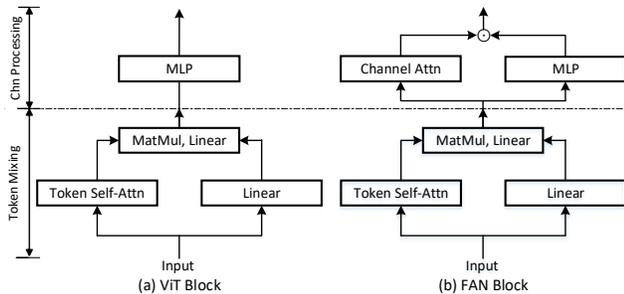


Figure 1. Main results on ImageNet-C (top figure) and clustering visualization (bottom row). Retention rate is defined as robust accuracy / clean accuracy. Left to right in bottom row: input image contaminated by corruption (snow) and the visualized clusters. Visualization is conducted on the output features (tokens) of the second last layers. All models are pretrained on ImageNet-1K. Input size is set to  $448 \times 448$  following (Caron et al., 2021).

et al., 2017) layers to aggregate the tokens and produce their representations. Since introduction, ViTs have achieved good performance in many visual recognition tasks.

Unlike ConvNets, ViTs incorporate the modeling of non-local relations using self-attention, giving it an advantage in several ways. An important one is the robustness against various corruptions. Unlike standard recognition tasks on clean images, several works show that ViTs consistently outperform ConvNets by significant margins on corruption robustness (Bai et al., 2021; Xie et al., 2021; Zhu et al., 2021; Paul & Chen, 2022; Naseer et al., 2021). The strong robustness in ViTs is partly attributed to their self-attention designs, but this hypothesis is recently challenged by an emerging work ConvNeXt (Liu et al., 2022), where a network constructed from standard ConvNet modules without



**Figure 2. Comparison between conventional ViT block and the proposed FAN block.** (a) ViT block: Input tokens are first aggregated by self-attention, followed by a linear projection and an MLP is appended to the self attention block for feature transformation. (b) FAN block: both token self-attention and channel attention are applied, which makes the entire network fully attentional. The linear projection layer after the channel attention is removed.

self-attention competes favorably against ViTs in generalization and robustness. This raises an interesting question on the actual role of self-attention in robust generalization.

**Our approach:** In this paper, we aim to find an answer to the above question. Our journey begins with the intriguing observation that meaningful segmentation of objects naturally emerge in ViTs during image classification (Caron et al., 2021). This motivates us to wonder whether self-attention promotes improved mid-level representations (and thus robustness) via visual grouping - a hypothesis that echoes the odyssey of early computer vision (U.C. Berkeley). As a further examination, we analyze the output tokens from each ViT layer using spectral clustering (Ng et al., 2002), where the significant<sup>1</sup> eigenvalues of the affinity matrix correspond to the main cluster components. Our study shows an interesting correlation between the number of significant eigenvalues and the perturbation from input corruptions: both of them decrease significantly over mid-level layers, which indicates the symbiosis of grouping and robustness over these layers.

To understand the underlying reason for the grouping phenomenon, we interpret SA from the perspective of information bottleneck (IB) (Tishby et al., 2000; Tishby & Zaslavsky, 2015), a compression process that “squeezes out” unimportant information by minimizing the mutual information between the latent feature representation and the target class labels, while maximizing mutual information between the latent features and the input raw data. We show that under mild assumptions, self-attention can be written as an iterative optimization step of the IB objective. This partly explains the emerging grouping phenomenon since IB is known to promote clustered codes.

As shown in Fig.2 (a), previous Vision Transformers of-

<sup>1</sup>eigenvalues are larger than a predefined threshold  $\epsilon$ .

ten adopt a multi-head attention design, followed by an MLP block to aggregate the information from multiple separate heads. Since different heads tend to focus on different components of objects, the multi-head attention design essentially forms a mixture of information bottlenecks. As a result, how to aggregate the information from different heads matters. We aim to come up with an aggregation design that strengthens the symbiosis of grouping and robustness. As shown in Fig.2 (b), we propose a novel attentional channel processing design which promotes channel selection through reweighting. Unlike the static convolution operations in the MLP block, the attentional design is dynamic and content-dependent, leading to more compositional and robust representations. The proposed module results in a new family of Transformer backbone, coined Fully Attentional Networks (FANs) after their designs.

Our contributions can be summarized as follows:

- Instead of focusing on empirical studies, this work provides an explanatory framework that unifies the trinity of grouping, information bottleneck and robust generalization in Vision Transformers.
- The proposed fully attentional design is both efficient and effective, bringing systematically improved robustness with marginal extra costs. Compared with state-of-the-art architectures such as ConvNeXt, our model shows favorable performance in both clean and robust accuracy in image classification. For instance, our model achieves 47.7% mCE on ImageNet-C with 28M parameters, better than ResNet-50, Swin-T and recent SOTA ConvNeXt-T by 29.0%, 11.9% and 5.5% under the comparable model size. By scaling the FAN model to 76.8M model size, we achieve 35.8% mCE, the new state-of-the-art robustness under all supervised trained models.
- We also conduct extensive experiments in semantic segmentation and object detection. We show that the significant gain in robustness from our proposed design is transferrable to these downstream tasks.

Our study indicates the non-trivial benefit of attention representations in robust generalization, and is in line with the recent line of research observing the intriguing robustness in ViTs. We hope our observations and discussions can lead to a better understanding of the representation learning in ViTs and encourage the community to go beyond standard recognition tasks on clean images.

## 2. Fully Attentional Networks

In this section, we examine some emerging properties in ViTs and interpret these properties from an information bottleneck perspective. We then present the proposed Fully Attentional Networks (FANs).

## 2.1. Preliminaries on Vision Transformers

A standard ViT first divides an input image into  $n$  patches uniformly and encodes each patch into a token embedding  $\mathbf{x}_i \in \mathbb{R}^d, i = 1, \dots, n$ . Then, all these tokens are fed into a stack of transformer blocks. Each transformer block leverages self-attention for token mixing and MLPs for channel-wise feature transformation. The architecture of a transformer block is illustrated in the left of Figure 2.

**Token mixing.** Vision transformers leverage self-attention to aggregate global information. Suppose the input token embedding tensor is  $X = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ , SA applies linear transformation with parameters  $W_K, W_Q, W_V$  to embed them into the key  $K = W_K X \in \mathbb{R}^{d \times n}$ , query  $Q = W_Q X \in \mathbb{R}^{d \times n}$  and value  $V = W_V X \in \mathbb{R}^{d \times n}$  respectively. The SA module then computes the attention matrix and aggregates the token features as follows:

$$Z^\top = \text{SA}(X) = \text{Softmax}\left(\frac{Q^\top K}{\sqrt{d}}\right) V^\top W_L, \quad (1)$$

where  $W_L \in \mathbb{R}^{d \times d}$  is a linear transformation and  $Z = [\mathbf{z}_1, \dots, \mathbf{z}_n]$  is the aggregated token features and  $\sqrt{d}$  is a scaling factor. The output of the SA is then normalized and fed into the MLP to generate the input to the next block.

**Channel processing.** Most ViTs adopt an MLP block to transform the input tokens into features  $Z$ :

$$Z' = \text{MLP}(Z). \quad (2)$$

The block contains two Linear layers and a GELU layer.

## 2.2. Intriguing Properties of Self-Attention

We begin with the observation that meaningful clusters emerge on ViT’s token features  $\mathbf{z}$ . We examine such phenomenon using spectral clustering (Ng et al., 2002), where the token affinity matrix is defined as  $S_{ij} = \mathbf{z}_i^\top \mathbf{z}_j$ . Since the number of major clusters can be estimated by the multiplicity of significant eigenvalues (Zelnik-Manor & Perona, 2004) of  $S$ , we plot the number of (in)significant eigenvalues across different ViT-S blocks (Figure 3 (a)). We observe that by feeding Gaussian noise  $\mathbf{x} \sim \mathcal{N}(0, 1)$ , the resulting perturbation (measured the by normalized feature norm) decreases rapidly together with the number of significant eigenvalues. Such observation indicates the symbiosis of grouping and improved robustness over middle blocks.

We additionally visualize the same plot for FAN-S-ViT in Figure 3 (b) where similar trend holds even more obviously. The noise decay of ViT and FAN is further compared to ResNet-50 in Figure 3 (c). We observe that: 1) the robustness of ResNet-50 tends to improve upon downsampling but plateaus over regular convolution blocks. 2) The final noise decay of ResNet-50 less significant. Finally, we visualize the grouped tokens obtained at different blocks in Figure 4,

which demonstrates the process of visual grouping by gradually squeezing out unimportant components. Additional visualizations on different features (tokens) from different backbones are provided in the appendix.

## 2.3. An Information Bottleneck Perspective

The emergence of clusters and its symbiosis with robustness in Vision Transformers draw our attention to early pioneer works in visual grouping (U.C. Berkeley; Buhmann et al., 1999). In some sense, visual grouping can also be regarded as some form of lossy compression (Yang et al., 2008). We thus present the following explanatory framework from an information bottleneck perspective.

Given a distribution  $X \sim \mathcal{N}(X', \epsilon)$  with  $X$  being the observed noisy input and  $X'$  the target clean code, IB seeks a mapping  $f(Z|X)$  such that  $Z$  contains the relevant information in  $X$  for predicting  $X'$ . This goal is formulated as the following information-theoretic optimization problem:

$$f_{\text{IB}}^*(Z|X) = \arg \min_{f(Z|X)} I(X, Z) - I(Z, X'), \quad (3)$$

Here the first term compresses the information and the second term encourages to maintain the relevant information.

In the case of an SA block,  $Z = [\mathbf{z}_1, \dots, \mathbf{z}_n] \in \mathbb{R}^{d \times n}$  denote the output features and  $X = [\mathbf{x}_1, \dots, \mathbf{x}_n] \in \mathbb{R}^{d \times n}$  the input. Assuming  $i$  is the data point index, we have:

**Proposition 2.1.** *Under mild assumptions, the iterative step to optimize the objective in Eqn. (3) can be written as:*

$$\mathbf{z}_c = \sum_{i=1}^n \frac{\log[n_c/n]}{n \det \Sigma} \frac{\exp\left[\frac{\mu_c^\top \Sigma^{-1} \mathbf{x}_i}{1/2}\right]}{\sum_{c=1}^n \exp\left[\frac{\mu_c^\top \Sigma^{-1} \mathbf{x}_i}{1/2}\right]} \mathbf{x}_i, \quad (4)$$

or in matrix form:

$$Z = \text{Softmax}(Q^\top K/d)V^\top, \quad (5)$$

with  $V = [\mathbf{x}_1, \dots, \mathbf{x}_n] \frac{\log[n_c/n]}{n \det \Sigma}$ ,  $K = [\mu_1, \dots, \mu_N] = W_K X$ ,  $Q = \Sigma^{-1}[\mathbf{x}_1, \dots, \mathbf{x}_n]$  and  $d = 1/2$ . Here  $n_c$ ,  $\Sigma$  and  $W_K$  are learnable variables.

