Fully Attentional Networks with Self-emerging Token Labeling

Bingyin Zhao[1,2*] Zhiding Yu[1†] Shiyi Lan[1] Yutao Cheng[3] Anima Anandkumar[1,4]
Yingjie Lao[2] Jose M. Alvarez[1]
[1]NVIDIA [2]Clemson University [3]Fudan University [4]Caltech

Abstract

Recent studies indicate that Vision Transformers (ViTs) are robust against out-of-distribution scenarios. In particular, the Fully Attentional Network (FAN) - a family of ViT backbones, has achieved state-of-the-art robustness. In this paper, we revisit the FAN models and improve their pre-training with a self-emerging token labeling (STL) framework. Our method contains a two-stage training framework. Specifically, we first train a FAN token labeler (FAN-TL) to generate semantically meaningful patch token labels, followed by a FAN student model training stage that uses both the token labels and the original class label. With the proposed STL framework, our best model based on FAN-L-Hybrid (77.3M parameters) achieves 84.8% Top-1 accuracy and 42.1% mCE on ImageNet-1K and ImageNet-C, and sets a new state-of-the-art for ImageNet-A (46.1%) and ImageNet-R (56.6%) without using extra data, outperforming the original FAN counterpart by significant margins. The proposed framework also demonstrates significantly enhanced performance on downstream tasks such as semantic segmentation, with up to 1.7% improvement in robustness over the counterpart model.

1. Introduction

Vision Transformers (ViTs) [1] have recently achieved remarkable success in visual recognition tasks. Such success is not only attributed to their self-attention representation but also to the newly developed training recipes. For instance, refinements in training techniques such as strong data augmentation and knowledge distillation [2] greatly alleviate ViT's issue of being data-hungry and make them more accessible for training on ImageNet-1K.

Another important development in the training recipe is token labeling [3], where patch tokens are assigned with labels to ViTs in a dense manner. In some sense, token labeling can also be considered as an alternative form of hard knowledge distillation. However, the dense nature of token labeling allows ViTs to leverage more fine-grained information in an image and take different categories and object localization into account. Compared to traditional knowledge distillation methods, token labeling enables ViTs to exploit a wider range of information in the image, leading to more accurate results. The success of token labeling depends on carefully-designed token-level annotators (i.e., token-labelers) that can provide accurate location-specific information (i.e., token labels) to patch tokens. In [3], this is done by a special re-labeling process [4] using convolutional neural networks (CNNs) [5] pre-trained on ImageNet-1K. While Vision Transformers have shown great promise in representation learning, less exploration has been conducted on modeling them as token-labelers. This raises two interesting questions:

1. Can Transformer-based models self-produce meaningful token labels?

2. Can one improve the pre-training of ViTs with self-produced knowledge instead of external teachers?

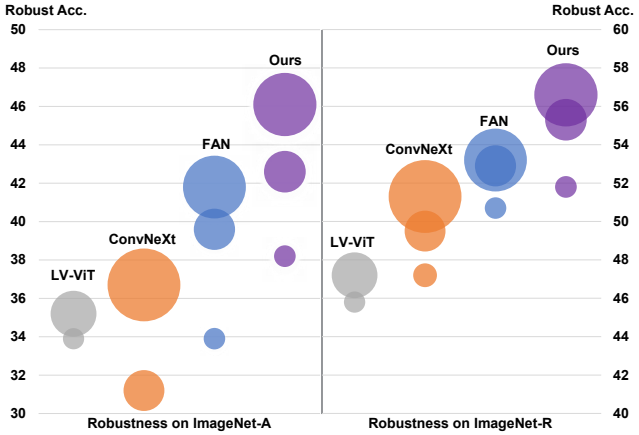

Figure 1. **Results of zero-shot robustness against ImageNet-A and ImageNet-R.** Models trained on ImageNet-1K with self-emerging token labels from FAN show superior robustness to the out-of-distribution data. Our best model (with only 77.3M parameters) achieves robust accuracy of 46.1% and 56.6% and sets a new record on ImageNet-A and ImageNet-R.

*Work done during an internship at NVIDIA.
†Corresponding author.

Our approach: In this paper, we aim to answer the above questions. We propose a self-emerging token labeling (STL) framework that employs the self-produced token labels by ViT token-labelers instead of relying on CNNs. Our work is built on the recently proposed Fully Attentional Network (FAN) [6] for two reasons. First, FAN exhibits excellent self-emerging visual grouping on token features, which can be leveraged to generate high-quality token labels. Second, FAN is a family of ViT backbones with state-of-the-art accuracy and robustness. We aim to further improve this family of powerful backbones through a principled token-labeling design and validate its effectiveness. Our contributions can be summarized as follows:

- Our work demonstrates that ViT models can be effective token-labelers. We propose a simple yet effective way to train a FAN token-labeler that can produce semantically meaningful token labels.
- We perform an in-depth analysis and show critical contributors to the accuracy of token labels. On top of the observations, we design a solution that retains more accurate token labels of the target object for improved model pre-training.
- Our models trained with STL set a new record on out-of-distribution datasets without using extra data than ImageNet-1K. Our best model achieves robust accuracy of 46.1% on ImageNet-A and 56.6% on ImageNet-R with only 77M parameters, as shown in Fig. 1.
- Experiments on downstream tasks demonstrate that the improved performance in backbone models is transferable to semantic segmentation and object detection.

Our STL framework is akin to the teacher-student training strategy introduced in knowledge distillation and consists of two stages:

First stage: We train a FAN token-labeler (FAN-TL) model to generate token-level annotations. Our task is essentially a "chicken or the egg" problem since there is no explicit supervision on how the token labels are generated. We tackle this by assigning supervising both the class token and the global average-pooled token. This produces semantically meaningful token labels as shown in Fig. 2(b).

Second stage: We train a FAN student model using the original class labels and the token labels from FAN-TL. Observing the imperfect quality of token labels, we introduce a token selection approach based on Gumbel-Softmax that adaptively selects tokens with high confidence. Labels of the selected tokens are of better quality and object grounding in general, leading to improved pre-training.

2. Related Work

2.1. Vision Transformers

Vision Transformers [1] are a family of visual recognition models built upon Transformers [7]. ViT splits an input image into a series of small patches, projects each as an embedding (a.k.a patch token) and appends with position embeddings. The resulting patch tokens and an extra learnable class token that aggregates global information for classification are then fed into a sequence of Transformer encoders consisting of multi-head self-attention and FFN blocks. A linear projection layer is appended to the class token to predict the class probabilities.

