Train ViT on Small Dataset With Translation Perceptibility

Huan Chen Student[1]
chenhuan21s@ict.ac.cn
Wentao Wei Student[2]
weiwentao@seu.edu.cn
Ping Yao Prof[1]
yaoping@ict.ac.cn

[1] Institute of Computing Technology,
Chinese Academy of Sciences
Beijing, China

[2] Chien-Shiung Wu College,
Southeast University
Nanjing, China

Abstract

The Vision Transformer (ViT) has emerged as a prominent model in the domain of computer vision in recent years, effectively supplanting traditional Convolutional Neural Network (CNN) models. However, due to the absence of certain properties intrinsic to CNNs, such as parameter sharing and translation invariance, ViTs tend to necessitate larger quantities of training data. To circumvent this limitation, a multitude of methods have been proposed by the academic community to optimize ViT's performance when dealing with smaller datasets. In this paper, we introduce a novel self-supervised auxiliary task designed to guide ViTs in acquiring the capability of translation perceptibility. This strategy facilitates the models in obtaining inductive bias more efficiently from smaller datasets, obviating the need for pre-training on larger datasets or modifications to the network architecture. The efficacy of our approach is corroborated through its application to multiple small datasets, demonstrating impressive scale perceptibility. Furthermore, when utilized in tandem with current state-of-the-art methods, our approach yields significant performance enhancements.

1 Introduction

ViT (Vision Transformer) [9, 20, 32, 34] is a rapidly developing vision model in recent years, gradually replacing traditional CNN models [14, 17, 23, 38, 40, 43]. The foundation of ViT is the Transformer model in the NLP domain [8, 42]. In vision tasks, ViT [9] adopts patch as input, dividing the image into several non-overlapping patches, which enables ViT to have a global view. A notable feature of ViT is that it tends to require a larger amount of data than CNN models to extract relevant characteristics, known as "data hungry." [13] The possible reason is that ViT lacks some ideal properties inherent in CNN architecture, which particularly suitable for solving visual tasks, such as localization and translation invariance. In order to alleviate the above problems, it is beneficial for ViTs to learn some attributes of CNN, such as translation invariance [19, 45]. However, experiments and structural analysis of ViTs have shown that it is extremely difficult to modify the structure directly to bestow ViT the translation invariance property, unless the basic unit of ViTs is changed to Conv, which consequently leads to the diminution of ViTs' global perspective. Meanwhile, the basic unit of self-attention is composed of fully connected layers, and we believe that self-attention may be able to obtain properties similar to translation invariance. Therefore, our objective is to enhance the effectiveness of ViTs in learning translation invariance without altering the original network structure, by incorporating an additional auxiliary task, particularly focusing on the acquisition of translation perceptibility.

Our contributions are as follows:

1. We propose a simple self-supervised auxiliary task to guide ViTs in learning translation perceptibility without changing the model structure, which helps the model learn inductive bias more efficiently from small datasets, aiding ViTs in achieving better performance.

2. We conclude from extensive experiments that our method performs well on small datasets across various resolutions, and is especially effective with larger input resolutions. For instance, our method can enhance performance by over 5% compared to the baseline at lower resolutions, and achieve a 10% improvement at higher resolutions.

3. Our proposed method exhibits broad extensibility and can be easily integrated into existing state-of-the-art methods as a plug-and-play component, achieving an average performance improvement of 2%.

2 Related Work

In this section, we will introduce recent works on ViTs for small datasets and self-supervised learning. With the rapid advancements of Transformer models in the NLP domain, researchers have attempted to introduce them into the computer vision field. The original Vision Transformer [9] model was the first to apply self-attention mechanisms to computer vision, directly dividing images into non-overlapping patches as input for the self-attention mechanism. With the support of abundant training data [35, 37] and powerful data augmentation strategies [5, 16, 39, 50, 51], ViTs have successfully entered the computer vision field. Researchers have tried various approaches to improve Vision Transformers and attempted to train them from scratch [12, 28, 28, 30, 41, 46, 49]. Nonetheless, most of these studies are conducted on large datasets [35, 37], which typically entail significant training expenses. The performance of ViTs trained from scratch on small datasets remains limited, mainly because ViTs rely on large amounts of training data to learn visual features, and are prone to underfitting when data is scarce. To address these issues, researchers have proposed various methods to optimize ViTs' performance on small datasets.