**Remark.** We defer the proof to the appendix. The above proposition establishes an interesting connection between the vanilla self-attention (1) and IB (3), by showing that SA aggregates similar inputs  $\mathbf{x}_i$  into representations  $Z$  with cluster structures. Self-attention updates the token features following an IB principle, where the key matrix  $K$  stores the temporary cluster center features  $\mu_c$  and the input features  $\mathbf{x}$  are clustered to them via soft association (softmax). The new cluster center features  $\mathbf{z}$  are output as the updated token features. The stacked SA modules in ViTs can be broadly regarded as an iterative repeat of this optimization which promotes grouping and noise filtering.

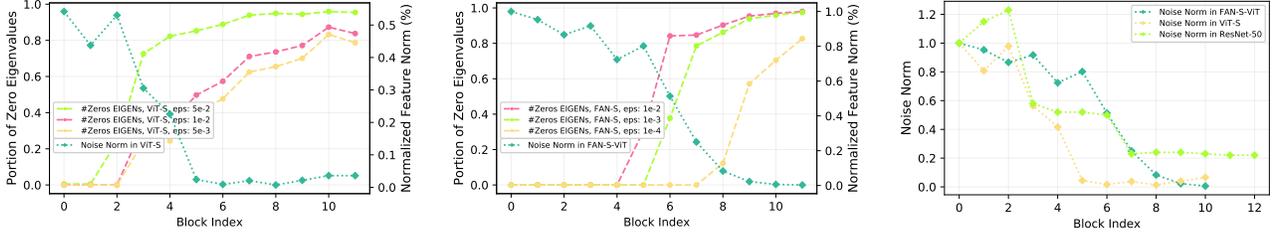


Figure 3. Analysis on the grouping of tokens and noise decay. (a) and (b) shows the # of insignificant (zero) eigenvalues and the noise input decay of ViT-S and FAN-S respectively; (c) shows the comparison of noise norm across different blocks in FAN-S, ViT-S and ResNet-50. Plots shown in (a) and (b) show that the number of zero eigenvalues increases as the model goes deeper, which indicates the emerging grouping of tokens. Given the input Gaussian noise, its magnitude similarly decays over more self-attention blocks. Such a phenomenon is not observed in the ResNet-50 model.

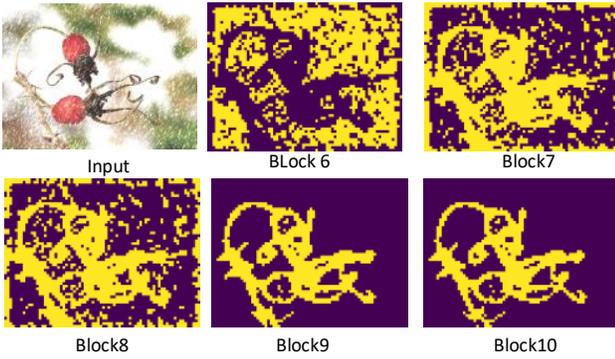


Figure 4. Clustering visualization for different blocks. The visualization is based on our proposed FAN-S model as detailed in Table 1. The cluster visualizations are generated by applying spectral clustering on token features from each FAN block.

**Multi-head Self-attention (MHSA).** Many current Vision Transformer architectures adopt an MHSA design where each head tends to focus on different object components. In some sense, MHSA can be interpreted as a mixture of information bottlenecks. We are interested in the relation between the number of heads versus the robustness under a fixed total number of channels. As shown in Figure 7, having more heads leads to improved expressivity and robustness. But the reduced channel number per head also causes decreased clean accuracy. The best trade-off is achieved with 32 channels per head.

## 2.4. Fully Attentional Networks

With the above mixture of IBs interpretation, we intend to design a channel processing module that strengthens robust representation through the aggregation across different heads. Our design is driven by two main aspects: 1) To promote more compositional representation, it is desirable to introduce channel reweighting since some heads or channels do capture more significant information than the others. 2) The reweighting mechanism should involve more spatially holistic consideration of each channel to leverage the pro-

motored grouping information, instead of making “very local” channel aggregation decisions.

A starting point towards the above goals is to introduce a channel self-attention design similar to XCiT (El-Nouby et al., 2021). As shown in Figure 5 (a), the channel attention (CA) module adopts a self-attention design which moves the MLP block into the self-attention block, followed by matrix multiplication with the  $D \times D$  channel attention matrix from the channel attention branch.

**Attentional feature transformation.** A FAN block introduces the following channel attention (CA) to perform feature transformation which is formulated as:

$$CA(Z) = \text{Softmax} \left( \frac{(W'_Q Z)(W'_K Z)^T}{\sqrt{n}} \right) \text{MLP}(Z), \quad (6)$$

Here  $W'_Q \in \mathbb{R}^{d \times d}$  and  $W'_K \in \mathbb{R}^{d \times d}$  are linear transformation parameters. Different from SA, CA computes the attention matrix along the channel dimension instead of the token dimension (recall  $Z \in \mathbb{R}^{d \times n}$ ), which leverages the feature covariance (after linear transformation  $W'_Q, W'_K$ ) for feature transformation. Strongly correlated feature channels with larger correlation values will be aggregated while outlier features with low correlation values will be isolated. This aids the model in filtering out irrelevant information. With the help of CA, the model can filter irrelevant features and thus form more precise token clustering for the foreground and background tokens. We will give a more formal description on such effects in the following section.

We will verify the improved robustness from CA over existing ViT models in the rest of the paper.

## 2.5. Efficient Channel Self-attention

There are two limits of applying the conventional self-attention calculation mechanism along the channel dimension. The first one is the **computational overhead**. The computational complexity of CA introduced in Eqn 6 is

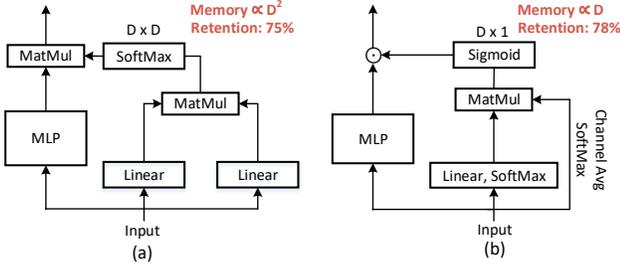


Figure 5. **Comparison among channel attention designs.** (a) CA: a channel self attention design similar to XCiT (El-Nouby et al., 2021), but differently applied on the output of the MLP block. (b) The proposed efficient channel attention (ECA).

quadratically proportional to  $D^2$ , where  $D$  is the channel dimension. For modern pyramid model designs (Wang et al., 2021; Liu et al., 2021), the channel dimension becomes larger and larger at the top stages. Consequently, direct applying CA can cause a large computational overhead. The second one is the **low parameter efficiency**. In conventional SA module, the attention distribution of the attention weights is sharpened via a Softmax operation. Consequently, only a partial of the channels could contribute to the representation learning as most of the channels are diminished by being multiplied with a small attention weights. To overcome these, we explore a novel self-attention like mechanism that is equipped with both the high computational efficiency and parameter efficiency. Specifically, two major modifications are proposed. First, instead of calculating the co-relation matrix between the tokens features, we first generate a token prototype,  $\bar{Z}$ ,  $\bar{Z} \in R^{n \times 1}$ , by averaging over the channel dimension. Intuitively,  $\bar{Z}$  aggregates all the channel information for each spatial positions represented by tokens. Thus, it is informative to calculate the co-relation matrix between the token features and token prototype  $\bar{Z}$ , resulting in learn complexity with respect to the channel dimension. Secondly, instead of applying a Softmax function, we use a Sigmoid function for normalizing the attention weights and then multiply it with the token features instead of using MatMul to aggregate channel information. Intuitively, we do not force the channel to select only a few of the ‘‘important’’ token features but re-weighting each channel based on the spatial co-relation. Indeed, the channel features are typically considered as independent. A channel with large value should not restrain the importance of other channels. By incorporating those two design concepts, we propose a novel channel self-attention and it is calculated via Eqn. (7):

$$\text{ECA}(Z) = \text{Norm} \left( \frac{(W'_Q \sigma(Z)) \sigma(\bar{Z})^\top}{\sqrt{n}} \right) \odot \text{MLP}(Z), \quad (7)$$

Here,  $\sigma$  denotes the Softmax operation along the token dimension and  $\bar{Z}$  denotes the token prototype ( $\bar{Z} \in R^{1 \times N}$ ). We use sigmoid as the Norm. The detailed block architecture

design is also shown in Figure 5. We verify that the novel efficient channel self-attention takes consumes less computational cost while improve the performance significantly. The detailed results will be shown in Sec. 3.2.

## 3. Experiment Results & Analysis

### 3.1. Experiment details

**Datasets and evaluation metrics.** We verify the model robustness on Imagenet-C (IN-C), Cityscape-C and COCO-C without extra corruption related fine-tuning. The suffix ‘-C’ denotes the corrupted images based on the original dataset with the same manner proposed in (Hendrycks & Dietterich, 2019). To test the generalization to other types of out-of-distribution (OOD) scenarios, we also evaluate the accuracy on ImageNet-A (Hendrycks et al., 2021) (IN-A) and ImageNet-R (IN-R) (Hendrycks & Dietterich, 2019). In the experiments, we evaluate the performance with both the clean accuracy on ImageNet-1K (IN-1K) and the robustness accuracy on these out-of-distribution benchmarks. To quantify the resilience of a model against corruptions, we propose to calibrate with the clean accuracy. We use retention rate (Ret R) as the robustness metric, defined as  $R = \frac{\text{Robust Acc.}}{\text{Clean Acc.}} = \frac{\text{IN-C}}{\text{IN-1K}}$ . We also report the mean corruption error (mCE) following (Hendrycks & Dietterich, 2019). For more details, please refer to Appendix A.2. For Cityscapes, we take the average mIoU for three severity levels for the noise category, following the practice in SegFormer (Xie et al., 2021). For all the rest of the datasets, we take the average of all five severity levels.

**Model selection.** We design four different model sizes (Tiny, Small, Base and large) for our FAN models, abbreviated as ‘-T’, ‘-S’, ‘-B’ and ‘-L’ respectively. Their detailed configurations are shown in Table 1. For ablation study, we use ResNet-50 as a representative model for CNNs and ViT-S as a representative model for the conventional vision transformers. ResNet-50 and ViT-S have similar model sizes and computation budget as FAN-S. When comparing with SOTA models, we take the most recent vision transformer and CNN models as baselines.

