How to Train Vision Transformer on Small-scale Datasets?

Hanan Gani
hanan.ghani@mbzuai.ac.ae
Muzammal Naseer
muzammal.naseer@mbzuai.ac.ae
Mohammad Yaqub
mohammad.yaqub@mbzuai.ac.ae

Mohamed Bin Zayed University of Artificial Intelligence,
Abu Dhabi, UAE

Abstract

Vision Transformer (ViT), a radically different architecture than convolutional neural networks offers multiple advantages including design simplicity, robustness and state-of-the-art performance on many vision tasks. However, in contrast to convolutional neural networks, Vision Transformer lacks inherent inductive biases. Therefore, successful training of such models is mainly attributed to pre-training on large-scale datasets such as ImageNet with 1.2M or JFT with 300M images. This hinders the direct adaption of Vision Transformer for small-scale datasets. In this work, we show that self-supervised inductive biases can be learned directly from small-scale datasets and serve as an effective weight initialization scheme for fine-tuning. This allows to train these models without large-scale pre-training, changes to model architecture or loss functions. We present thorough experiments to successfully train monolithic and non-monolithic Vision Transformers on five small datasets including CIFAR10/100, CINIC10, SVHN, Tiny-ImageNet and two fine-grained datasets: Aircraft and Cars. Our approach consistently improves the performance of Vision Transformers while retaining their properties such as attention to salient regions and higher robustness. Our codes and pre-trained models are available at: https://github.com/hananshafi/vits-for-small-scale-datasets.

1 Introduction

Since their inception, Vision Transformers (ViTs) [13, 29, 44] have emerged as an effective alternative to traditional Convolutional Nerual Networks (CNNs) [21, 26, 33, 51, 53, 57]. The architecture of Vision Transformer is inspired by the prominent Transformer encoder [12, 56] used in natural language processing (NLP) tasks, which process data in the form of sequence of vectors or tokens. Similar to the word tokens in NLP Transformer, ViT typically splits the image into a grid of non-overlapping patches before passing them to a linear projection layer to adjust the token dimensionality. These tokens are then processed by a series of feed-forward and multi-headed self-attention layers. Due to their ability to capture global structure through self-attention [43], ViTs have found extensive applications in many tasks such as classification [13, 37, 40, 54, 55, 58, 60, 61, 62] object detection [2, 9, 28, 68], segmentation [48, 49], restoration [38], and 3D vision [4, 64].

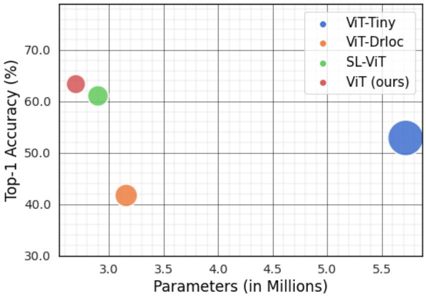

Figure 1: Our approach is simple in nature and yet outperforms [35, 39, 54] by notable margins both in terms of trainable parameters and generalization (top-1 accuracy) on Tiny-ImageNet [34]. This shows that inductive biases learned by our self-supervised approach serves an effective weight initialization to ease ViT optimization during supervised training. Our approach remains beneficial to different ViT designs as we thoroughly explored on multiple small-scale datasets in Sec. 4.1

Despite their advantages, ViTs fail to match the performance of CNNs when trained from scratch on small-scale datasets. This is primarily due to the lack of locality, inductive biases and hierarchical structure of the representations which is commonly observed in the CNN architectures [40, 58, 61]. As a result, ViTs require large-scale pre-training to learn such properties from the data [13] for better transfer learning to downstream tasks. Typically, ViTs are trained with a private JFT-300M [50] dataset with 303 million weakly supervised images or publicly available ImageNet-1k/22k datasets [11]. However, the absence of such large-scale pre-training hampers the performance of ViTs on small-scale datasets [35, 39].

To ease the optimization difficulties during ViT training, different architectural designs are proposed to induce the necessary inductive biases for the Vision Transformer [16, 37, 40, 58, 61]. These hybrid networks still remain sub-optimal for small datasets (Sec. 4.1) and either require further modifications to the loss functions [39] or network architecture [35]. Even with careful design choices, these methods remain sensitive to the type of data distribution *e.g.*, [39]'s performance degrades on Tiny-ImageNet, a complex data distribution as compared to other small datasets such CIFAR (Sec. 4.2).