Vision Transformers for Small Datasets: SL-ViT [26] combines Shifted Patch Tokenization (SPT) and Local Self-Attention (LSA) mechanisms and applies a range of data augmentation techniques [5, 16, 50, 51], enabling ViTs to effectively learn from scratch on small datasets. By introducing an auxiliary self-supervised task, Drloc [29] learns the spatial location information between image tokens, helping the model more effectively capture locality. With an efficient self-supervised weight learning strategy, vfsd [10] allows the model to quickly converge on small datasets, which consists of two stages: pre-training and fine-tuning. During pre-training, distillation guides student and teacher models to learn low-resolution local and global information, allowing the model to rapidly converge during the fine-tuning stage. Unlike the aforementioned methods, our approach aims to guide the model in learning translation perceptibility, allowing ViTs to obtain local bias information similar to CNNs without changing the network structure.

Self-supervised learning: In recent years, self-supervised learning has been extensively

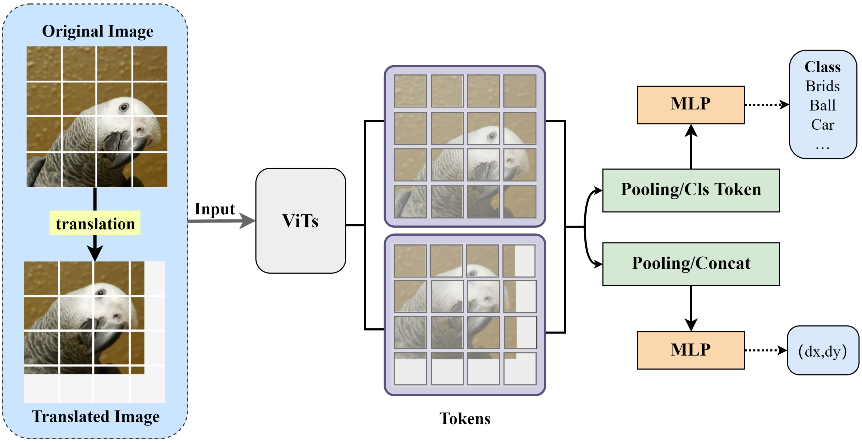


Figure 1: Illustration of the pipeline. In order to guide the model in learning translation perceptibility, we first apply an arbitrary translation to the input image along any direction and generate the corresponding translation labels. Subsequently, both the original and translated images are fed into the network for processing. The output tokens are utilized for classification tasks as well as translation perception prediction tasks.

applied to the pre-training of ViTs [2, 4, 15, 27, 47, 52]. Compared to supervised ViTs, self-supervised ViTs exhibit more explicit information about image features [4]. MAE [15] randomly masks input patches and feeds the remaining patches into the encoder, guiding the model to reconstruct the masked patches, thus validating the effectiveness of the MIM paradigm. SimMIM [47] employs the simplest MIM method: randomly masking a portion of the input image patches and predicting the original pixel values of the masked patches through an encoder-decoder. iBOT [52] distills the masked patch tokens and uses the teacher network as an online tokenizer to obtain visual semantic information. BEiT [3] proposes to randomly mask a certain percentage of image patches and predict the corresponding visual tokens of the masked patches. In BEiTv2 [33], a semantically-aware image tokenizer is used, and a bottleneck structure is designed to pre-train the CLS token. Drloc and vfsd [10, 29] introduce self-supervised learning on small datasets, enabling ViTs to better capture spatial locality on small datasets. Our proposed approach differs from others in that it introduces translation perceptibility through simple self-supervised learning. By constructing random translations and performing translation regression prediction on the output tokens, we guide the model to learn translation perceptibility, which is similar to the translation invariance of CNNs. Moreover, our approach is a plug-and-play self-supervised task, which can be combined with current state-of-the-art methods to achieve even better performance.

3 Method

3.1 Translation Equivariance in CNNs

In this section, we discuss the relationship between translation equivariance and translation invariance in Convolutional Neural Networks (CNNs). Let $x \in R^{b*c*h*w}$ denote the input, $y \in R^{b*n}$ the output of the neural network, and F the network function. The inference process of the network can be represented as $y = F(x)$. Translation invariance of an image can be expressed as shown in Eq. 1.

$$y = F(trans(x)) = F(x) \tag{1}$$

where trans denotes random translation of the input.