Table 1. Details and abbreviations of different FAN variants.

Model	#Blocks	Channel Dim.	#Heads	Param.	FLOPs
FAN-T	12	192	4	7.3M	1.4G
FAN-S	12	384	8	28.3M	5.3G
FAN-B	18	448	8	54.0M	10.4G
FAN-L	24	480	10	80.5M	15.8G

### 3.2. Analysis

In this section, we present a series of ablation studies to analyze the contribution of self-attention in model robustness. Since multiple advanced training recipes have been recently introduced, we first investigate their effects in improving

model robustness. We then compare ViTs and CNNs with exactly the same training recipes to exclude factors other than architecture design that might affect model robustness.

**Effects of advanced training tricks.** We empirically evaluate how different training recipes could be used to improve the robustness, with the results reported in Table 2. Interestingly, it is observed that widely used tricks such as knowledge distillation (KD) and large dataset pretraining do improve the absolute accuracy. However, they do not significantly reduce the performance degradation when transferred to ImageNet-C. The main improvement comes from the advanced training recipe such as the CutMix and RandAugmentation adopted in DeiT training recipe. In the following comparison, we use the ViT-S trained with DeiT recipe and increased block number with reduced channel dimension, denoted as ViT-S\*. In addition, to make fair comparison, we first apply those advanced training techniques to reproduce the ResNet-50 performance.

Table 2. Impacts of various performance improvement tricks on model robustness (%).

Model	IN-1K	IN-C	Retention	mCE ( $\downarrow$ )
ViT-S	77.9	54.2	70	63.5
+ DeiT Recipe	79.3	57.1	72	57.1
+ #Blocks (8 $\uparrow$ 12)	79.9	58.0	72	56.2
+ KD	81.3	59.6	73	54.0
+ IN22K w/o KD	81.8	59.7	73	54.2

**Adding new training recipes to CNNs.** We make a step by step empirical study on how the robustness of ResNet-50 model changes when adding advanced tricks. We examine three design choices: training recipe, attention mechanism and down-sampling methods. For the training recipe, we adopt the same one as used in training the above ViT-S model. We use Squeeze-and-Excite (SE) attention (Hu et al., 2018) and apply it along the channel dimension for the feature output of each block. We also investigate different downsampling strategies, i.e., average pooling (ResNet-50 default) and strided convolution. The results are reported in Table 3. As can be seen, adding attention (Squeeze-and-Excite (SE) attention) and using more advanced training recipe do improve the robustness of ResNet-50 significantly. We take the best-performing ResNet-50 with all these tricks, denoted as ResNet-50\*, for the following comparison.

Table 3. Robustness of ResNet-50 with various performance improvement tricks (%).

Model	IN-1K	IN-C	Retention	mCE ( $\downarrow$ )
ResNet-50	76.0	38.8	51	76.7
+ DeiT Recipe	79.0	43.9	46	69.7
+ SE	79.8	50.1	63	63.1
+ Strided Conv	80.2	52.1	65	61.6

**Advantages of ViTs over CNNs on robustness.** To make fair comparison, we use all the above validated training

tricks to train the ViT-S and ResNet-50 to their best performance. Specifically, ResNet-50\* is trained with DeiT recipe, SE and strided convolution; ViT-S\* is also trained with DeiT recipe and has 12 blocks with 384 embedding dimension for matching the model size as ResNet-50. Results in Table 4 show that even with the same training recipe, ViTs still outperform ResNet-50 in robustness. These results indicate that the improved robustness in ViTs may come from their architectural advantages with self-attention. This motivates us to further improve the architecture of ViTs by leveraging self-attention more broadly to further strengthen the model’s robustness.

Table 4. Robustness comparison between ResNet-50 and ViT-S (%).

Model	Param	IN-1K	IN-C	Retention	mCE ( $\downarrow$ )
ResNet-50*	25M	80.2	52.1	65	61.6
ViT-S*	22M	79.9	58.0	72	56.2

**Difference among ViT, SWIN-ViT and ConvNeXt.** Very recent CNN models has shown superiority of the robustness over the recent state-of-the-art transformer based models SWIN transformer. We here interpret this from the view of information bottleneck. As explained in Sec. 2.3, the SA module is forming an IB to select essential tokens. As SWIN transformer deploys a window based local self-attention mechanism, it forces the model to select information from a predefined window area. Such a local window IB forces each window to select tokens from a local constrained features. Intuitively, when a selected window contains no essential information, a local SA is forced to select some key tokens and thus resulting a set of sub-optimal clusters. Thus, the robustness of SWIN transformer is worse than the recent SOTA CNN model ConvNeXt. However, as shown in Table 5, DeiT achieve better robustness with 24.1% less number of parameters, compared to ConvNeXt model. We thus argue that transformers with global SA module are still more robust than the state-of-the-art ConvNeXt model.

Table 5. Robustness comparison among Swin, ConvNeXt, DeiT and FAN. The mIoU of ConvNeXt, DeiT, Swin and SegFormer models are our reproduced results.

Model	Param.	ImageNet			Cityscapes		
		Clean	Corrupt	Reten.	Clean	Corrupt	Reten.
ConvNeXt (Liu et al.)	29M	82.1	59.1	72.0	79.0	54.2	68.6
SWIN (Liu et al.)	28M	81.3	55.4	68.1	78.0	47.3	61.7
DeiT-S (Touvron et al.)	22M	79.9	58.1	72.7	76.0	55.4	72.9
FAN-Hybrid-S (Ours)	26M	83.5	64.7	78.2	81.5	66.4	81.5

### 3.3. Fully Attentional Networks

In this subsection, we investigate how the new FAN architecture improves the model’s robustness among different architectures.

**Impacts of efficient channel attention** We first ablate the

impacts of different forms of channel attentions in terms of GPU memory consumption, clean image accuracy and robustness. The results are shown in Table 6. Compared to the original self-attention module, SE attention consumes less memory and achieve comparable clean image accuracy and model robustness. By taking the spatial relationship into consideration, our proposed CSA produces the best model robustness with comparable memory consumption to the SE attention.

Table 6. Effects of different channel attentions on model robustness (%).

Model	Mem.(M)	IN-1K	IN-C	Retention	mCE ( $\downarrow$ )
FAN-ViT-S-SA	235	81.3	61.7	76	51.4
FAN-ViT-S-SE	126	81.2	62.0	76	50.0
FAN-ViT-S-ECA	127	82.5	64.6	78	47.7

**FAN-ViT & FAN-Swin.** Using the FAN block to replace the conventional transformer block forms the FAN-ViT. FAN-ViT significantly enhances the robustness. However, compared to ViT, the robustness of Swin architecture (Liu et al., 2021) (which uses shifted window attention) drops. This is possibly because their local attention hinders global clustering and the IB-based information extraction, as detailed in Section 3.2. The drop in robustness can be effectively remedied by using the FAN block. By adding the ECA to the feature transformation of SWIN models, we obtain the FAN-SWIN, a new FAN model whose spatial self-attention is augmented by the shifted window attention in SWIN. As shown in Table 7, adding FAN block improves the accuracy on ImageNet-C by 5%. Such a significant improvement shows that our proposed CSA does have significant effectiveness on improving the model robustness.

Table 7. Effects of architectural changes on model robustness (%).

Model	IN-1K	IN-C	Retention	mCE ( $\downarrow$ )
ViT-S*	79.9	58.1	73	56.2
+ FAN	81.3	61.7	76	51.4
Swin-T	81.4	55.4	68	59.6
+ FAN	81.9	59.4	73	54.5
ConvNeXt-T	82.1	59.1	72	54.8
+ FAN	82.5	60.8	74	53.1

**FAN-Hybrid.** From the clustering process as presented in Figure 3, we find that the clustering mainly emerges at the top stages of the FAN model, implying the bottom stages to focus on extracting local visual patterns. Motivated by this, we propose to use convolution blocks for the bottom two stages with down-sampling and then append FAN blocks to the output of the convolutional stages. Each stage includes 3 convolutional blocks. This gives the FAN-Hybrid model. In particular, we use the ConvNeXt (Liu et al., 2022), a very recent CNN model, to build the early stages of our hybrid model. As shown in Table 7, we find original ConvNeXt exhibits strong robustness than SWIN transformer, but performs less robust than FAN-ViT and FAN-Swin models.

However, the FAN-Hybrid achieves comparable robustness as FAN-ViT and FAN-SWIN and presents higher accuracy for both clean and corrupted datasets, implying FAN can also effectively strengthen the robustness of a CNN-based model. Similar to FAN-SWIN, FAN-Hybrid enjoys efficiency for processing large-resolution inputs and dense prediction tasks, making it favorable for downstream tasks. Thus, for all downstream tasks, we use FAN-Hybrid model to compare with other state-of-the-art models. More details on the FAN-Hybrid and FAN-SWIN architecture can be found in the appendix.

### 3.4. Comparison to SOTAs on various tasks

In this subsection, we evaluate the robustness of FAN with other SOTA methods against common corruptions on different downstream tasks, including image classification (ImageNet-C), semantic segmentation (Cityscapes-C) and object detection (COCO-C). Additionally, we evaluate the robustness of FAN on various other robustness benchmarks including ImageNet-A and ImageNet-R to further show its non-trivial improvements in robustness.

**Robustness in image classification.** We first compare the robustness of FAN with other SOTA models by directly applying them (pre-trained on ImageNet-1K) to the ImageNet-C dataset (Hendrycks & Dietterich, 2019) without any fine-tuning. We divide all the models into three groups according to their model size for fair comparison. The results are shown in Table 8 and the detailed results are summarized in Table 12. From the results, one can clearly observe that all the transformer-based models show stronger robustness than CNN-based models. Under all the models sizes, our proposed FAN models surpass all other models significantly. They offer strong robustness to all the types of corruptions. Notably, FANs perform excellently robust for bad weather conditions and digital noises, making them very suitable for vision applications in mobile phones and self-driving cars.

We also evaluate the zero-shot robustness of the Swin transformer and the recent ConvNeXt. Both of them demonstrate weaker robustness than the transformers with global self-attention. However, adding FAN to them improves their robustness, enabling the resulted FAN-SWIN and FAN-Hybrid variants to inherit both high applicability for downstream tasks and strong robustness to corruptions. We will use FAN-Hybrid variants in the applications of segmentation and detection.