2.2. Fully Attentional Networks

Several concurrent works point out that ViTs exhibit excellent zero-shot robustness against out-of-distribution samples [8–11]. Some works propose to use negative data augmentation [12] and adversarial training [13, 14] to further enhance the robustness. Recently, FAN [15] was introduced as a family of ViT backbones with state-of-the-art accuracy and robustness. FAN inherits the self-attention blocks of plain ViT but additionally introduces a channel attention block that adopts an attention-based design that aggregates the cross-channel information in a more holistic manner, leading to improved representation.

2.3. Token Labeling

Token labeling [3] has been proposed to improve ViT pre-training. From the perspective of training strategy, it is similar to knowledge distillation [2, 16] since both adopt a teacher-student mode. It is also related to ReLabel [17], which provides images with multi-label annotations instead of single ones. However, both ReLabel and knowledge distillation depend on image-level labels as global supervision while token labeling assigns labels to each image patch token and supervises the student model in a dense manner. Token labeling is also inherently related to tokenization in BEiT [18], where an offline pre-trained discrete VAE is employed as the tokenizer to encode patches into visual tokens (i.e., code from a visual codebook). Different from token labels, these visual tokens do not possess explicit semantically meanings since they originate from an unsupervisedly trained codebook. Our method differs from prior token labeling and distillation methods where pre-trained convolutional neural networks are widely used as the token-labeler. Instead, our approach to unifies both teacher and student under homogeneous Vision Transformer architectures to generate high-quality token labels.

2.4. Emerging Properties of ViTs

It was found that the localization of objects emerges in image classification with CNNs. This interesting phe-

nomenon, also known as class activation maps (CAM) [19], lays the foundation for token labeling. Recent studies reveal that ViTs demonstrate excellent capability for object localization without explicit supervision. For instance, DINO [20] shows that self-supervised ViT features generate semantically meaning object segmentation. Methods like GroupViT [21] show that semantic segmentation emerges in ViTs using only text supervision. Similarly, FAN reveals that the robustness of ViT models is correlated to their excellent visual grouping capability. This feature motivates us to develop self-emerging token labeling on top of the FAN models.

3. Method

As mentioned, we propose a self-emerging token labeling (STL) framework that uses self-produced token labels to improve ViT pre-training. STL consists of two stages: 1) training an effective token-labeler, and 2) training a student model with self-emerging token labels.

In the first stage, we train a FAN token-labeler (FAN-TL) to generate high-quality token labels. As discussed in Sec. 1 and Sec. 2, FAN demonstrates strong robustness and capability to obtain semantically meaningful visual grouping that can correctly captures the object gestalt. These great features of FAN allow us to obtain high quality token labels without bells and whistles. In the second stage, we then train a FAN student model with image-level labels and self-emerging token labels from FAN-TL. At high level, FAN-TL and FAN student models follow the same architectural design as FAN but slightly modify the structure of patch tokens. We attach a linear layer to each patch token to accommodate token labels, similar to the classification head added to the class token in the original FAN and ViT design. The rest of this section describes the implementation details of the above two-stage pipeline.

3.1. Training FAN Token-Labelers

The original FAN employs the training paradigm that only takes image-level labels as supervision. We denote the input image as $\mathbf{I}$, the sequence of small patches as $[\mathbf{I}_{p_1}, \mathbf{I}_{p_2}, \dots \mathbf{I}_{p_N}]$, the output of FAN encoder as $[\mathbf{T}_{cls}, \mathbf{T}_{p_1}, \dots, \mathbf{T}_{p_N}]$, where N is the number of patch tokens, $\mathbf{T}_{cls}$ represents the class token and $\mathbf{T}_{p_1}, \dots, \mathbf{T}_{p_N}$ represent the patch tokens, respectively. The training objective can be mathematically expressed as follows:

$$\mathcal{L} = \mathcal{H}(\mathbf{T}_{cls}, \mathbf{Y}_{cls}), \tag{1}$$

where $\mathcal{H}(\cdot)$ is the softmax cross entropy loss and $\mathbf{Y}_{cls}$ is the image-level label of the corresponding class.

The key to token labeling is the generation of accurate token labels that provide location-specific information. However, following the conventional training paradigm in Eq. 1,

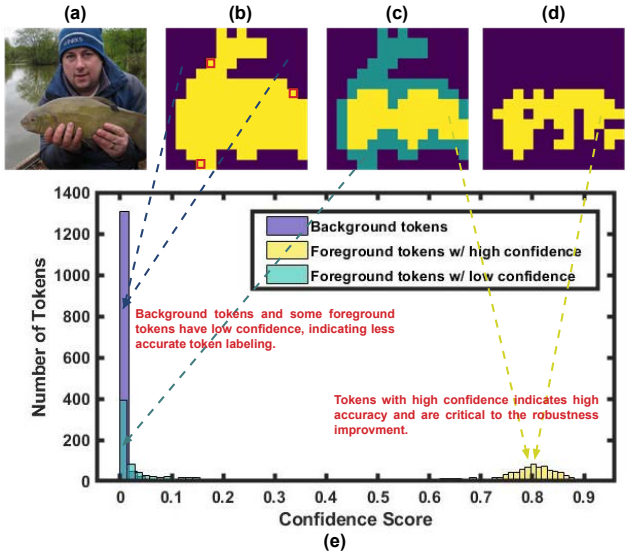

Figure 2. **Illustration of token labels generated by FAN-TL and the token label confidence score distribution.** (a). original image (class: "tench"), (b). binary color map of token labels (yellow: tokens classified as "tench", dark blue: tokens not classified as "tench") (c). trinary color map of token labels (cyan: *foreground tokens* with low confidence, yellow: *foreground tokens* with high confidence), (d). binary color map of *foreground tokens* selected by Gumbel-Softmax, (e). token label confidence score distribution of a batch of 16 images.