CNNs exhibit a certain degree of translation invariance. Let D represent the downsample operation (e.g., pooling), W the convolutional weights, and $W * x$ the convolution operation applied to the input. For CNNs, translation invariance can be expressed as shown in Eq. 2.

$$y_{cnn} = D(W * x) = D(W * trans(x)) \tag{2}$$

In reality, convolution only possesses translation equivariance, which means that the translation of the target in the input is reflected in the output feature map after convolution (as shown in Eq. 3). Only when combined with downsampling operations does a CNN exhibit translation invariance.

$$trans(W * x) = W * trans(x) \tag{3}$$

3.2 Guiding ViT to Achieve Translation Perceptibility

As discussed in the preceding section, convolution exhibits translation equivariance properties. In terms of implementation details, convolution is commonly executed at a lower level by unfolding the parameter weights into a matrix and conducting matrix computations (e.g., im2col+gemm [18], Winograd [24, 44]), which is quite similar to the computation in linear layers (which are prevalent in ViTs). Inspired by the acceleration of convolutional computation [18], we believe that, theoretically, a fully connected layer can perform similarly to a convolutional layer and possess most of its characteristics. In other words, ViTs can exhibit a similar translation equivariance property, which we refer to as translation perceptibility. Subsequently, we introduce the training process for translation perceptibility (see Fig. 1). Given an original input x, we denote its randomly translated version as $trans(x)$. By feeding both the original and translated images into the ViT network, we obtain the outputs y and y_t, respectively.

$$y = ViT(x) \tag{4}$$

$$y_t = ViT(trans(x)) \tag{5}$$

We hope that the translation operation can be easily perceived in the output $diff(y, y_t)$ and aim to guide the model to preserve local information in the input while performing self-attention. As a result, we propose a translation perceptibility module that utilizes an MLP as the header for perceiving translation, as shown in Eq. 6,

$$(dx, dy) = MLP(Concat(y, y_t)) \tag{6}$$

where (dx, dy) represent the predicted relative offsets in the x and y coordinates. We denote (lx, ly) as the offset labels generated during random translation, which can be randomly generated as follows:

$$lx = norm_{0,1}(randn\_ratio\_x \times img\_width),\ randn\_ratio\_x \in (-\alpha_1,\ \alpha_1) \tag{7}$$

$$ly = norm_{0,1}(randn\_ratio\_y \times img\_height),\ randn\_ratio\_y \in (-\alpha_2,\ \alpha_2) \tag{8}$$

By incorporating the translation perceptibility module into the ViT model, we enable the model to maintain local information in the input while performing self-attention. This allows

Attributes	Depth	Patch-size	Token Dimension	Heads	MLP-ratio	Window-size
ViT	9	[4,8,16]	192	12	2	-
Swin	[2,4,6]	[2,4,8]	96	[3,6,12]	2	4,7
CaiT	[2,4]	[4,8]	192	4	2	-

Table 1: Details of ViT architectures in our approach

Dataset	Train Size	Test Size	Resolution	Classes
CIFAR10 [22]	50,000	10,000	32×32	10
CIFAR100 [22]	50,000	10,000	32×32	100
CINIC10 [7]	90,000	90,000	32×32	10
SVHN [22]	73,257	26,032	64×64	10
Imagenet-100	50,000	5,000	64×64	100
Imagenet-200 [25]	100,000	10,000	64×64	200
Imagenet-1k [36]	1000,000	150,000	64×64	1000
flowers102 [31]	6,507	1,682	224×224	102
WHU-RS19 [6]	797	208	224×224	19
UCMerced_LandUse [48]	1,680	420	256×256	21

Table 2: Details of datasets in terms of sample size and class num used in our experiments.

the ViT model to exhibit translation equivariance, similar to CNNs, without changing the network structure. It is worth noting that, the translation perceptibility module is a plug-and-play self-supervised task that can be combined with state-of-the-art methods to achieve better performance in vision tasks. We present the loss function as shown in Eq. 9:

$$L_{trans} = \frac{1}{2n} \sum_{1...n}^{i} [|lx_i - dx_i| + |ly_i - dy_i|] \tag{9}$$

where n denotes the batch size. The final loss is $L_{tot} = L_{cls} + \lambda L_{trans}$.We set $\lambda = 0.5$ in all the experiments with ViT, and $\lambda = 0.6$ in case of Swin.

4 Experiment

In this section, we discuss the experimental settings, including datasets and training details, qualitative analysis (Sec. 4.1), and ablative analysis (Sec. 4.2).