**Robustness in semantic segmentation.** We further evaluate robustness of our proposed FAN model for the segmentation task. We use the Cityscapes-C for evaluation, which expands the Cityscapes validation set with 16 types of natural corruptions. We compare our model to variants of DeeplabV3+ and latest SOTA models. The results are summarized in Table 9 and by category results are summarized

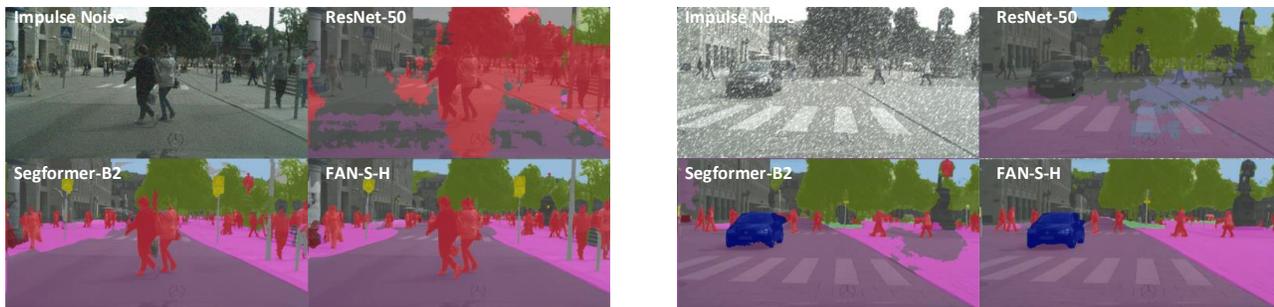


Figure 6. Segmentation visualization on corrupted images with impulse noise (severity 3) and snow (severity 3). We select the recent state-of-the-art SegFormer model (Xie et al., 2021) as a strong baseline. FAN-S-H denotes our hybrid model. Under comparable model size and computation, FAN achieve significantly improved segmentation results over ResNet-50 and SegFormer-B2 model. A video demo is available via [external players](#) and in Figure 8 in the appendix.

Table 8. Main results on image classification. FAN models show improved performance in both clean accuracy and robustness than other models. † denotes models are pretrained on ImageNet-22K.

Model	Param./FLOPs	IN-1K	IN-C	Retention
ResNet18 (He et al.)	11M/1.8G	69.9	32.7	46.8%
MBV2 (Sandler et al.)	4M/0.4G	73.0	35.0	47.9%
EffiNet-B0 (Tan & Le)	5M/0.4G	77.5	41.1	53.0%
PVTv2-B0 (Wang et al.)	3M/0.6G	70.5	36.2	51.3%
PVTv2-B1 (Wang et al.)	13M/2.1G	78.7	51.7	65.7%
FAN-T-ViT	7M/1.3G	79.2	57.5	<b>72.6%</b>
FAN-T-Hybrid	7M/3.5G	80.1	57.4	71.7%
ResNet50 (He et al.)	25M/4.1G	79	50.6	64.1%
DeiT-S (Touvron et al.)	22M/4.6G	79.9	58.1	72.7%
Swin-T (Liu et al.)	28M/4.5G	81.3	55.4	68.1%
ConvNeXt-T (Liu et al.)	29M/4.5G	82.1	59.1	71.9%
FAN-S-ViT	28M/5.3G	82.9	64.5	<b>77.8%</b>
FAN-S-Hybrid	26M/6.7G	83.5	64.7	77.5%
Swin-S (Liu et al.)	50M/8.7G	83.0	60.4	72.8%
ConvNeXt-S (Liu et al.)	50M/8.7G	83.1	61.7	74.2%
FAN-B-ViT	54M/10.4G	83.6	67.0	80.1%
FAN-B-Hybrid	50M/11.3G	83.9	66.4	79.1%
FAN-B-Hybrid†	50M/11.3G	85.6	70.5	<b>82.4%</b>
DeiT-B (Touvron et al.)	89M/17.6G	81.8	62.7	76.7%
Swin-B (Liu et al.)	88M/15.4G	83.5	60.4	72.3%
ConvNeXt-B (Liu et al.)	89M/15.4G	83.8	61.7	73.6%
FAN-L-ViT	81M/15.8G	83.9	67.7	80.7%
FAN-L-Hybrid	77M/16.9G	84.3	68.3	81.0%
FAN-L-Hybrid†	77M/16.9G	86.5	73.6	<b>85.1%</b>

in Table 13. Our model significantly outperforms previous models. FAN-S-Hybrid surpasses the latest SegFormer—a transformer based segmentation model—by 6.8% mIoU with comparable model size. The results indicate strong robustness of FAN.

**Robustness in object detection.** We also evaluate the robustness of FAN for the detection task on COCO-C dataset, an extension of COCO generated similarly as Cityscapes-C. The results are summarized in Table 10 and the detailed results are summarized in Table 14. FAN demonstrates strong robustness again, yielding improvement over recent

Table 9. Main results on semantic segmentation. ‘R-’ and ‘X-’ refer to DeepLabv3+, ResNet and Xception. The mIoUs of DeepLabv3+ framework are reported from (Kamann & Rother, 2020). FAN shows significantly stronger clean accuracy and robustness than other models.

Model	Encoder Size	City	City-C	Retention
DeepLabv3+ (R50)	25.4M	76.6	36.8	48.0%
DeepLabv3+ (R101)	47.9M	77.1	39.4	51.1%
DeepLabv3+ (X65)	22.8M	78.4	42.7	54.5%
DeepLabv3+ (X71)	-	78.6	42.5	54.1%
ICNet (Zhao et al.)	-	65.9	28.0	42.5%
FCN8s (Long et al.)	50.1M	66.7	27.4	41.1%
DilatedNet (Yu & Koltun)	-	68.6	30.3	44.2%
ResNet38 (Wu et al.)	-	77.5	32.6	42.1%
PSPNet (Zhao et al.)	13.7M	78.8	34.5	43.8%
ConvNeXt-T (Liu et al.)	29.0M	79.0	54.4	68.9%
SETR (Heo et al.)	22.1M	76.0	55.3	72.8%
SWIN-T (Liu et al.)	28.4M	78.1	47.3	60.6%
SegFormer-B0 (Xie et al.)	3.4M	76.2	48.8	64.0%
SegFormer-B1 (Xie et al.)	13.1M	78.4	52.7	67.2%
SegFormer-B2 (Xie et al.)	24.2M	81.0	59.6	73.6%
SegFormer-B5 (Xie et al.)	81.4M	82.4	65.8	79.9%
FAN-T-Hybrid (Ours)	7.4M	81.2	57.1	70.3%
FAN-S-Hybrid (Ours)	26.3M	81.5	66.4	81.5%
FAN-B-Hybrid (Ours)	50.4M	82.2	66.9	81.5%
FAN-L-Hybrid (Ours)	76.8M	82.3	68.7	<b>83.5%</b>

SOTA Swin transformer (Liu et al., 2021) by 6.2% mAP with comparable model size (26M vs 29M) under same training settings and showing a new state-of-the-art results of 42.0% mAP with only 76.8M number of parameters for the encoder model.

**Robustness against out-of-distribution.** The FAN encourages token features to form clusters and implicitly selects the informative features, which would benefit generalization performance of the model. To verify this, we directly test our ImageNet-1K trained models for evaluating their robustness, in particular for out-of-distribution samples, on ImageNet-A and ImageNet-R. The experiment results are summarized in Table 11. Among these models, ResNet-50 (Liu et al.) presents weakest generalization ability while the

**Table 10. Main results on object detection.** FAN shows stronger clean accuracy and robustness than other models. ‘†’ denotes the accuracy pretrained on ImageNet-22K.

Model	Encoder Size	COCO	COCO-C	Retention
Mask R-CNN				
ResNet-50 (He et al., 2016)	25.4M	39.9	21.3	53.3%
ResNet101 ((He et al., 2016))	44.1M	41.8	23.3	55.7%
DeiT-S (Touvron et al., 2021a)	22.1M	40.0	26.9	67.3%
Swin-T (Liu et al., 2021)	28.0M	46.0	29.3	63.7%
FAN-T-Hybrid	7.4M	45.8	29.7	64.8%
FAN-S-Hybrid	26.3M	49.1	35.5	72.3%
Cascade R-CNN				
FAN-T-Hybrid	7.4M	50.2	33.1	65.9%
FAN-S-Hybrid	26.3M	53.3	38.7	72.6%
FAN-L-Hybrid	76.8M	54.1	40.6	75.0%
FAN-L-Hybrid†	76.8M	55.1	42.0	<b>76.2%</b>

recent ConvNeXt substantially improves the generalization performance of CNNs. The transformer-based models, Swin and RVT performs comparably well as ConvNeXt and much better than ResNet-50. Our proposed FANs outperform all these models significantly, implying the fully-attentional architecture aids generalization ability of the learned representations as the irrelevant features are effectively processed.

**Table 11. Main results on out-of-distribution generalization.** FAN models show improved generalization across all datasets. ‘‡’ denotes results with finetuning on  $384 \times 384$  image resolution. IN-C is measured by mCE ( $\downarrow$ ). All metrics are scaled by (%).

Model	Params (M)	Clean	IN-A	IN-R	IN-C
ImgNet-1k pretrain					
XCiT-S12 (El-Nouby et al.)	26.3	81.9	25.0	45.5	51.5
XCiT-S24 (El-Nouby et al.)	47.7	82.6	27.8	45.5	49.4
RVT-S* (Mao et al.)	23.3	81.9	25.7	47.7	51.4
RVT-B* (Mao et al.)	91.8	82.6	28.5	48.7	46.8
Swin-T (Liu et al.)	28.3	81.2	21.6	41.3	59.6
Swin-S (Liu et al.)	50	83.4	35.8	46.6	52.7
Swin-B (Liu et al.)	87.8	83.4	35.8	64.2	54.4
ConvNeXt-T (Liu et al.)	28.6	82.1	24.2	47.2	53.2
ConvNeXt-S (Liu et al.)	50.2	82.1	31.2	49.5	51.2
ConvNeXt-B (Liu et al.)	88.6	83.8	36.7	51.3	46.8
FAN-S-ViT (Ours)	28.0	82.5	29.1	50.4	47.7
FAN-B-ViT (Ours)	54.0	83.6	35.4	51.8	44.4
FAN-L-ViT (Ours)	80.5	83.9	37.2	53.1	43.3
FAN-S-Hybrid (Ours)	26.0	83.6	33.9	50.7	47.8
FAN-B-Hybrid (Ours)	50.0	83.9	39.6	52.9	45.2
FAN-L-Hybrid (Ours)	76.8	84.3	41.8	53.2	43.0
ImgNet-22k pretrain					
ConvNeXt-B‡ (Liu et al.)	88.6	86.8	62.3	64.9	43.1
FAN-L-Hybrid (Ours)	76.8	86.5	60.7	64.3	<b>35.8</b>
FAN-L-Hybrid‡ (Ours)	76.8	<b>87.1</b>	<b>74.5</b>	<b>71.1</b>	36.0

## 4. Related Works

Vision Transformers (Vaswani et al., 2017) are a family of transformer-based architectures on computer vision tasks. Unlike CNNs relying on certain inductive biases (e.g., locality and translation invariance), ViTs perform the global

interactions among visual tokens via self-attention, thus having less inductive bias about the input image data. Such designs have offered significant performance improvement on various vision tasks including image classification (Dosovitskiy et al., 2020; Yuan et al., 2021; Zhou et al., 2021a;b), object detection (Carion et al., 2020; Zhu et al., 2020; Dai et al., 2021; Zheng et al., 2020) and segmentation (Wang et al., 2020; Liu et al., 2021; Zheng et al., 2020). The success of vision transformers for vision tasks triggers broad debates and studies on the advantages of self-attention versus convolutions (Raghu et al., 2021; Tang et al., 2021). Compared to convolutions, an important advantage is the robustness against observable corruptions. Several works (Bai et al., 2021; Xie et al., 2021; Zhu et al., 2021; Paul & Chen, 2022; Naseer et al., 2021) have empirically shown that the robustness of ViTs against corruption consistently outperforms ConvNets by significant margins. However, how the key component (i.e. self-attention) contributes to the robustness is under-explored. In contrast, our work conducts empirical studies to reveal intriguing properties (i.e., token grouping and noise absorbing) of self-attention for robustness and presents a novel fully attentional architecture design to further improve the robustness.