token outputs of FAN models are not semantically well-guided since they are not supervised during training. We propose a simple yet effective method to address this issue. Our idea is inspired by the intriguing phenomenon in ViTs training that meaningful object segmentation naturally emerges [20]. Unlike the self-supervised training in DINO, we leverage FAN's strong capability of visual grouping [22] and devise a fully supervised approach that allows FAN to generate accurate and semantically meaningful token labels. We perform global average-pooling on all patch tokens and then simultaneously assign the class label to the class and the average-pooled tokens. The training objective of FAN-TL can be written as follows:

$$\mathcal{L} = \mathcal{H}(\mathbf{T}_{cls}, \mathbf{Y}_{cls}) + \alpha \cdot \mathcal{H}(\frac{1}{N}\sum_{i=1}^{N}\mathbf{T}_{p_i}, \mathbf{Y}_{cls}), \tag{2}$$

where α weights the importance of two loss functions. We set α to 1 in our experiments. We demonstrate a visualization example of the token labels generated by FAN-TL in Fig. 2(b). The yellow area (i.e., foreground) represents the tokens with the same labels as the image-level label (we term *foreground tokens*). In contrast, the dark blue area (i.e., background) represents the tokens with different labels from the image-level label (we term *background tokens*). It can be seen that the self-emerging token labels show a meaningful segmentation of the target object "tench".

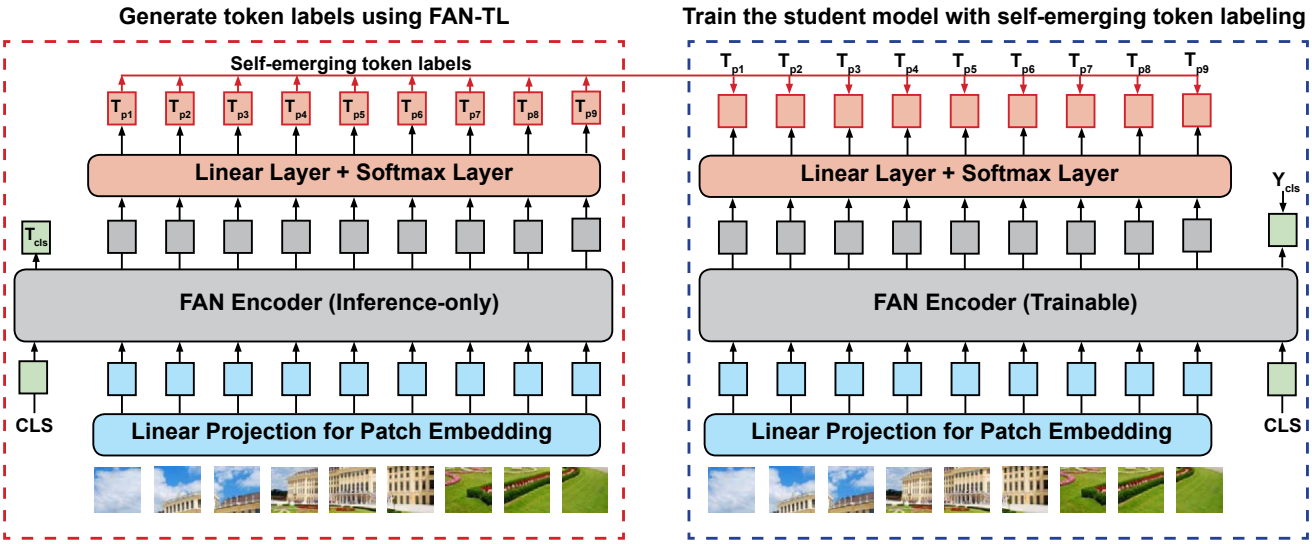

Figure 3. **Illustration of Stage 2: Training student models with self-emerging token labels.** In the training, token labels are generated by FAN-TL and assigned to patch tokens of student models. We incorporate the token labels and class labels to train student models jointly. FAN-TL can self-identify the incorrect token labels upon the confidence score. Tokens with high confidence scores offer a more accurate segmentation of objects and are crucial to robustness improvement. By applying spatial-only data augmentation to the inputs and Gumbel-Softmax to the token outputs of FAN-TL, we obtain the most accurate and critical token labels.

3.2. Training Student Models with STL

Training student models is straightforward. As illustrated in Fig. 3, we take the token outputs of FAN-TL and assign them as the labels for patch tokens of student models. We then incorporate token labels with class labels and jointly optimize the loss on the class token and all patch tokens. The training objective is shown as follows:

$$\mathcal{L} = \mathcal{H}(\mathbf{T}_{cls}, \mathbf{Y}_{cls}) + \beta \cdot \frac{1}{N} \sum_{i=1}^{N} \mathcal{H}(\mathbf{T}_{p_i}, \mathcal{F}(\mathbf{I}_{p_i})) \quad (3)$$

where $\mathcal{F}(\cdot)$ represents the patch token outputs of FAN-TL and β is the hyper-parameter to balance two loss functions. β is set to 1 in our experiments. Note the correctness of token labels (especially the *foreground tokens*) is critical since wrong labels introduce worthless local information. Thus, we propose several implementation tricks to generate and retain highly accurate token labels based on the following two observations.

Observation 1: Data augmentations remarkably affect the accuracy of token labels. Recent studies [2, 23] indicate data augmentations improve the classification accuracy and robustness of ViT models. However, most augmentations undermine the accuracy of token labels. For example, augmentations in RandAug [24], such as posterize and solarize, alter pixel values, CutOut [25] randomly erases a part of images, Mixup [26] and CutMix [27] mix content from two images, which all make it difficult for FAN-TL to assign correct labels to patch tokens. To this end, we disable these data augmentations and keep spatial-only ones (i.e., flip, rotate, shear and translation) for FAN-TL when generating token labels. Note that we still apply full data augmentations for the student model. The concrete settings of data augmentations are summarized in Table 1. Moreover, the strong data augmentations can be regarded as noises akin to corruptions and perturbations in out-of-distribution datasets. The input data augmentation discrepancy between FAN-TL and the student model enables the self-emerging token labels to provide clean and correct information regardless of the noise, thus improving the model's robustness.

Model (Stage)	Spatial	RandAug	CutOut	MixUp	CutMix
FAN-TL (Stage1)	✓	✓	✓	✓	✓
Student (Stage2)	✓	✓	✓	✓	✓
FAN-TL (Stage2)	✓	✗	✗	✗	✗

Table 1. **Detailed settings of data augmentations.** We apply spatial-only data augmentations to inputs of FAN-TL in stage 2 to improve the accuracy of token labels.