Vision Transformer Encoders: We validate our proposed approach on ViTs (ViT, Swin [9, 30]). Our baseline configuration is inherited from vfsd [10], with the distinction that we increase the patch size and window size for larger inputs. Specifically, when the input size is 224, we increase the patch size for ViT to 16 and Swin to 8 (or 16), and set the window size for Swin to 7 (Table 1).

Datasets: We validate our approach on small-scale datasets (CIFAR10/100 [22], SVHN [11], CINIC10 [7], Imagenet-100/200/1K [25, 36]), two small remote sensing datasets (WHU-RS19 [6], UC Merced LandUse [48]) and a fine-grained dataset (flowers102 [31]). Details about the dataset size, sample resolution and the number of classes are provided in Table 2.

Training Set: We follow the supervised learning training framework presented in vfsd [10], applying standard data augmentations for consistency [5, 16, 50, 51]. All models are trained on a single Nvidia RTX6000 24GB GPU (Batch Size = 64 for 224×224, 256 for 32×32 and 64×64). The Adam optimizer [21] is employed with a learning rate of 0.001 and a learning decay rate of 5e-2 using cosine scheduling.

Model	Imagenet-200	CIFAR10	CIFAR100	CINIC10	SVHN
ResNet18 [14]	53.32	90.44	64.49	77.79	96.78
ResNet56 [14]	56.51	94.65	74.44	85.34	97.61
ResNet101 [14]	59.77	95.27	76.18	86.81	97.82
EfficientNet B0 [40]	55.48	88.38	61.64	75.64	96.06
ViT(scrach)	54.07	93.58	73.81	83.73	97.82
SL-ViT [26]	58.75	94.53	76.92	84.48	97.79
ViT-Drloc [29]	54.44	81.00	58.29	71.50	94.02
ViT-vfsd [10](reproduce)	58.56	96.06	76.41	86.90	98.02
ViT-Trans(ours)	59.47	96.26	77.16	86.45	98.09
ViT-vfsd-Trans(ours)	**59.48**	**96.74**	**78.01**	**87.64**	**98.20**
Swin(scrach)	60.05	93.97	77.32	83.75	97.83
SL-Swin [26]	64.95	94.93	79.99	87.22	97.92
Swin-Drloc [29]	48.66	86.07	65.32	77.25	95.77
Swin-vfsd [10](reproduce)	64.28	96.52	80.67	87.96	98.02
Swin-Trans(ours)	62.27	96.87	80.28	88.26	98.15
Swin-vfsd-Trans(ours)	**65.05**	**97.08**	**81.25**	**88.63**	**98.17**

Table 3: Our approach, without modifying the model architecture, demonstrates favorable performance compared to various ViT baselines [35, 39] and CNNs. Moreover, by fine-tuning our proposed approach on the basis of vfsd [10], we achieve even better results.

4.1 Results

Generalization: We adopt two ViT architectures (as detailed in Table 1) and maintain the same patch size configuration as in previous research (vfsd [10]). Specifically, on Tiny-Imagenet-100/200 [25], we set the patch size to 8 (ViT) and 4 (Swin) to generate the corresponding input tokens for ViT and Swin. For other datasets, we reduce the patch size to 4 (ViT) and 2 (Swin). We observe that, particularly on Imagenet-100/200 [25], our proposed approach (ViT/Swin-xx-Trans) achieves significant performance improvements compared to other advanced techniques (as shown in Table 3, 4), and effectively leverages other methods to attain enhanced performance. On other challenging datasets, our proposed approach also performs well and outperforms CNN-based models (see Table 3). This observation highlights not only the effective characteristic-guiding ability of our approach but also its strong adaptability. Particularly, the experiments conducted across various approaches (SL-ViT, Drloc, vfsd [10, 26, 29]) consistently support this notion, emphasizing the versatility of our approach.

Performance on Fine-Grained Datasets: We conduct a comparison on flowers102 [31] using multiple ViT architectures (Table 1) against the current state-of-the-art methods. All groups have their epoch settings set to 100, with other parameters kept at their default values. For vfsd [10], we perform a pre-training of 200 epochs before fine-tuning for another 100 epochs. The experimental results (Table 5) show that our approach (ViT-Trans and Swin-Trans) exhibits a significant performance advantage in fine-grained image classification tasks compared to vfsd-based methods (ViT-vfsd and Swin-vfsd). Particularly in the Swin-Trans model, the accuracy reaches the highest value at 85.37%. These results suggest that our proposed approach may outperform other methods in such tasks, demonstrating its effectiveness in fine-grained image classification tasks when compared to other methods.