There exists a large body of work on improving robustness of deep learning models in the context of adversarial examples by developing robust training algorithms (Kurakin et al., 2016; Shao et al., 2021), which differs from the scope of our work. In this work, we focus the zero-shot robustness to the natural corruptions and mainly study improving model’s robustness from the model architecture perspective.

## 5. Conclusion

In this paper, we verified self-attention as a contributor of the improved robustness in vision transformers. Our study shows that self-attention promotes naturally formed clusters in tokens, which exhibits interesting relation to the extensive early studies in vision grouping prior to deep learning. We also established an explanatory framework from the perspective of information bottleneck to explain these properties of self-attention. To push the boundary of robust representation learning with self-attention, we introduced a family of fully-attentional network (FAN) architectures, where self-attention is leveraged in both token mixing and channel processing. FAN models demonstrate significantly improved robustness over their CNN and ViT counterparts. Our work provides a new angle towards understanding the working mechanism of vision transformers, showing the potential of inductive biases going beyond convolutions. Our work can benefit wide real-world applications, especially safety-critical ones such as autonomous driving.

## References

- Bai, Y., Mei, J., Yuille, A. L., and Xie, C. Are transformers more robust than cnns? In *NeurIPS*, 2021.
- Buhmann, J. M., Malik, J., and Perona, P. Image recognition: Visual grouping, recognition, and learning. *Proceedings of the National Academy of Sciences*, 96(25): 14203–14204, 1999.
- Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers. In *ECCV*, pp. 213–229. Springer, 2020.
- Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 9650–9660, 2021.
- Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., et al. Mmdetection: Open mmlab detection toolbox and benchmark. *arXiv:1906.07155*, 2019.
- Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, pp. 801–818, 2018.
- Contributors, M. Mmsegmentation: Openmmlab semantic segmentation toolbox and benchmark, 2020.
- Dai, Z., Cai, B., Lin, Y., and Chen, J. Up-detr: Unsupervised pre-training for object detection with transformers. In *CVPR*, pp. 1601–1610, 2021.
- Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2020.
- El-Nouby, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., et al. Xcit: Cross-covariance image transformers. In *NeurIPS*, 2021.
- He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, pp. 770–778, 2016.
- He, K., Gkioxari, G., Dollár, P., and Girshick, R. Mask r-cnn. In *ICCV*, pp. 2961–2969, 2017.
- He, T., Zhang, Z., Zhang, H., Zhang, Z., Xie, J., and Li, M. Bag of tricks for image classification with convolutional neural networks. In *CVPR*, pp. 558–567, 2019.
- Hendrycks, D. and Dietterich, T. Benchmarking neural network robustness to common corruptions and perturbations. *ICLR*, 2019.
- Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., and Song, D. Natural adversarial examples. In *CVPR*, pp. 15262–15271, 2021.
- Heo, B., Yun, S., Han, D., Chun, S., Choe, J., and Oh, S. J. Rethinking spatial dimensions of vision transformers. In *ICCV*, pp. 11936–11945, 2021.
- Hu, J., Shen, L., and Sun, G. Squeeze-and-excitation networks. In *CVPR*, pp. 7132–7141, 2018.
- Kamann, C. and Rother, C. Benchmarking the robustness of semantic segmentation models. In *CVPR*, pp. 8828–8838, 2020.
- Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, pp. 1097–1105, 2012.
- Kurakin, A., Goodfellow, I., and Bengio, S. Adversarial machine learning at scale. In *ICLR*, 2016.
- LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.
- Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pp. 10012–10022, 2021.
- Liu, Z., Mao, H., Wu, C.-Y., Feichtenhofer, C., Darrell, T., and Xie, S. A convnet for the 2020s. *CVPR*, 2022.
- Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation. In *CVPR*, pp. 3431–3440, 2015.
- Mao, X., Qi, G., Chen, Y., Li, X., Duan, R., Ye, S., He, Y., and Xue, H. Towards robust vision transformer. In *CVPR*, 2021.
- Naseer, M., Ranasinghe, K., Khan, S., Hayat, M., Khan, F. S., and Yang, M.-H. Intriguing properties of vision transformers. In *NeurIPS*, 2021.
- Ng, A. Y., Jordan, M. I., and Weiss, Y. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2002.
- Paul, S. and Chen, P.-Y. Vision transformers are robust learners. In *AAAI*, 2022.
- Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., and Dosovitskiy, A. Do vision transformers see like convolutional neural networks? *NeurIPS*, 34, 2021.

- Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28:91–99, 2015.
- Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., and Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pp. 4510–4520, 2018.
- Shao, R., Shi, Z., Yi, J., Chen, P.-Y., and Hsieh, C.-J. On the adversarial robustness of visual transformers. *arXiv:2103.15670*, 2021.
- Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pp. 6105–6114. PMLR, 2019.
- Tang, C., Zhao, Y., Wang, G., Luo, C., Xie, W., and Zeng, W. Sparse mlp for image recognition: Is self-attention really necessary? *arXiv:2109.05422*, 2021.
- Tishby, N. and Zaslavsky, N. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pp. 1–5. IEEE, 2015.
- Tishby, N., Pereira, F. C., and Bialek, W. The information bottleneck method. *physics/0004057*, 2000.
- Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. In *ICML*, pp. 10347–10357. PMLR, 2021a.
- Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., and Jégou, H. Going deeper with image transformers. In *ICCV*, pp. 32–42, 2021b.
- U.C. Berkeley. Reorganization: Grouping, contour detection, segmentation, ecological statistics. <https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/>.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. *NeurIPS*, 30, 2017.
- Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., Lu, T., Luo, P., and Shao, L. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *CVPR*, pp. 568–578, 2021.
- Wang, Y., Xu, Z., Wang, X., Shen, C., Cheng, B., Shen, H., and Xia, H. End-to-end video instance segmentation with transformers. *CVPR*, 2020.
- Wightman, R. Pytorch image models. <https://github.com/rwightman/pytorch-image-models>, 2019.
- Wu, Z., Shen, C., and Van Den Hengel, A. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 90:119–133, 2019.
- Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J. M., and Luo, P. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021.
- Yang, A. Y., Wright, J., Ma, Y., and Sastry, S. S. Unsupervised segmentation of natural images via lossy data compression. *Computer Vision and Image Understanding*, 110(2):212–225, 2008.
- Yu, F. and Koltun, V. Multi-scale context aggregation by dilated convolutions. *ICLR*, 2016.
- Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Tay, F. E., Feng, J., and Yan, S. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *ICCV*, 2021.
- Zelnik-Manor, L. and Perona, P. Self-tuning spectral clustering. In *NIPS*, 2004.
- Zhao, H., Shi, J., Qi, X., Wang, X., and Jia, J. Pyramid scene parsing network. In *CVPR*, pp. 2881–2890, 2017.
- Zhao, H., Qi, X., Shen, X., Shi, J., and Jia, J. Icnnet for real-time semantic segmentation on high-resolution images. In *ECCV*, pp. 405–420, 2018.
- Zheng, M., Gao, P., Wang, X., Li, H., and Dong, H. End-to-end object detection with adaptive clustering transformer. *BMVC*, 2020.
- Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., and Torr, P. H. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *CVPR*, pp. 6881–6890, 2021.
- Zhou, D., Kang, B., Jin, X., Yang, L., Lian, X., Hou, Q., and Feng, J. Deepvit: Towards deeper vision transformer. *arXiv:2103.11886*, 2021a.
- Zhou, D., Shi, Y., Kang, B., Yu, W., Jiang, Z., Li, Y., Jin, X., Hou, Q., and Feng, J. Refiner: Refining self-attention for vision transformers. *arXiv preprint arXiv:2106.03714*, 2021b.
- Zhu, C., Ping, W., Xiao, C., Shoeybi, M., Goldstein, T., Anandkumar, A., and Catanzaro, B. Long-short transformer: Efficient transformers for language and vision. In *NeurIPS*, 2021.
- Zhu, X., Su, W., Lu, L., Li, B., Wang, X., and Dai, J. Deformable detr: Deformable transformers for end-to-end object detection. *ICLR*, 2020.

## A. Supplementary Details

### A.1. Proof on the relationship between the Information Bottleneck and Self-Attention

Given a distribution  $X \sim \mathcal{N}(X', \epsilon)$  with  $X$  being the observed noisy input and  $X'$  the target clean code, IB seeks a mapping  $f(Z|X)$  such that  $Z$  contains the relevant information in  $X$  for predicting  $X'$ . This goal is formulated as the following information-theoretic optimization problem

$$q_{\text{IB}}^*(z|x) = \arg \min_{q(z|x)} I(X, Z) - \beta I(Z, X'), \quad (8)$$

subject to the Markov constraint  $Z \leftrightarrow X \leftrightarrow X'$ .  $\beta$  is a free parameter that trades-off the information compression by the first term and the relevant information maintaining by the second.

The information bottleneck approach can be applied for solving unsupervised clustering problems. Here we choose  $X$  to be the data point with index  $i$  that will be clustered into clusters with indices  $c$ .