Observation 2: Not all the self-emerging token labels are correct, even with spatial-only data augmentations. For example, as shown in Fig. 2(b), some patches (red squares) that only contain "human" and "lake" are misclassified as the target object "tench" by FAN-TL. It is particularly crucial to ensure the accuracy of foreground token labels as they contain the target object of the image that dominates the prediction result. However, it is challenging to determine which labels are incorrect due to the lack of patch-level ground truth. Interestingly, FAN-TL can

self-identify these misclassified *foreground tokens* according to the token label confidence score, which may also be attributed to its visual grouping ability. The token label confidence score is defined as the maximal class probability of each token output. We find that tokens with correct labels tend to have higher confidence scores than those with incorrect labels. As shown in Fig. 2(c) and Fig. 2(e), the yellow area indicates the *foreground tokens* with high confidence scores ($0.7 \sim 0.9$). These tokens yield a highly accurate segmentation of the target object. Contrastly, the cyan area represents the *foreground tokens* with low confidence scores ($0 \sim 0.2$), which are coincidentally the ones with incorrect labels.

We attempt to assign highly accurate labels to the target object (i.e., *foreground tokens*). However, it is intractable to exhaustively examine the confidence score of each patch token for all the images as it significantly increases the training time and requires much more computational resources. Therefore, we propose a lightweight alternative by applying the Gumbel-Softmax [28] on the top of token outputs. Mathematically, it can be expressed as:

$$\mathbf{y}_i = \frac{e^{(log(\pi_i)+\mathcal{G}_i)/\tau)}}{\sum_{j=1}^{k} e^{(log(\pi_j)+\mathcal{G}_j)/\tau)}} \tag{4}$$

where $\mathbf{y}$ is the k-dimension softmax vector, π are class probabilities, $\mathcal{G} \sim Gumbel(0, 1)$ are i.i.d. samples drawn from the standard Gumbel distribution and τ is the softmax temperature. Token labels with high confidence scores remain unchanged after applying Gumbel-Softmax, while labels with low confidence scores are highly likely to change. As shown in Fig. 2(d), we preserve the correct token labels and eliminate the incorrect ones in a simple yet effective way, achieving high accuracy of *foreground tokens* labels and more precise segmentation of the object. Furthermore, since the training objective of the patch tokens side can be considered a self-training process [29–31], we convert the softmax outputs to "one-hot" probability distribution (i.e., hard labels) as [32] shows that the use of hard labels in self-training encourages the model's predictions to be high-confidence via entropy minimization [33]. Following the aforementioned spatial-only data augmentation, we can then rewrite the training objective of the student model:

$$\mathcal{L} = \mathcal{H}(\mathbf{T}_{cls}, \mathbf{Y}_{cls}) + \beta \cdot \frac{1}{N} \sum_{i=1}^{N} \mathcal{H}(\mathbf{T}_{p_i}, \hat{\mathcal{F}}(\hat{\mathbf{I}}_{p_i})) \tag{5}$$

where $\hat{\mathcal{F}}(\cdot)$ is the one-hot encoded Gumbel-Softmax outputs of FAN-TL and $[\hat{\mathbf{I}}_{p_1} \dots \hat{\mathbf{I}}_{p_N}]$ are image patches with the spatial-only data augmentation. Meanwhile, we notice that all *background tokens* (i.e., the dark blue area) also have low confidence scores. This is probably because these tokens neither contain apparent features associated with the target object nor clear features related to any other class in the dataset. Nevertheless, we keep all tokens (even those with low confidence scores) in training, following the practice in [3] as involving more tokens for loss computation yields better performance.

4. Experiments

4.1. Datasets and Evaluation Metrics

We evaluate our method on the image classification task and its transferability to downstream semantic segmentation and object detection tasks.

Datasets. For image classification, we test model performance and robustness on ImageNet-1K (IN-1K) [34], ImageNet-C (IN-C) [35], ImageNet-A (IN-A) [36] and ImageNet-R (IN-R) [37]. IN-C contains natural corruptions from noise, blur, weather, and digital categories and is widely used to evaluate model's robustness against shifted distribution data. IN-A and IN-R consist of images with different distributions from ImageNet training distribution, such as natural adversarial examples and pictures generated by artistic rendition, thus are widely used to measure the robustness against out-of-distribution data. For semantic segmentation and object detection, we evaluate models on Cityscapes (City) [38], Cityscapes-C (City-C) and COCO [39]. Similar to IN-C, City-C has corruptions from the same four categories.

Metrics. We adopt standard evaluation metrics for image classification: clean accuracy for IN-1K and robust accuracy for IN-C, IN-A and IN-R. We also report mean corruption error (mCE) [35] on IN-C. For semantic segmentation and object detection, we evaluate the model performance using the clean and robust mean Intersection over Union (mIoU) on City and City-C and the mean average precision (mAP) on COCO. Additionally, we use retention rate as the metric to reflect the resilience of the model robustness and fairly compare models with different capacities. The retention rate is defined as $R = \frac{\text{Robust Acc.}}{\text{Clean Acc.}}$.

4.2. Implementation Details

Experiments are conducted on 8 NVIDIA Tesla V100s and codes are built upon Pytorch [40], timm [41] library and MMSegmentation [42] toolbox. We adopt FAN-Hybrid as the model architecture for FAN-TL and student models. For image classification, we train the models on ImageNet-1K using AdamW optimizer with a learning rate of 4e-3 and batch size of 2048 for 350 epochs. We employ the cosine scheduler with a decay rate of 0.1 to adjust the learning rate every 30 epochs. The loss weight α in Eq. 2 and β in Eq. 5 are set to 1. We apply spatial, RandAug, CutOut, Mixup and CutMix data augmentation in the training and a label smoothing ratio of 0.9 to class and token labels. As discussed in Sec. 3, we apply spatial-only data augmentations