Performance on Remote Sensing Scene Classification Datasets: Remote sensing scene

Model	Imagenet-200	Imagenet-100	Model	Imagenet-200	Imagenet-100
ViT(scrach)	54.07	62.56	Swin(scrach)	60.05	66.36
ViT-Trans(ours)	**59.47**	**65.50**	Swin-Trans(ours)	**62.27**	**69.00**
SL-ViT [26]	58.75	66.96	SL-Swin [26]	64.95	71.88
SL-ViT-Trans(ours)	**61.49**	**69.64**	SL-Swin-Trans(ours)	**66.80**	**74.81**
ViT-Drloc [29]	54.44	64.52	Swin-Drloc [29]	-	67.08
ViT-Drloc-Trans(ours)	**57.30**	**65.36**	Swin-Drloc-Trans(ours)	-	**69.96**
ViT-vfsd [10]	58.56	65.38	Swin-vfsd [10]	64.28	69.38
ViT-vfsd-Trans(ours)	**59.48**	**65.66**	Swin-vfsd-Trans(ours)	**65.05**	**71.30**

Table 4: Our training approach, integrated across methods, boosts performance: ViT improves by an average of 2.21% (max 5.40%), and Swin by 2.17% (max 2.93%). This consistency suggests our approach effectively teaches models translation-invariant traits via a simple unsupervised task, universally enhancing model performance.

classification datasets typically have limited data and multiple scene categories. We have validated our proposed approach on UCMerced_LandUse [48] and WHU-RS19 [6], and the experimental results show that our proposed approach significantly improves upon the baseline, maintaining a clear advantage even when compared to vfsd [10] (Table 5).

Attention to Salient Regions: We visualize the attention using attention rollout [1] on test samples from Imagenet-200 (Fig. 2). We observe that the output attention maps maintain relative translation when images before and after translation are used as input. This demonstrates that our method successfully guides the model to learn translation perceptibility.

Model	WHU-RS19	UCMerced_LandUse	flowers102
ViT(scrach)	82.69	83.57	68.67
ViT-vfsd [10]	89.76	91.66	69.01
ViT-Trans(ours)	91.83	94.52	73.65
ViT-vfsd-Trans(ours)	**93.27**	**95.24**	**74.72**
Swin(scrach)	85.10	88.81	79.13
Swin-vfsd [10]	87.02	94.76	80.62
Swin-Trans(ours)	**94.71**	**97.62**	**85.37**
Swin-vfsd-Trans(ours)	**94.71**	96.43	84.66

Table 5: Our approach has significant advantages in Remote Sensing Scene and fine-grained classification tasks and can be further improved by leveraging current state-of-the-art methods (vfsd [10]).

4.2 Ablative Analysis

Different Input: We test our proposed approach on different input sizes using [31]:32×32, 64×64 and 224×224. As shown in Fig. 3, for smaller input sizes (32×32, Fig. 3(a)), our approach and vfsd exhibit comparable performance. Furthermore, significant performance improvements are achieved by our approach when fine-tuned with vfsd's pretrained weights. As the input size increases to 64×64 (Fig. 3(b)), our approach outperforms vfsd, and it still manages to achieve a slight performance improvement using the pretrained weights obtained from vfsd. With an input size of 224×224 (Fig. 3(c)), our approach significantly outperforms vfsd; however, the benefit gains obtained from the vfsd pretrained weights are reduced. In summary, our approach to guiding translation perceptibility is better suited for larger input sizes, and as the input size increases, the benefits from vfsd weights diminish.