To alleviate these problems, we propose an effective two-stage framework to train ViTs on small-scale low resolution datasets from *scratch*. **Low-resolution View Prediction as Weight Initialization Scheme:** We observe that ViTs are sensitive to weight initialization and converge to vastly different solutions depending on the network initialization (Sec. 3). Large-scale pre-training captures inductive biases from the data [13] and allows successful transfer learning on small datasets. In the absence of huge datasets, however, we hypothesize that ViTs can benefit from the inductive biases directly learned on the target small dataset such as CIFAR10 or CIFAR100. To this end, we introduce self-supervised weight learning scheme based on feature prediction of our *low-resolution* global and local views via self-distillation [3]. **Self-supervised to Supervised Learning for small-scale Datasets:** In the second stage, we fine-tune the same ViT network on the same target dataset using simply cross-entropy loss. Our approach therefore is agnostic to ViT architectures, independent to changes in loss functions, and provides significant gains in comparison to different weight initialization schemes [14, 20, 45, 67] and existing works [35, 39] (Fig. 1). We demonstrate the effectiveness of our method on five small datasets across different monolithic and non-monolith Vision Transformers (Sec. 4.1). Our contributions are as follows:

1. We propose a self-supervised weight learning scheme from low-resolution views created on small datasets. This serves as an effective weights initialization to successfully train ViTs from scratch, thus eliminating the need for large-scale pre-training.

2. Our proposed self-supervised inductive biases improve the performance of ViTs on small datasets without modifying the network architecture or loss functions.

3. We show that our training approach scales well with the input resolution. For instance, when trained on high-resolution samples, our method improves by 8% (CIFAR10) and 7% (CIFAR100) w.r.t the state-of-the-art baseline [39] for training ViTs on small datasets. Furthermore, we validate the efficacy of our method by observing its robustness against natural corruptions, and attention to salient regions in the input sample.

2 Related Work

In this section, we discuss the related existing works in the application of ViTs for small datasets, self-supervised learning and weight initialization. There have been attempts to train ViTs on ImageNet from scratch [16, 40, 54, 55, 58, 62]. [54] improves the performance of ViT through data augmentations, regularization, and knowledge distillation. [62] introduces a new image tokenization strategy by recursively aggregating the neighboring tokens in order to model the locality into the network. [40] introduces a hierarchical vision transformer which processes the input at various scales and limits the self-attention to non-overlapping patches by the use of shifted windows. [58] replaces the projection and multilayer perceptron layers with convolution layers in order to introduce the shift, scale, and distortion invariance. Recently, there have been few attempts to train ViTs on small datasets [35, 39].

Vision Transformers for Small Datasets: [35] applies a series of augmentations [8, 24, 52, 63, 65] on the input data and introduces shifted patch tokenization (SPT) and locality self-attention (LSA), which enable ViT to learn from scratch even on small datasets. [39] trains a ViT with an additional proxy task of learning the spatial location of the encoded image tokens in order to learn the phenomena of locality. Different from these approaches, we show that without any modification to the internal layers or addition of new loss function, our approach learns better generalizable features from the existing small target datasets.

Self-supervised learning: In recent years, several self-supervised techniques have been proposed to pre-train ViTs [1, 3, 22, 36, 59, 66]. In [3], the pretext task is to match the local and global features by minimizing the cross entropy loss. In [22, 59], the input patches are masked and the network is tasked to predict the masked pixels. [66] pretrains the network with two pretext tasks based on local-global feature matching and masked encoding. All these methods have shown impressive results on Imagenet linear evaluation and have been applied to numerous downstream tasks. Such pre-training strategies are computationally expensive and are designed for large sized datasets at higher resolution. Instead, we apply self-supervision for low-resolution small dataset to observe decent improvements.

Weight Initialization. ImageNet pre-trained weights have been the default choice for network initialization in most computer vision tasks. However, given the amount of training time and computational resources required for such training, some past works [27, 67] have proposed methods to efficiently initialize the model weights. [27] introduces a weight initialization scheme that eliminates the problem of learning rate warmup in NLP transformers, enabling deep transformer models to train without difficulty. [67] presents a model agnostic initialization scheme which adjusts the norm of each network layer by introducing a multiplier variable in front of each parameter block. Apart from these approaches, majority of the models are initialized using the basic weight initialization schemes [14, 20, 45], etc. Different from these approaches, we learn the initial weights of the ViT using self-supervised learning directly from small datasets without any changes in the architecture or the optimizer.