Model	Param./FLOPs	IN-1K	IN-C	Retention
ResNet18 [45]	11M/1.8G	69.9	32.7	46.8%
MBV2 [46]	4M/0.4G	73.0	35.0	47.9%
EffiNet-B0 [47]	5M/0.4G	77.5	41.1	53.0%
PVTV2-B0 [48]	3M/0.6G	70.5	36.2	51.3%
PVTV2-B1 [48]	13M/2.1G	78.7	51.7	65.7%
LV-ViT-T [3]	9M/2.1G	79.1	51.6	65.2%
FAN-T-Hybrid [15]	7M/3.5G	80.1	57.4	71.4%
STL (FAN-T-Hybrid)	8M/3.6G	79.9	**58.2**	**72.8%**
ResNet50 [45]	25M/4.1G	79.0	50.6	64.1%
DeiT-S [2]	22M/4.6G	79.9	58.1	72.7%
Swin-T [43]	28M/4.5G	81.3	55.4	68.1%
ConvNeXt-T [49]	29M/4.5G	82.1	59.1	71.9%
LV-ViT-S [3]	26M/6.6G	83.3	59.7	71.7 %
FAN-S-Hybrid [15]	26M/6.7G	83.5	64.7	77.5%
STL (FAN-S-Hybrid)	27M/6.8G	83.4	**65.5**	**78.5%**
Swin-S [43]	50M/8.7G	83.0	60.4	72.8%
ConvNeXt-S [49]	50M/8.7G	83.1	61.7	74.2%
LV-ViT-M [3]	56M/16.0G	84.0	62.0	73.8%
FAN-B-Hybrid [15]	50M/11.3G	83.9	66.4	79.1%
STL (FAN-B-Hybrid)	51M/11.4G	84.5	**68.2**	**80.7%**
DeiT-B [2]	89M/17.6G	81.8	62.7	76.7%
Swin-B [43]	88M/15.4G	83.5	60.4	72.3%
ConvNeXt-B [49]	89M/15.4G	83.8	63.0	75.2%
FAN-L-Hybrid [15]	77M/16.9G	84.3	68.3	81.0%
STL (FAN-L-Hybrid)	77M/17.0G	84.7	**68.8**	**81.2%**

Table 2. **Results on image classification.** We report clean and robust accuracy on ImageNet-1K and ImageNet-C. Retention rate is defined as $\frac{\text{Robust Acc.}}{\text{Clean Acc.}}$. LV-ViTs are vanilla ViTs trained with a CNN token-labeler. Our models trained with STL achieve superior robustness and retention rate in all cases. Meanwhile, our method also improves the clean accuracy of models with a larger capacity (e.g., FAN-B-Hybrid and FAN-L-Hybrid).

on inputs of FAN-TL and Gumbel-Softmax on patch token outputs to obtain accurate token labels. We employ pre-trained image classification models for semantic segmentation as encoders and the SegFormer [11] head as the decoder. We follow the same training recipe as SegFormer and train our models on Cityscapes using AdamW with a learning rate of 6e-5 and a batch size of 8 for 160K iterations. The learning rate scheduler is set to "poly" with a default factor of 1.0. Random resizing, flipping, and cropping are applied as data augmentations in training. For object detection, we follow the same practice as Swin Transformer [43] + Cascade Mask R-CNN [44] and employ AdamW (initial learning rate of 1e-4, weight decay of 0.05, and batch size of 16) to train our models on COCO for 36 epochs.

4.3. Results on Image Classification

We first show the performance of models trained with STL on the image classification task and compare them with other SOTA models in Table 2. To evaluate the zero-shot robustness against the distributional shift, all models are trained on ImageNet-1K data and directly used for inference on ImageNet-C without finetuning. We use the same model type for FAN-TL and the student model. It can be seen that Transformer-based models are more robust than CNN-based models in general. At all size levels, our models show superior robust accuracy and retention rate to other models, including the original FAN models trained solely with the class labels, indicating the effectiveness of STL in improving model robustness. Notably, our models surpass LV-ViTs [3] (i.e., vanilla ViT models trained with a CNN token-labeler) in both clean and robust accuracy by significant margins, which validates the importance of channel attention block in FAN and reveals the potential of self-emerging token labels from Transformer-based models.

Model	Params (M)	Clean	IN-A	IN-R	mCE (↓)
Swin-T [50]	28.3	81.2	21.6	41.3	59.6
ConvNext-T [49]	28.6	82.1	24.2	47.2	53.2
RVT-S [51]	23.3	81.9	25.7	47.7	51.4
XCiT-S12 [52]	26.3	81.9	25.0	45.5	51.5
LV-ViT-S [3]	26.0	83.3	33.9	45.8	52.9
FAN-S-Hybrid [15]	26.3	83.5	33.9	50.7	47.8
STL (FAN-S-Hybrid)	26.5	83.4	**38.2**	**51.8**	**47.3**
Swin-S [50]	50.0	83.4	35.8	46.6	52.7
ConvNext-S [49]	50.2	82.1	31.2	49.5	51.2
XCiT-S24 [52]	47.7	82.6	27.8	45.5	49.4
LV-ViT-M [3]	56.0	84.0	35.2	47.2	50.5
FAN-B-Hybrid [15]	50.4	83.9	39.6	52.9	45.2
STL (FAN-B-Hybrid)	50.9	84.5	**42.6**	**55.3**	**43.6**
Swin-B [50]	87.8	83.4	35.8	46.6	54.4
MAE-ViT-B [53]	86.0	83.6	35.9	48.3	51.7
ConvNext-B [49]	88.6	83.8	36.7	51.3	46.8
RVT-B [51]	91.8	82.6	28.5	48.7	46.8
DAT-AugReg-ViT [14]	86.0	81.5	30.2	47.3	44.7
FAN-L-Hybrid [15]	76.8	84.3	41.8	53.2	43.0
STL (FAN-L-Hybrid)	77.3	84.7	**46.1**	**56.6**	**42.5**

Table 3. **Results on out-of-distribution datasets.** We report the mean corruption error (mCE) for ImageNet-C, a lower value indicates better robustness. The improved robustness of models trained with STL is well generalized to out-of-distribution datasets and achieves even better robustness on ImageNet-A and ImageNet-R.