Exploring Scalability: Experiments (Table 4) are carried out on a range of approaches utilizing Imagenet-100/200, leading to the formation of two distinct groups: the first group trains and fine-tunes the models in accordance with their original configurations, while the second group integrates our self-supervised task without any alterations to the training configuration. The experimental findings indicate that each approach demonstrates a consider-

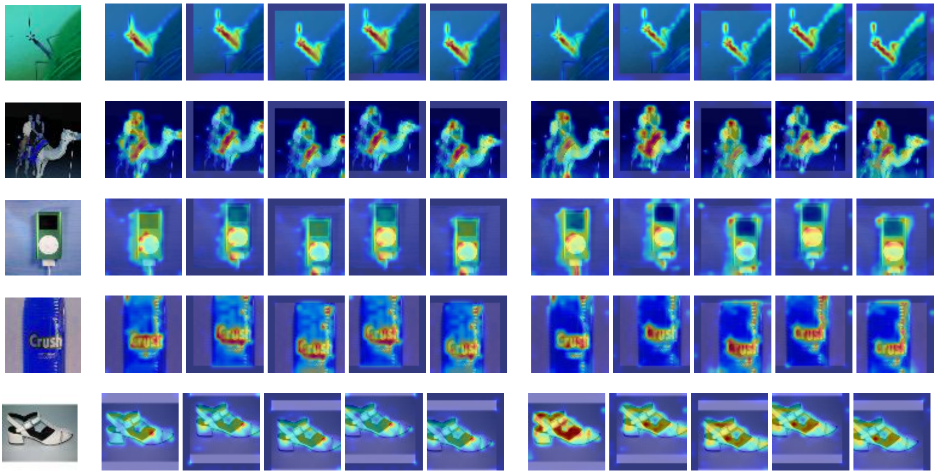


Figure 2: Using attention rollout [1] on low-res Imagenet-100 test samples, we compare image translations from our method (left) and vfsd (right). Each group consists of five images, with the first image having no translation and the subsequent four images being translated in four different directions. Our method demonstrates improved resilience to image translations.

able enhancement in performance when our approach is incorporated, thereby emphasizing the effectiveness and robust generalizability of our proposed approach.

Model λ	ViT-Trans	Swin-Trans
0.1	69.68	84.07
0.2	69.80	84.96
0.3	71.21	85.02
0.4	73.04	85.02
0.5	**73.59**	85.32
0.6	72.52	**85.37**
0.7	72.32	84.78
0.8	71.83	84.72
0.9	71.65	84.08
1.0	71.14	83.32

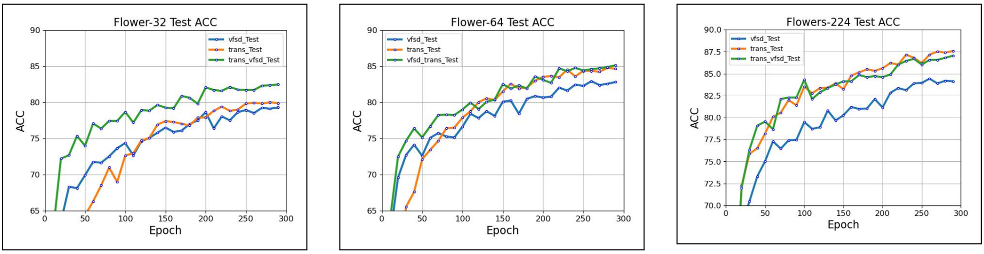
Table 6: Exploring the performance of our approach (on flowers102) under different Loss Function Weight(λ).

Model α	ViT-Trans	Swin-Trans
0.0	68.40	78.72
0.1	73.59	84.90
0.2	**73.65**	85.37
0.3	73.47	**85.55**
0.4	73.35	85.32
0.5	73.17	84.66
0.6	73.06	84.36

Table 7: Exploring the performance of our approach (on flowers102) under different translation magnitude(α).

Exploring Patch Sizes: In the context of an exploratory study conducted on the fine-grained image dataset flowers102 [31], we examine the influence of various patch configurations for Swin. As depicted in Table 8, the experimental outcomes indicate that, given an input image size of 224×224, Swin with a patch size of 8 outperforms its counterpart with a patch size of 16. Importantly, our proposed training approach enhances model performance by effectively guiding the model to learn translation-invariant characteristics. This improvement is consistently observed, irrespective of whether the patch size is 8 or 16.

Exploring Batch Size: In this experiment (Table 9), we employ default settings, with the sole alteration being the batchsize. Our proposed approach exhibits a preference for a batch-



(a) size:32×32 (b) size:64×64 (c) size:224×224

Figure 3: We conduct a detailed analysis of convergence performance across flowers102 with different input sizes((a) 32×32; (b) 64×64; (c) 224×224). Experimental results indicate that our proposed approach exhibits good convergence characteristics across all input sizes. As the input size increases, the performance gains from vfsd pre-training gradually decrease.