As mentioned above, we assume the following data distribution:

$$p(\mathbf{x}|i) \propto \exp \left[ -\frac{1}{2\epsilon^2} \|\mathbf{x} - \mathbf{x}_i\|^2 \right], \quad (9)$$

where  $s$  is a smoothing parameter. We assume the marginal to be  $p(i) = \frac{1}{N}$ , where  $N$  is the number of data points.

Using the above notations, the  $t$ -th step in the iterative IB for clustering is formulated as

$$\begin{aligned} q^{(t)}(c|i) &= \frac{\log q^{(t-1)}(c)}{K(\mathbf{x}, \beta)} \exp \left[ -\beta \text{KL}[p(\mathbf{x}|i)|q^{(t-1)}(\mathbf{x}|c)] \right], \\ q^{(t)}(c) &= \frac{n_c^{(t)}}{N}, \\ q^{(t)}(\mathbf{x}|c) &= \frac{1}{n_c^{(t)}} \sum_{i \in S_c^{(t)}} p(\mathbf{x}|i). \end{aligned} \quad (10)$$

Here  $K(\mathbf{x}, \beta)$  is the normalizing factor and  $S_c$  denotes the set of indices of data points assigned to cluster  $c$ .

We choose to replace  $q(\mathbf{x}|c)$  with a Gaussian approximation  $g(\mathbf{x}|c) = \mathcal{N}(\mathbf{x}|\mu_c, \Sigma_c)$  and assume  $\epsilon$  is sufficiently small. Then,

$$\text{KL}[p(\mathbf{x}|i)|g(\mathbf{x}|c)] \propto (\mu_c - \mathbf{x}_i)^\top \Sigma_c^{-1} (\mu_c - \mathbf{x}_i) + \log \det \Sigma_c + B, \quad (11)$$

where  $B$  denotes terms not dependent on the assignment of data points to clusters and thus irrelevant for the objective. Thus the above cluster update can be written as:

$$\begin{aligned} q^{(t)}(c|i) &= \frac{\log q^{(t-1)}(c) \exp \left[ -(\mu_c - \mathbf{x}_i)^\top \Sigma_c^{-1} (\mu_c - \mathbf{x}_i) \right]}{\det \Sigma_c \frac{Z(\mathbf{x}, \beta)}{Z(\mathbf{x}, \beta)}} \\ &= \frac{\log q^{(t-1)}(c) \exp \left[ -(\mu_c - \mathbf{x}_i)^\top \Sigma_c^{-1} (\mu_c - \mathbf{x}_i) \right]}{\det \Sigma_c \sum_c \exp \left[ -(\mu_c - \mathbf{x}_i)^\top \Sigma_c^{-1} (\mu_c - \mathbf{x}_i) \right]}. \end{aligned} \quad (12)$$

The next step is to update  $\mu_c$  to minimize the KL-divergence between  $g(\mathbf{x}|c)$  and  $p(\mathbf{x}|c)$ :

$$\begin{aligned} \text{KL}[q(\mathbf{x}|c)|g(\mathbf{x}|c)] &= - \int q(\mathbf{x}|c) \log g(\mathbf{x}|c) d\mathbf{x} - H[q(\mathbf{x}|c)] \\ &= - \frac{1}{n_c} \int \sum_{i \in S_c} \mathcal{N}(\mathbf{x}; \mathbf{x}_i, \epsilon^2) \log g(\mathbf{x}|c) d\mathbf{x} - H[q(\mathbf{x}|c)] \\ &\approx - \frac{1}{n_c} \sum_{i \in S_c} \log g(\mathbf{x}_i|c) - H[q(\mathbf{x}|c)] \end{aligned} \quad (13)$$

Minimizing the above w.r.t.  $\mu_c$  gives:

$$\mu_c^{(t)} = \frac{1}{N} \sum_{i=1}^N q(c|i)x_i = \sum_{i=1}^N \frac{\log q^{(t-1)}(c)}{N \det \Sigma_c} \frac{\exp [-(\mu_c - \mathbf{x}_i)^\top \Sigma_c^{-1}(\mu_c - \mathbf{x}_i)]}{\sum_c \exp [-(\mu_c - \mathbf{x}_i)^\top \Sigma_c^{-1}(\mu_c - \mathbf{x}_i)]} \mathbf{x}_i. \quad (14)$$

By properly re-arranging the above terms and writing them into a compact matrix form, the relationship between the IB approach and self-attention would become clearer. Assume  $\Sigma_c = \Sigma$  is shared across all the clusters. Assume  $\mu_c$  are normalized w.r.t.  $\Sigma_c^{-1}$ , i.e.,  $\mu_c^\top \Sigma_c^{-1} \mu_c = 1$ .

$$\mu_c^{(t)} = \sum_{i=1}^N \frac{\log [n_c/N]}{N \det \Sigma} \frac{\exp \left[ \frac{\mu_c^\top \Sigma^{-1} \mathbf{x}_i}{1/2} \right]}{\sum_c \exp \left[ \frac{\mu_c^\top \Sigma^{-1} \mathbf{x}_i}{1/2} \right]} \mathbf{x}_i. \quad (15)$$

Define  $Z = [\mu_1^{(t)\top}; \dots; \mu_N^{(t)\top}]$ ,  $V = [\mathbf{x}_1, \dots, \mathbf{x}_N] W_V$ ,  $K = [\mu_1^{(t-1)}, \dots, \mu_N^{(t-1)}]$ ,  $Q = \Sigma^{-1}[\mathbf{x}_1, \dots, \mathbf{x}_N]$ . Define  $d = 1/2$ . Then the above update (15) can be written as:

$$Z = \text{Softmax} \left( \frac{Q^\top K}{d} \right) V. \quad (16)$$

Here the softmax normalization is applied along the row direction. Thus we conclude the proof for Proposition 2.1.

Proposition 2.1 can be proved by following the above road map.

## A.2. Implementation details

**Architecture implementation** When comparing to other state-of-the-art methods, we add in a depthwise convolution layer in the MLP block following the practice in previous methods (Xie et al., 2021; El-Nouby et al., 2021). For the shortcut connection, we multiply the residual path with a learnable parameter to stabilize the training, following the same practice in (Touvron et al., 2021b).

**ImageNet classification** For all the experiments and ablation studies, the models are pretrained on ImageNet-1K if not specified additionally. The training recipes follow the one used in (Touvron et al., 2021a) for both the baseline model and our proposed FAN model family. Specifically, we train FAN for 300 epochs using AdamW with a learning rate of  $2e-3$ . We use 5 epochs to linearly warmup the model. We adopt a cosine decaying schedule afterward. We use a batch size of 2048 and a weight decay of 0.05. We adopt the same data augmentation schemes as (Touvron et al., 2021a) including Mixup, Cutmix, RandAugment, and Random Erasing. We use Exponential Moving Average (EMA) to speed up the model convergence in a similar manner as timm library (Wightman, 2019). For the image classification tasks, we also include two class attention blocks at the top layers as proposed by Touvron et al..

**Semantic segmentation and object detection** For FAN-ViT, we follow the same decoder proposed in semantic transformer (SETR) (Zheng et al., 2021) and the same training setting used in Segformer (Xie et al., 2021). For object detection, we finetune the faster RCNN (Ren et al., 2015) with 2x multi-scale training. The resolution of the training image is randomly selected from  $640 \times 640$  to  $896 \times 896$ . We use a deterministic image resolution of size  $896 \times 896$  for testing.

For FAN-Swin and FAN-Hybrid, We finetune Mask R-CNN (He et al., 2017) on the COCO dataset. Following Swin Transformer (Liu et al., 2021), we use multi-scale training, AdamW optimizer, and 3x schedule. The codes are developed using MMSegmentation (Contributors, 2020) and MMDetection (Chen et al., 2019) toolbox.

**Corruption dataset preparation** For ImageNet-C, we directly download it from the mirror image provided by Hendrycks & Dietterich. For Cityscape-C and COCO-C, we follow Kamann & Rother and generate 16 algorithmically generated corruptions from noise, blur, weather and digital categories.

**Evaluation metrics** For ImageNet-C, we use *retention* as a main metric to measure the robustness of the model which is defined as  $\frac{\text{ImageNet-C Acc.}}{\text{ImageNet Clean Acc.}}$ . It measures how much accuracy can be reserved when evaluated on ImageNet-C dataset.

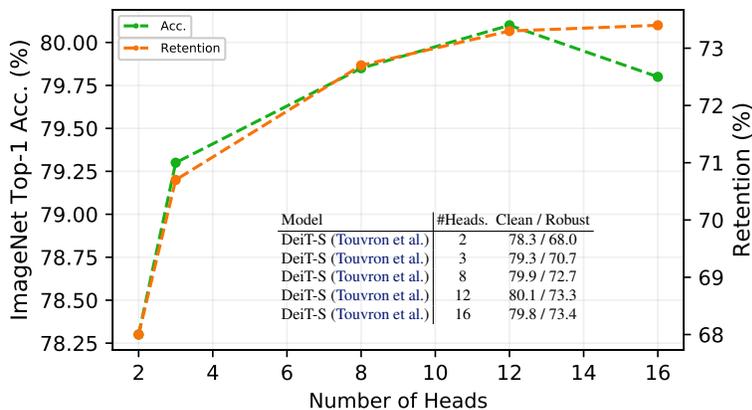


Figure 7. Impacts of head number on model robustness.

When comparing with other models, we also report the mean corruption error (mCE) in the same manner defined in the ImageNet-C paper (Hendrycks & Dietterich, 2019). The evaluation code is based on timm library (Wightman, 2019). For semantic segmentation and object detection, we load the ImageNet-1k pretrained weights and finetune on Cityscapes and COCO clean image dataset. Then we directly evaluate the performance on Cityscapes-C and COCO-C. We report semantic segmentation performance using mean Intersection over Union (mIoU) and object detection performance using mean average precision (mAP).

### A.3. Impact of head numbers

### A.4. Detailed benchmark results on corrupted images on classification, segmentation and detection

The by category robustness of selected models and FAN models are shown in Tab. 12, Tab. 13 and Tab. 14 respectively. As shown, the strong robustness of FAN is transferrable to all downstreaming tasks.

### A.5. Architecture details of FAN-Swin and FAN-Hybrid

For FAN-Swin architecture, we follow the same macro architecture design by only replacing the conventional self-attention module with the efficient window shift self-attention in the same manner as proposed in the Swin transformer (Liu et al., 2021). For the FAN-Hybrid architecture, we use three convolutional building blocks for each stage in the same architecture as proposed in ConvNeXt (Liu et al., 2022).

### A.6. Feature clustering and visualizations

To cluster the token features, we first normalize the tokens taken from the second last block’s output with a SoftMax function. We then calculate a self-correlation matrix based on the normalized tokens and use it as the affinity matrix for spectral clustering. Figure 9 provides more visualization on clustering results of token features from our FAN, ViT and CNN models. The visualization on Cityscape is shown in Figure 8.