4.4. Robustness against Out-of-distribution Data

Token labels embed rich local information of image patches. We adopt spatial-only data augmentation and Gumbel-Softmax on FAN-TL to retain highly accurate labels for foreground tokens to ensure that self-emerging token labels always provide correct information for student models. The practice promotes the generalization performance of student models as they can make robust predictions even with different input data distributions. To verify this, we then evaluate model robustness against out-of-distribution data, and results are summarized in Table 3. Similarly, models are not fine-tuned for testing. We find that

Model	Encoder Size	City	City-C	Retention
DeepLabv3+ (R50) [54]	25.4M	76.6	36.8	48.0%
DeepLabv3+ (R101) [54]	47.9M	77.1	39.4	51.1%
DeepLabv3+ (X65) [54]	22.8M	78.4	42.7	54.5%
DeepLabv3+ (X71) [54]	-	78.6	42.5	54.1%
ICNet ([55])	-	65.9	28.0	42.5%
FCN8s ([56])	50.1M	66.7	27.4	41.1%
DilatedNet ([57])	-	68.6	30.3	44.2%
ResNet38 ([58])	-	77.5	32.6	42.1%
PSPNet ([59])	13.7M	78.8	34.5	43.8%
ConvNeXt-T ([49])	29.0M	79.0	54.4	68.9%
SETR ([60])	22.1M	76.0	55.3	72.8%
SWIN-T ([43])	28.4M	78.1	47.3	60.6%
SegFormer-B0 ([11])	3.4M	76.2	48.8	64.0%
SegFormer-B1 ([11])	13.1M	78.4	52.7	67.2%
SegFormer-B2 ([11])	24.2M	81.0	59.6	73.6%
SegFormer-B5 ([11])	81.4M	82.4	65.8	79.9%
FAN-B-Hybrid [15]	50.4M	82.2	66.9	81.5%
STL (FAN-B-Hybrid)	50.9M	**82.5**	**68.6**	**83.2%**
FAN-L-Hybrid [15]	76.8M	82.3	68.7	83.5%
STL (FAN-L-Hybrid)	77.3M	**82.8**	**69.2**	**83.6%**

Table 4. **Results on semantic segmentation.** We use mIoU as the evaluation metric. 'R-' and 'X-' refer to ResNet and Xception, respectively. Models trained with STL demonstrate an impressive transferability to the downstream task and achieve significantly better mIoU than other models on Cityscapes and Cityscapes-C.

LV-ViTs and original FAN models generalize well to out-of-distribution data, while other Transformer-based models and the SOTA CNN-based ConvNext models perform weaker. Despite the impressive performance of original FAN models, models trained with STL demonstrate an even better generalization ability and outperform all other models. The performance gains on IN-A and IN-R are more significant than IN-C and set a new state-of-the-art, indicating that the accurate self-emerging token labels are crucial to robustness against out-of-distribution data.

4.5. Transferability to Semantic Segmentation

[61] shows pre-trained models using different training recipes perform differently in downstream tasks. [3] validates that token labeling with CNN token-labelers benefits semantic segmentation and improves the clean mIoU. We also evaluate the transferability of STL to semantic segmentation. As shown in Table 4, pre-trained models with STL reveal better transferability than original FAN counterparts and other prestigious backbones. Remarkably, our approach achieves superior results on both clean and corrupted datasets. As far as we know, this is the first work to reveal that applying dense supervision in backbone pre-training improves not only the clean performance but also the robustness of the downstream task.

Model	Encoder Size	COCO (mAP)
Cascade Mask-RCNN 3× schedule		
ResNet-50 [45]	25M	46.3
ResNeXt-101-32 [62]	-	48.1
ResNeXt-101-64 [62]	-	48.3
Swin-T [43]	28M	50.4
ConvNeXt-T [49]	29M	50.4
FAN-S-Hybrid [15]	26M	53.3
STL (FAN-S-Hybrid)	26M	**53.4**
Swin-S [43]	50M	51.9
ConvNeXt-S [49]	50M	51.9
FAN-B-Hybrid [15]	50M	53.5
STL (FAN-B-Hybrid)	50M	**53.9**
Swin-B [43]	88M	51.9
ConvNeXt-B [49]	89M	52.7
FAN-L-Hybrid [15]	77M	**54.1**
STL (FAN-L-Hybrid)	77M	**54.1**

Table 5. **Results on object detection.** Models trained with STL outperform most CNN-based and transformer-based backbones. Even compared with original FAN models, our models achieve at least on par or even better mAP on COCO, indicating our method can also benefit object detection.

4.6. Transferability to Object Detection

We also validate STL's transferability to the object detection task on COCO and present the results in Table 5. Models trained with STL outperform most CNN-based and Transformer-based backbones. Even compared with original FAN models, STL brings a notable performance gain for FAN-B while achieving comparable mean average precision on FAN-L and FAN-S. We notice the overall improvement in object detection is not as good as in image classification and semantic segmentation, possibly because the semantic segmentation task is more similar to token labeling from the perspective of dense prediction, while object detection involves different techniques such as regression [63, 64] and multi-stage refinement [44, 63].

5. Ablation Study

5.1. Impacts of Different Data Augmentation

We apply spatial-only data augmentation on FAN-TL to obtain accurate token labels that provide clean and correct information toward student models, which is vital to model robustness. The strategy is a "clean teacher noisy student" design, where the student still uses all the augmentations. Our motivation behind this design is to let the teacher and student spatially aligned, with the teacher being as clean as possible to generate high quality token labels (Strong augmentations make it harder to generate good token labels as shown in Fig. 4). We compare the impacts of different data augmentation imposed on FAN-TL in Table 6. Experiments are conducted on the FAN-S-Hybrid model. The student

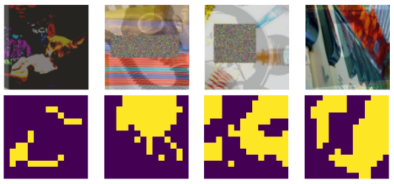

Figure 4. **Visualization results of token labels generated by FAN-TL with full data augmentations.** Strong augmentations significantly affect the quality of token labels.

Spatial	RandAug	CutOut	MixUp	CutMix	IN-1K	IN-C
✓	✗	✗	✗	✗	83.4	**65.5**
✓	✓	✓	✗	✗	82.6	63.5
✓	✓	✓	✓	✓	**83.6**	64.9

Table 6. **Ablation on impact to the student model using various data augmentations on FAN-TL.** The different data augmentations are only imposed on the token-labeler's inputs, while the student model's inputs are fully augmented. Strong data augmentations on FAN-TL undermine token label accuracy and significantly affect the student model's robust accuracy.

model trained with token labels generated by spatial-only data augmentation achieves the best robustness while applying stronger augmentations harms the robust accuracy. Interestingly, applying the consistent full data augmentation on FAN-TL and the student model yields better clean accuracy, which reveals that different combinations of data augmentation may play different roles in improving model robustness and clean performance.