Model	Patch Size:8		Patch Size:16	
	Epoch 100	Epoch 300	Epoch 100	Epoch 300
Swin(scrach)	82.27	86.31	74.07	80.68
Swin-vfsd [10]	87.46	90.67	80.62	84.78
Swin-Trans(ours)	**89.77**	**93.64**	**85.37**	**87.81**
Swin-vfsd-Trans(ours)	89.60	92.27	84.66	87.16

Table 8: Explored the influence of different patch sizes for Swin on flowers102.

Model batchsize	Swin-Trans(ours)	Swin-vfsd-Trans
Batchsize=16	85.26	**86.09**
Batchsize=32	**85.84**	85.55
Batchsize=64	85.37	84.66
Batchsize=128	83.23	84.30
Batchsize=256	83.41	83.41

Table 9: Exploring the performance of our approach under different batchsizes on flowers102.

size of 64 when handling larger inputs. Additionally, when fine-tuning based on the vfsd training weights, the optimal performance is achieved with a smaller batchsize (16).

Exploring Translation Perceptibility Loss Function Weight(λ): (Eq.9) In this experiment, we conduct an extensive exploration of the Translation Perceptibility Loss Function Weight (λ) for the ViT and Swin using flowers102. As illustrated in Table 6, the experimental findings indicate that for the ViT model, the optimal performance is attained with $\lambda = 0.5$, while for Swin, the best performance is achieved when $\lambda = 0.6$.

Exploring Translation Magnitude(α): (Eq.7,8) By reseting the translation magnitude (α), we conduct an extensive exploration of α for the ViT and Swin on flowers102. As illustrated in Table 7, we find that α significantly improves model performance from 0.1 to 0.6 (contrast with α = 0.0). The optimal performance is attained with $\alpha = 0.2$ for the ViT model, while for Swin, the best performance is achieved when $\alpha = 0.3$.

Performance Comparison with Self-Supervised Learning Based Techniques: Compared to other self-supervised training methods, our approach demonstrates significant advantages (Table 10).

Efficiency in Terms of Epochs: We observe that our proposed approach, after only 300 training epochs, outperforms drloc (which requires 600 training epochs) and the current state-of-the-art (SOTA) method (which requires 500 training epochs, including 200 for pre-training and 300 for fine-tuning) in model performance (Table 11).

Exploring CaiT: In the experiments related to the CaiT model, we reused the parameter configuration from the Swin model experiments. The results, as shown in Table 12, demonstrate that our method still maintains strong competitiveness on general datasets such as Tiny-Imagenet, CIFAR-10/100.

Performance on Imagenet-1k: We conducted relevant experiments on the Imagenet-1k using the ViT model, as shown in Table 13. The results indicate that our method maintains a

Model	Imagenet-200	CIFAR10	CIFAR100
SimCLR	58.87	85.84	74.77
MOCO-V3	52.39	85.55	72.22
Drloc	54.44	81.00	58.29
vfsd	58.56	96.06	76.65
Trans(ours)	**59.48**	**96.74**	**78.01**

Table 10: A comparative analysis between our proposed self-supervised learning method and existing techniques, with ViT as a common baseline.

Model	Epochs	CIFAR100
ViT-Drloc	600	68.29
ViT-vfsd	200(pretrain) + 100(finetune)	75.37
ViT-vfsd	200(pretrain) + 300(finetune)	76.41
ViT-Trans(ours)	300	**77.17**

Table 11: In the case of sufficient epochs, we compare the performance of our proposed approach with current approaches.

Model	Imagenet-200	CIFAR10	CIFAR100
CaiT(scrach)	58.87	94.91	76.89
CaiT-vfsd [10]	62.18	96.50	79.64
CaiT-Trans(ours)	62.00	96.73	80.66
CaiT-vfsd-Trans(ours)	**62.84**	**97.32**	**80.90**

Table 12: Experiments on small general datasets show our method's competitiveness with the CaiT model.

Model	Imagenet-1k
ViT(scarch)	61.55
ViT-Trans(ours)	**63.96**

Table 13: Our method retains effectiveness on Imagenet-1k.

certain level of effectiveness on larger-scale datasets. However, it should be noted that our epoch is set to 100, and we cannot yet determine the performance under other epoch settings.