## Understanding The Robustness in Vision Transformers

**Table 12. Comparison of model robustness on ImageNet-C (%).** FAN shows stronger robustness than other models under all the image corruption settings. ‘ResNet-50\*’ denotes our reproduced results with the same training and augmentation recipes for fair comparison.

Model	Param.	Average	Blur				Noise				Digital				Weather			
			Motion	Defoc	Glass	Gauss	Gauss	Impul	Shot	Speck	Contr	Satur	JPEG	Pixel	Bright.	Snow	Fog	Frost
Mobile Setting (< 10M)																		
ResNet-18 (He et al.)	11M	32.7	29.6	28.0	22.9	32.0	22.7	17.6	20.8	27.7	30.8	52.7	46.3	42.3	58.8	24.1	41.7	28.2
MobileNetV2 (Sandler et al.)	4M	35.0	33.4	29.6	21.3	32.9	24.4	21.5	23.7	32.9	57.6	49.6	38.0	62.5	28.4	45.2	37.6	28.3
EfficientNet-B0 (Tan & Le)	5M	41.1	36.4	26.8	26.9	39.3	39.8	38.1	47.1	39.9	65.2	58.2	52.1	69.0	37.3	55.1	44.6	37.4
PVT-V2-B0 (Wang et al.)	3M	36.2	30.8	24.9	34.0	35.8	33.1	35.2	44.2	50.6	59.3	50.8	36.6	61.9	38.6	50.7	45.9	41.8
PVT-V2-B1 (Wang et al.)	13M	51.7	45.7	41.3	30.5	43.9	48.1	46.2	46.6	55.0	57.6	68.6	59.9	50.2	71.0	49.8	56.8	53.0
FAN-T-ViT-P16 (Ours)	7M	<b>57.5</b>	52.4	<b>48.3</b>	<b>37.4</b>	<b>51.5</b>	54.8	54.7	53.1	60.2	<b>66.6</b>	72.8	62.7	56.7	74.3	<b>55.5</b>	<b>61.4</b>	<b>53.6</b>
FAN-T-Hybrid-P16 (Ours)	8M	57.4	<b>52.6</b>	46.7	34.3	50.3	<b>55.5</b>	<b>55.8</b>	<b>54.5</b>	<b>61.4</b>	65.8	<b>73.3</b>	<b>63.8</b>	<b>47.9</b>	<b>74.5</b>	55.0	61.4	52.8
GPU Setting (20M+)																		
ResNet-50* (He et al.)	25M	50.6	42.1	40.1	27.2	42.2	42.2	36.8	41.0	50.3	51.7	69.2	59.3	51.2	71.6	38.5	53.9	42.3
ViT-S (Dosovitskiy et al.)	22M	54.2	49.7	45.2	38.4	48.0	50.2	47.6	49.0	57.5	58.4	70.1	61.6	57.3	72.5	51.2	50.6	57.0
DeiT-S (Touvron et al.)	22M	58.1	52.6	48.9	38.1	51.7	57.2	55.0	54.7	60.8	63.7	71.8	64.0	58.3	73.6	55.1	61.1	60.7
FAN-S-ViT (Ours)	28M	64.5	<b>61.4</b>	<b>56.3</b>	<b>45.6</b>	<b>58.7</b>	62.1	63.0	61.1	67.1	<b>70.9</b>	77.1	<b>69.4</b>	<b>63.5</b>	78.4	<b>63.5</b>	<b>68.2</b>	61.2
FAN-S-Hybrid (Ours)	26M	<b>64.7</b>	60.8	56.0	44.5	58.6	<b>65.6</b>	<b>66.2</b>	<b>64.8</b>	<b>69.7</b>	67.5	<b>77.4</b>	68.7	61.0	<b>78.4</b>	63.2	66.1	<b>62.4</b>
GPU Setting (50M+)																		
ResNet-101 (He et al.)	45M	59.2	57.0	51.9	35.6	55.0	51.9	51.2	51.2	61.2	67.8	75.5	67.3	59.9	53.6	66.2	66.4	56.4
ViT-B* (Dosovitskiy et al.)	88M	59.7	60.2	55.6	50.0	57.6	54.9	52.9	53.2	62.0	52.3	71.5	68.7	71.7	74.9	52.8	57.1	41.7
DeiT-B (Touvron et al.)	89M	62.7	56.7	52.2	43.6	55.1	64.9	63.5	61.2	65.7	68.2	74.6	66.9	61.7	76.2	59.7	68.2	64.9
Swin-S (Liu et al.)	50M	60.4	56.7	51.4	34.8	53.4	60.07	58.4	57.8	62.3	65.9	73.8	66.4	62.4	76.0	55.9	67.4	60.7
FAN-B-ViT (Ours)	54M	67.0	64.2	58.4	49.7	60.8	66.0	67.3	65.0	69.8	72.9	78.1	71.2	66.9	79.3	64.5	70.9	62.8
FAN-B-Hybrid (Ours)	50M	66.4	62.5	58.0	47.2	60.9	67.6	67.9	67.1	71.2	70.8	78.0	69.3	62.1	78.9	64.8	69.8	63.3
FAN-B-Hybrid-IN22K (Ours)	50M	<b>70.5</b>	<b>67.4</b>	<b>62.9</b>	<b>55.6</b>	<b>65.4</b>	<b>70.3</b>	<b>71.6</b>	<b>70.1</b>	<b>73.8</b>	<b>74.1</b>	<b>79.8</b>	<b>74.3</b>	<b>79.8</b>	<b>81.0</b>	<b>70.2</b>	<b>72.2</b>	<b>65.4</b>
GPU Setting (80M+)																		
DeiT-B (Touvron et al.)	86M	59.7	60.22	55.6	50.0	57.6	54.9	52.9	53.2	62.0	52.3	71.5	68.7	71.7	74.9	52.9	57.1	54.1
Swin-B-IN22K (Liu et al.)	88M	68.6	66.1	62.1	48.2	63.2	67.3	66.2	66.4	70.5	71.7	77.8	73.5	74.0	80.3	66.2	74.0	66.9
ConvNeXt-B (Liu et al.)	89M	63.6	59.6	52.9	39.2	55.2	65.5	64.8	63.7	66.7	69.9	76.2	68.9	64.6	77.8	59.2	66.7	64.3
FAN-L-ViT (Ours)	81M	67.7	64.6	58.8	49.6	61.1	66.8	68.5	65.6	70.1	72.5	78.4	71.3	69.8	79.7	66.5	71.5	64.8
FAN-L-Hybrid (Ours)	77M	68.3	65.1	59.2	49.2	61.9	70.1	71.1	69.4	72.7	72.4	77.6	71.8	66.6	79.6	65.6	71.3	65.7
FAN-L-Hybrid-IN22K (Ours)	77M	<b>73.6</b>	<b>71.2</b>	<b>67.5</b>	<b>58.9</b>	<b>69.3</b>	<b>73.9</b>	<b>75.1</b>	<b>73.4</b>	<b>76.6</b>	<b>76.4</b>	<b>81.6</b>	<b>76.8</b>	74.0	<b>82.5</b>	<b>73.6</b>	<b>74.3</b>	<b>69.6</b>

**Table 13. Comparison of Model Robustness on Cityscapes-C (%).** FAN shows stronger robustness than both CNN and transformer models, for all the image corruption settings. ‘DLv3+’ refer to DeepLabv3+ (Chen et al., 2018). The mIoUs of compared CNN models are adopted from (Kamann & Rother, 2020). The mIoU of ConvNeXt, DeiT, Swin and SegFormer models are our reproduced results.

Model	Average	Blur				Noise				Digital				Weather			
		Motion	Defoc	Glass	Gauss	Gauss	Impul	Shot	Speck	Bright	Contr	Satur	JPEG	Snow	Spatt	Fog	Frost
DLv3+ (R50)	36.8	58.5	56.6	47.2	57.7	6.5	7.2	10.0	31.1	58.2	54.7	41.3	27.4	12.0	42.0	55.9	22.8
DLv3+ (R101)	39.4	59.1	56.3	47.7	57.3	13.2	13.9	16.3	36.9	59.2	54.5	41.5	37.4	11.9	47.8	55.1	22.7
DLv3+ (X65)	42.7	63.9	59.1	52.8	59.2	15.0	10.6	19.8	42.4	65.9	59.1	46.1	31.4	19.3	50.7	63.6	23.8
DLv3+ (X71)	42.5	64.1	60.9	52.0	60.4	14.9	10.8	19.4	41.2	68.0	58.7	47.1	40.2	18.8	50.4	64.1	20.2
ICNet (Zhao et al.)	28.0	45.8	44.6	47.4	44.7	8.4	8.4	10.6	27.9	41.0	33.1	27.5	34.0	6.3	30.5	27.3	11.0
FCN8s (Long et al.)	27.4	42.7	31.1	37.0	34.1	6.7	5.7	7.8	24.9	53.3	39.0	36.0	21.2	11.3	31.6	37.6	19.7
DilatedNet (Yu & Koltun)	30.3	44.4	36.3	32.5	38.4	15.6	14.0	18.4	32.7	52.7	32.6	38.1	29.1	12.5	32.3	34.7	19.2
ResNet-38	32.6	54.6	45.1	43.3	47.2	13.7	16.0	18.2	38.3	60.0	50.6	46.9	14.7	13.5	45.9	52.9	22.2
PSPNet (Zhao et al.)	34.5	59.8	53.2	44.4	53.9	11.0	15.4	15.4	34.2	60.4	51.8	30.6	21.4	8.4	42.7	34.4	16.2
ConvNeXt-T (Liu et al.)	54.4	64.1	61.4	49.1	62.1	34.9	31.8	38.8	56.7	76.7	68.1	76.0	51.1	25.0	58.7	74.2	35.1
SETR (DeiT-S) (Zheng et al.)	55.5	61.8	61.0	59.2	62.1	36.4	33.8	42.2	61.2	73.1	63.8	69.1	49.7	41.2	60.8	63.8	32.0
Swin-T (Liu et al.)	47.5	62.1	61.0	48.7	62.2	22.1	24.8	25.1	42.2	75.8	62.1	75.7	33.7	19.9	56.9	72.1	30.0
SegFormer-B0 (Xie et al.)	48.9	59.3	58.9	51.0	59.1	25.1	26.6	30.4	50.7	73.3	66.3	71.9	31.2	22.1	52.9	65.3	31.2
SegFormer-B1 (Xie et al.)	52.6	63.8	63.5	52.0	29.8	23.3	35.4	56.2	76.3	70.8	74.7	36.1	56.2	28.3	60.5	70.5	36.3
SegFormer-B2 (Xie et al.)	55.8	68.1	67.6	58.8	68.1	23.8	23.1	27.2	47.0	79.9	76.2	78.7	46.2	34.9	64.8	76.0	42.1
FAN-T-Hybrid (Ours)	57.9	67.1	66.0	57.2	66.6	33.2	34.3	36.2	55.6	80.8	72.1	79.1	54.3	30.6	66.1	78.2	43.8
FAN-S-Hybrid (Ours)	66.4	68.6	68.9	61.0	70.0	57.5	61.3	62.2	71.5	80.5	74.9	79.4	62.1	47.4	70.8	77.9	48.8
FAN-B-Hybrid (Ours)	67.3	70.0	69.0	64.3	70.3	55.9	60.4	61.1	70.9	81.2	76.1	80.0	57.0	<b>54.8</b>	72.5	78.4	52.3
FAN-L-Hybrid (Ours)	<b>68.5</b>	<b>70.0</b>	<b>69.9</b>	<b>65.3</b>	<b>71.6</b>	<b>60.0</b>	<b>64.5</b>	<b>63.3</b>	<b>71.6</b>	<b>81.4</b>	<b>76.2</b>	<b>80.1</b>	62.3	53.1	<b>73.9</b>	<b>78.9</b>	<b>54.4</b>

Table 14. Comparison of model robustness on COCO-C (%). FAN shows stronger robustness than other models.