5.2. Impacts of Gumbel-Softmax

We propose to use Gumbel-Softmax as a lightweight solution to retain correct token labels for foreground tokens and eliminate incorrect ones, which yields a more accurate segmentation of the object. We evaluate the impact of Gumbel-Softmax using the FAN-S-Hybrid model in Table 7. As can be seen, models trained with token labels generated by Softmax and Gumbel-Softmax achieve comparable clean accuracy while Gumbel-Softmax improves robust accuracy. This further validates that accurate foreground token labels are critical to model robustness.

Token Labeling	IN-1K	IN-C	mCE (↓)	IN-A	IN-R
Softmax	83.4	65.2	47.6	37.2	51.7
Gumbel-Softmax	83.4	**65.5**	**47.3**	**38.2**	**51.8**

Table 7. **Comparison of Softmax and Gumbel-Softmax.** Both methods achieve comparable clean accuracy while token labels generated by Gumbel-Softmax yield better robustness.

5.3. Training with Heterogeneous Token-Labelers

In the previous experiments, we train student models with isomorphic FAN token-labelers (e.g., FAN-TL-S-Hybrid for FAN-S-Hybrid). We are interested in the impacts of training with a heterogeneous token-labeler and conduct the ablation in Table 8. For each student model, we train with three different FAN token-labelers of model sizes from small to large. Models trained with heterogeneous FAN-TL can achieve at least comparable or even superior performance to ones trained with isomorphic token-labelers. Our large model trained with FAN-TL-B-Hybrid further boosts the clean and robust accuracy to 84.8% and 69.2% with the mCE of 42.1%. This indicates STL is robust to different token-labelers. Such robustness enables us to train a larger student model with a smaller token-labeler, which can reduce the training cost.

Model	Token-Labeler	IN-1K	IN-C	mCE (↓)
FAN-S-Hybrid	-	83.5	64.7	47.8
FAN-S-Hybrid	FAN-TL-S-Hybrid	83.4	65.5	47.3
FAN-S-Hybrid	FAN-TL-B-Hybrid	83.3	**65.8**	**46.8**
FAN-S-Hybrid	FAN-TL-L-Hybrid	**83.5**	65.5	47.4
FAN-B-Hybrid	-	83.9	66.4	45.2
FAN-B-Hybrid	FAN-TL-S-Hybrid	84.4	**68.5**	**43.2**
FAN-B-Hybrid	FAN-TL-B-Hybrid	**84.5**	68.2	43.6
FAN-B-Hybrid	FAN-TL-L-Hybrid	84.3	68.1	43.5
FAN-L-Hybrid	-	84.3	68.3	43.0
FAN-L-Hybrid	FAN-TL-S-Hybrid	84.7	69.0	42.4
FAN-L-Hybrid	FAN-TL-B-Hybrid	**84.8**	**69.2**	**42.1**
FAN-L-Hybrid	FAN-TL-L-Hybrid	84.7	68.8	42.5

Table 8. **Performance comparison of training with different token labelers.** The robustness of student models can be further improved by training with a heterogeneous token labeler.

Model	Token-Labeler	IN-1K	IN-C	mCE (↓)
FAN-S-Hybrid	-	**83.5**	64.7	47.8
FAN-S-Hybrid	NFNet-F6 (CNN)	83.2	65.8	46.9
FAN-S-Hybrid	FAN-TL-B-Hybrid	83.3	**65.8**	**46.8**
FAN-B-Hybrid	-	83.9	66.4	45.2
FAN-B-Hybrid	NFNet-F6 (CNN)	83.5	67.4	44.9
FAN-B-Hybrid	FAN-TL-S-Hybrid	**84.4**	**68.5**	**43.2**
FAN-L-Hybrid	-	84.3	68.3	43.0
FAN-L-Hybrid	NFNet-F6 (CNN)	83.9	68.3	43.2
FAN-L-Hybrid	FAN-TL-B-Hybrid	**84.8**	**69.2**	**42.1**

Table 9. **Comparison to the prior SOTA token labeling method.** Our proposed method that employs self-emerging token labels always yields better robustness than NFNet-F6, which validates the benefit of encoding local information of image patches via self-attention.

5.4. Comparison to Prior Token Labeling Method

As shown in Sec. 4.3 and Sec. 4.4, models trained with FAN-TL demonstrate superior performance and robustness than LV-ViTs [3] trained with a CNN token-labeler (i.e., NFNet-F6 [5]), which may attribute to the self-emerging token labeling and the channel attention design of FAN models. To better understand the effect of token labeling, we

train the same FAN models with different methods. Results are summarized in Table 9. It can be seen that STL still significantly outperforms the CNN token-labeler even with the same student model. Note that NFNet-F6 has more than 400M parameters and achieves an 86.3% Top-1 accuracy on ImageNet-1K while the largest FAN-TL (i.e., FAN-TL-L-Hybrid) only has 77.3M parameters and 84.3% Top-1 accuracy. However, FAN-TL consistently yields better results than NFNet-F6. The rationale behind this is possibly because self-emerging token labels provide self-consistent information to student models.

5.5. Impacts of Loss Weight

Choosing good loss weights is important when multiple losses are jointly optimized. To study the impact of loss weights and verify STL's robustness against different loss weights when training student models, we vary β with various values and present the results in Table 10. We find STL not sensitive to β. Both clean and robust accuracy fluctuates in a small range. We thus set $\beta = 1$ to balance the loss with equal weights for simplicity. Similarly, for the training of token-labelers, α is also set to 1.