5 Conclusion

In this paper, we propose an efficient, plug-and-play self-supervised training method that allows Vision Transformers (ViTs) to learn from scratch on small datasets without the need for large-scale pre-training. Our approach draws inspiration from the acceleration of convolutional computation [18], attempting to guide ViTs in learning translation perceptibility. Extensive experiments indicate that our method achieves competitive results when trained from scratch on small datasets with varying input resolutions and performs favorably compared to existing state-of-the-art methods. Notably, the benefits of our proposed approach become more apparent as the input size increases. Futhermore, we integrate our approach across different methods and generally observe an improvement (up to 5.40%), suggesting that our approach exhibits broad extensibility.

References

[1] Samira Abnar and Willem Zuidema. Quantifying attention flow in transformers, 2020.

[2] Sara Atito, Muhammad Awais, and Josef Kittler. Sit: Self-supervised vision transformer, 2022.

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

[4] 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.

[5] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.

[6] Dengxin Dai and Wen Yang. Satellite image classification via two-layer sparse coding with biased image representation. *IEEE Geoscience and remote sensing letters*, 8(1): 173–176, 2010.

[7] Luke N Darlow, Elliot J Crowley, Antreas Antoniou, and Amos J Storkey. Cinic-10 is not imagenet or cifar-10. *arXiv preprint arXiv:1810.03505*, 2018.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pretraining of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[10] Hanan Gani, Muzammal Naseer, and Mohammad Yaqub. How to train vision transformer on small-scale datasets? *arXiv preprint arXiv:2210.07240*, 2022.

[11] Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, and Vinay Shet. Multi-digit number recognition from street view imagery using deep convolutional neural networks, 2014.

[12] Ben Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference, 2021.

[13] Ali Hassani, Steven Walton, Nikhil Shah, Abulikemu Abuduweili, Jiachen Li, and Humphrey Shi. Escaping the big data paradigm with compact transformers, 2022.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[15] 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 (CVPR)*, pages 16000–16009, June 2022.

[16] Dan Hendrycks, Norman Mu, Ekin D Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *arXiv preprint arXiv:1912.02781*, 2019.

[17] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks, 2018.

[18] Bo Kågström, Per Ling, and Charles Van Loan. Gemm-based level 3 blas: high-performance model implementations and performance evaluation benchmark. *ACM Transactions on Mathematical Software (TOMS)*, 24(3):268–302, 1998.

[19] Osman Semih Kayhan and Jan C van Gemert. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14274–14285, 2020.

[20] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Computing Surveys*, 54(10s):1–41, jan 2022. doi: 10.1145/3505244. URL `https://doi.org/10.1145%2F3505244`.

[21] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[22] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[24] Andrew Lavin and Scott Gray. Fast algorithms for convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4013–4021, 2016.

[25] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015.

[26] Seung Hoon Lee, Seunghyun Lee, and Byung Cheol Song. Vision transformer for small-size datasets, 2021.

[27] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning, 2022.

[28] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers, 2021.

[29] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco De Nadai. Efficient training of visual transformers with small datasets, 2021.

[30] 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, 2021.

[31] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729. IEEE, 2008.

[32] Namuk Park and Songkuk Kim. How do vision transformers work?, 2022.

[33] Zhiliang Peng, Li Dong, Hangbo Bao, Qixiang Ye, and Furu Wei. Beit v2: Masked image modeling with vector-quantized visual tokenizers. *arXiv preprint arXiv:2208.06366*, 2022.

[34] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.

[35] Tal Ridnik, Emanuel Ben-Baruch, Asaf Noy, and Lihi Zelnik-Manor. Imagenet-21k pretraining for the masses. *arXiv preprint arXiv:2104.10972*, 2021.

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115: 211–252, 2015.

[37] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017.

[38] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[39] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.

[40] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020.

[41] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers, 2021.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[43] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, and Bin Xiao. Deep high-resolution representation learning for visual recognition, 2020.

[44] Shmuel Winograd. *Arithmetic complexity of computations*, volume 33. Siam, 1980.

[45] Daniel E Worrall, Stephan J Garbin, Daniyar Turmukhambetov, and Gabriel J Brostow. Harmonic networks: Deep translation and rotation equivariance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5028–5037, 2017.

[46] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers, 2021.

[47] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling, 2022.

[48] Yi Yang and Shawn Newsam. Bag-of-visual-words and spatial extensions for land-use classification. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 270–279, 2010.

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

[50] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.

[51] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.

[52] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer, 2022.