Model	Average	Blur				Noise				Digital				Weather			
		Motion	Defoc	Glass	Gauss	Gauss	Impul	Shot	Speck	Bright.	Contr	Satur	JPEG	Snow	Spatter	Fog	Frost
ResNet-50(Faster-RCNN) (Ren et al.)	21.3	16.6	18.2	11.4	19.9	17.1	17.3	14.0	22.5	35.0	21.8	33.7	18.2	18.5	26.5	31.6	23.1
ResNet-101 (Faster-RCNN) (Ren et al.)	23.3	18.8	20.2	13.8	22.0	19.2	16.2	19.2	24.4	37.1	23.7	35.7	20.0	20.4	28.8	33.9	24.9
SWIN-T (Liu et al.)	29.3	24.2	25.8	18.2	27.8	24.6	23.7	24.8	30.5	42.1	31.2	41.0	26.6	26.1	36.3	40.6	30.8
DeiT-S (Touvron et al.)	26.9	23.6	23.7	21.8	25.2	22.0	21.0	22.3	27.7	37.0	25.2	35.6	29.7	25.9	31.9	34.4	28.4
FAN-T-Hybrid (Ours)	29.7	24.2	25.8	18.8	27.4	23.5	22.7	24.0	30.3	42.4	33.5	41.2	27.9	28.3	36.5	41.0	32.8
FAN-S-Hybrid (Ours)	35.5	29.1	29.5	23.5	31.5	31.6	32.1	32.3	37.1	33.3	46.1	40.4	45.3	33.5	42.4	45.9	35.5
FAN-B-Hybrid (Ours)	39.0	31.8	32.1	27.4	34.2	34.6	35.5	35.4	40.6	50.4	43.3	49.6	36.9	39.4	46.4	49.6	41.8
FAN-L-Hybrid (Ours)	40.6	33.0	33.2	28.7	35.6	37.1	38.0	37.5	42.2	51.3	46.0	50.3	39.2	41.1	47.5	50.3	43.0
FAN-B-Hybrid-IN22k (Ours)	40.6	33.1	33.1	28.5	35.5	36.7	38.0	37.3	42.3	51.5	45.6	50.7	39.0	41.2	48.1	50.7	43.0
FAN-L-Hybrid-IN22k (Ours)	<b>42.0</b>	<b>34.1</b>	<b>34.5</b>	<b>30.8</b>	<b>37.1</b>	<b>38.5</b>	<b>39.7</b>	<b>39.1</b>	<b>43.6</b>	<b>52.1</b>	<b>47.6</b>	<b>51.3</b>	<b>40.8</b>	<b>42.3</b>	<b>49.2</b>	<b>51.8</b>	<b>44.3</b>

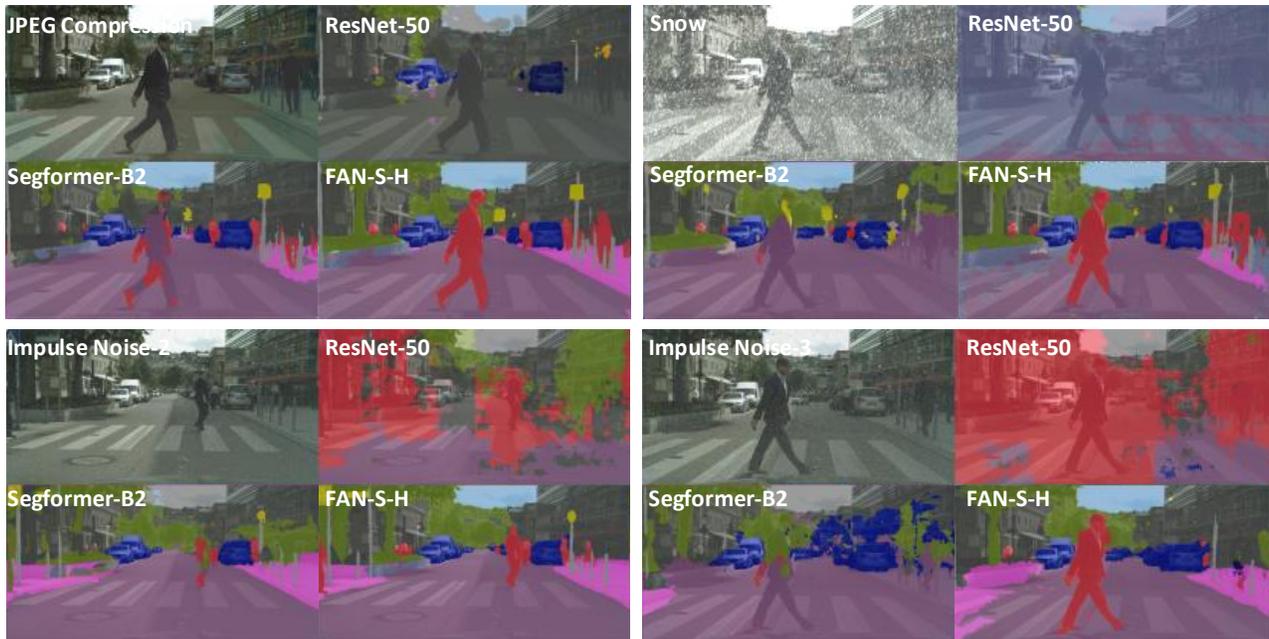


Figure 8. Visualization on Cityscapes. A video demonstration is available with external player

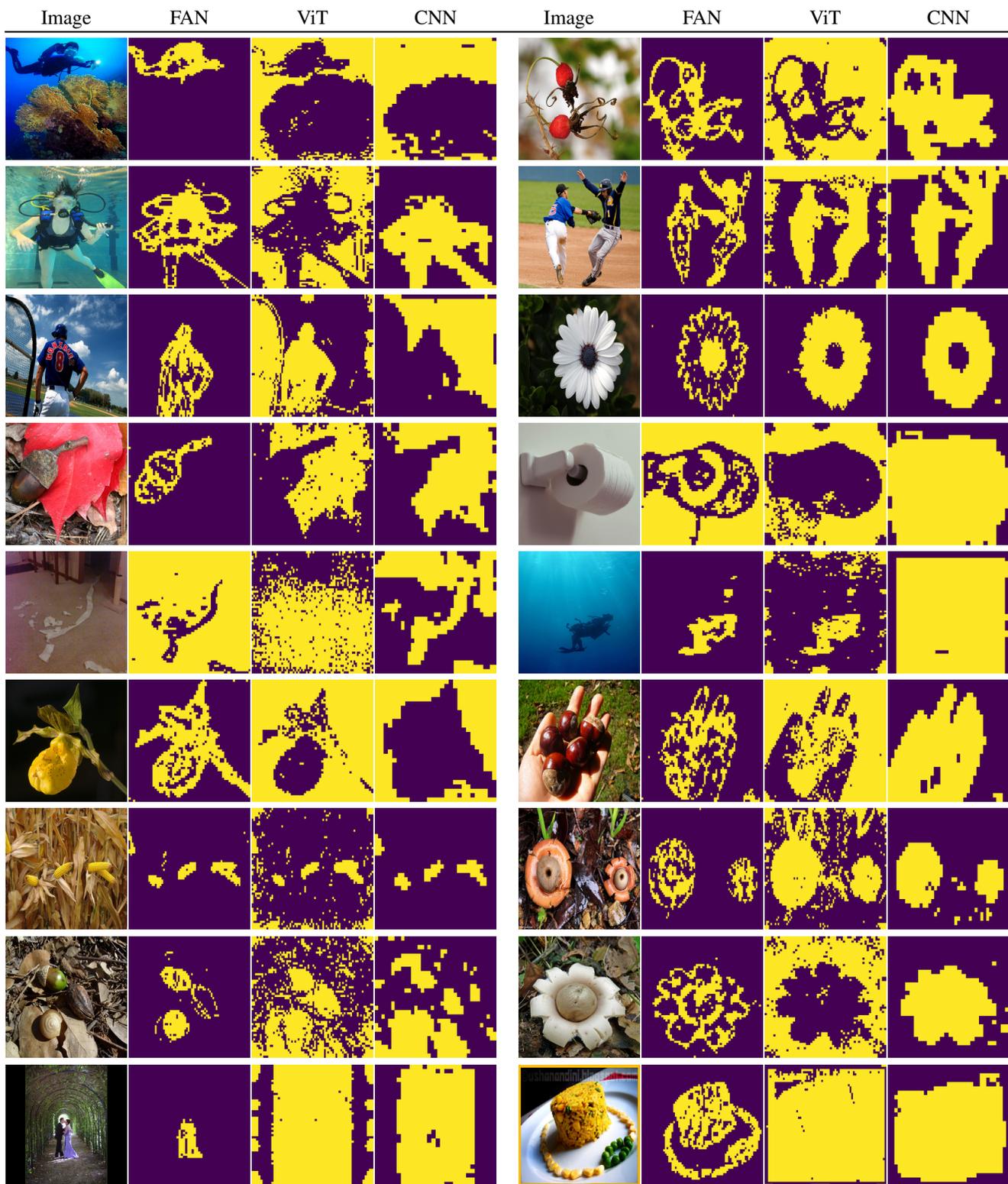


Figure 9. clustering visualization. Our FAN model provides much clearer clusters that feature important regions of foreground objects.