β	IN-1K	IN-C	mCE (↓)	IN-A	IN-R
0.5	83.5	65.5	47.3	38.5	51.7
1.0	83.4	65.5	47.3	38.2	51.8
2.0	83.5	65.6	47.2	37.3	51.1

Table 10. **Ablation study of loss weight β.**

6. Conclusion

In this paper, we propose a self-emerging token labeling (STL) framework built on Transformer-based models instead of CNNs. STL enables FAN token-labelers to self-produce accurate and semantically meaningful token labels for training student models with dense supervision. Through extensive experiments and ablation studies, we demonstrate that models trained with STL significantly surpass the original FAN counterparts trained only with image-level labels and achieve remarkable robustness improvement in various visual recognition tasks. Our study validates that the self-produced knowledge from ViTs can indeed benefit their pre-training. We hope this work sheds light on the potential and understanding of self-emerging token labeling from ViTs and motivates future research.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2020. 1, 2

[2] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357. PMLR, 2021. 1, 2, 4, 6

[3] Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. *NeurIPS*, 2021. 1, 2, 5, 6, 7, 8

[4] Sangdoo Yun, Seong Joon Oh, Byeongho Heo, Dongyoon Han, Junsuk Choe, and Sanghyuk Chun. Re-labeling imagenet: from single to multi-labels, from global to localized labels. *arXiv:2101.05022*, 2021. 1

[5] Andrew Brock, Soham De, Samuel L Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. *arXiv:2102.06171*, 2021. 1, 8

[6] Wenjie Luo, Yujia Li, Raquel Urtasun, and Richard Zemel. Understanding the effective receptive field in deep convolutional neural networks. *NeurIPS*, 29, 2016. 2

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *NeurIPS*, 30, 2017. 2

[8] Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. In *NeurIPS*, 2021. 2

[9] Yutong Bai, Jieru Mei, Alan L Yuille, and Cihang Xie. Are transformers more robust than cnns? In *NeurIPS*, 2021. 2

[10] Sayak Paul and Pin-Yu Chen. Vision transformers are robust learners. In *AAAI*, 2022. 2

[11] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, 2021. 2, 6, 7

[12] Yao Qin, Chiyuan Zhang, Ting Chen, Balaji Lakshminarayanan, Alex Beutel, and Xuezhi Wang. Understanding and improving robustness of vision transformers through patch-based negative augmentation. In *NeurIPS*, 2022. 2

[13] Charles Herrmann, Kyle Sargent, Lu Jiang, Ramin Zabih, Huiwen Chang, Ce Liu, Dilip Krishnan, and Deqing Sun. Pyramid adversarial training improves vit performance. In *CVPR*, pages 13419–13429, 2022. 2

[14] Xiaofeng Mao, Yuefeng Chen, Ranjie Duan, Yao Zhu, Gege Qi, Shaokai Ye, Xiaodan Li, Rong Zhang, and Hui Xue. Enhance the visual representation via discrete adversarial training. *arXiv preprint arXiv:2209.07735*, 2022. 2, 6

[15] Daquan Zhou, Zhiding Yu, Enze Xie, Chaowei Xiao, Animashree Anandkumar, Jiashi Feng, and Jose M Alvarez. Understanding the robustness in vision transformers. In *ICML*, 2022. 2, 6, 7

[16] Li Yuan, Francis EH Tay, Guilin Li, Tao Wang, and Jiashi Feng. Revisiting knowledge distillation via label smoothing regularization. In *CVPR*, pages 3903–3911, 2020. 2

[17] Sangdoo Yun, Seong Joon Oh, Byeongho Heo, Dongyoon Han, Junsuk Choe, and Sanghyuk Chun. Re-labeling imagenet: from single to multi-labels, from global to localized labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2340–2350, 2021. 2

[18] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021. 2

[19] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 3

[20] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9650–9660, 2021. 3

[21] Jiarui Xu, Shalini De Mello, Sifei Liu, Wonmin Byeon, Thomas Breuel, Jan Kautz, and Xiaolong Wang. Groupvit: Semantic segmentation emerges from text supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18134–18144, 2022. 3

[22] Joachim M Buhmann, Jitendra Malik, and Pietro Perona. Image recognition: Visual grouping, recognition, and learning. *Proceedings of the National Academy of Sciences*, 96(25):14203–14204, 1999. 3

[23] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *ICCV*, 2021. 4

[24] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPR Workshops*, pages 702–703, 2020. 4

[25] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *AAAI*, 2020. 4

[26] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *ICLR*, 2017. 4

[27] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, pages 6023–6032, 2019. 4

[28] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. In *ICLR*, 2017. 5

[29] Geoffrey J McLachlan. Iterative reclassification procedure for constructing an asymptotically optimal rule of allocation in discriminant analysis. *Journal of the American Statistical Association*, 70(350):365–369, 1975. 5

[30] Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. Semi-supervised self-training of object detection models. 2005. 5

[31] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, pages 10687–10698, 2020. 5

[32] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020. 5

[33] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *NIPS*, 2004. 5

[34] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255. Ieee, 2009. 5

[35] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *ICLR*, 2019. 5

[36] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *CVPR*, pages 15262–15271, 2021. 5

[37] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *ICCV*, 2021. 5

[38] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5

[39] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Doll'a r, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, 2014. 5

[40] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv:1912.01703*, 2019. 5

[41] Ross Wightman. Pytorch image models. `https://github.com/rwightman/pytorch-image-models`, 2019. 5

[42] MMSegmentation Contributors. Mmsegmentation: Openmmlab semantic segmentation toolbox and benchmark, 2020. 5

[43] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 10012–10022, 2021. 6, 7

[44] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 6, 7

[45] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6, 7

[46] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, pages 4510–4520, 2018. 6

[47] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, pages 6105–6114. PMLR, 2019. 6

[48] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *CVPR*, pages 568–578, 2021. 6

[49] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *CVPR*, 2022. 6, 7

[50] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *CVPR*, 2021. 6

[51] Xiaofeng Mao, Gege Qi, Yuefeng Chen, Xiaodan Li, Ranjie Duan, Shaokai Ye, Yuan He, and Hui Xue. Towards robust vision transformer. In *CVPR*, 2021. 6

[52] Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. In *NeurIPS*, 2021. 6

[53] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 6

[54] Christoph Kamann and Carsten Rother. Benchmarking the robustness of semantic segmentation models. In *CVPR*, pages 8828–8838, 2020. 7

[55] Hengshuang Zhao, Xiaojuan Qi, Xiaoyong Shen, Jianping Shi, and Jiaya Jia. Icnet for real-time semantic segmentation on high-resolution images. In *ECCV*, pages 405–420, 2018. 7

[56] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 7

[57] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *ICLR*, 2016. 7

[58] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 90:119–133, 2019. 7

[59] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *CVPR*, pages 2881–2890, 2017. 7

[60] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *ICCV*, pages 11936–11945, 2021. 7

[61] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *CVPR*, pages 558–567, 2019. 7

[62] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 1492–1500, 2017. 7

[63] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 7

[64] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *NeurIPS*, 28:91–99, 2015. 7