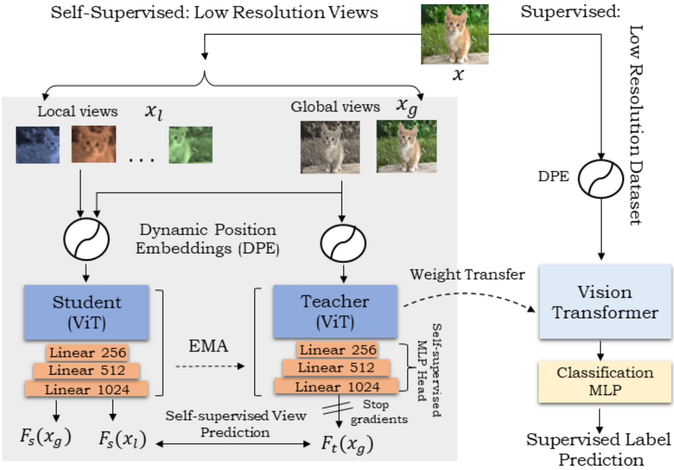

Figure 2: Our Training framework: At the pre-training stage, we learn self-supervised weights by predicting *low resolution* local and global views via self-distillation through student and teacher networks [8]. At the fine-tuning stage, self-supervised weights are used to initialize ViT for supervised learning. Our self-supervised inductive biases ease ViT optimization during supervised learning on small-scale datasets.

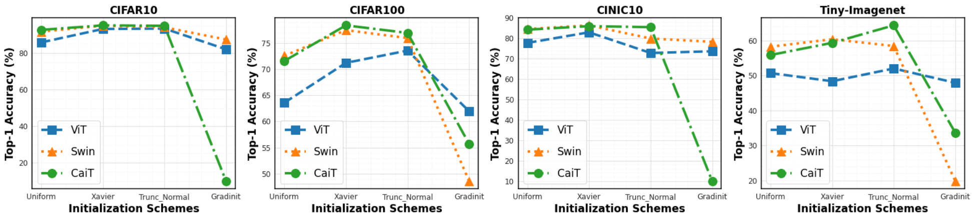

Figure 3: We train Vision Transformers with different weight initialization schemes including 'Uniform', 'Xavier [14]', 'Truncated normal [20]', and 'Gradinit [67]'. All models are trained for 100 epochs. We follow the default training setting for small-scale datasets as proposed by [35] in all our experiments for consistent comparisons. We observe that ViT training can be unstable depending upon weight initializations *e.g.* CaiT [55] performs poorly when initialized with Gradinit. Similarly, the generalization of ViT [54] and Swin [40] varies a lot with different weight initialization methods.

3 Vision Transformer Training on Small Datasets

Inherent inductive biases allow to train CNNs on small-scale datasets from scratch. Vision Transformer on the other hand need large-scale pre-training for successful transfer learning [13, 54]. Our goal is to eliminate the large-scale data requirement and train ViTs directly on a given small dataset. Neural networks are sensitive to weight initialization schemes [19]. We observe that ViTs' convergence is affected by the weight initialization scheme (Fig. 3) during training from scratch. Therefore, we propose to learn the weight initialization from the given data distribution ($\mathcal{Q}$) to inject the necessary inductive biases within ViT architecture. Our training approach (Fig. 2), hence, consists of two stages including ***a) Self-supervised View Prediction*** followed by ***b) Supervised Label Prediction*** tasks. Note that both of these tasks are learned on the same data distribution ($\mathcal{Q}$) with the same model backbone $\mathcal{F}$. The only architectural difference between the both learning tasks is the self-supervised and supervised MLP projection (Fig. 2). In this manner, our approach is independent to large-scale pre-training. We describe the ViT encoders designs next in Sec. 3.1 before explaining our proposed training in detail (Sec. 3.2 and 3.3).

Attributes →	Depth	Patch-size	Token Dimension	Heads	MLP-ratio	Window-size
ViT [54]	9	[4,8]	192	12	2	-
Swin [40]	[2,4,6]	[2,4]	96	[3,6,12]	2	4
CaiT [55]	24	[4,8]	192	4	2	-

Table 1: Details of ViT encoders used in our proposed training approach (Fig. 2).

3.1 Vision Transformer Encoders

We train different monolithic [54] and non-monolithic (Swin [40] and CaiT [55]) ViTs (Table 1). These ViTs are originally designed for higher resolution inputs (224 or 384) with patch sizes of 16 or 32. However, small-scale datasets have low resolution inputs *e.g.* 32 or 64 in the case of CIFAR and Tiny-ImageNet, respectively. Therfore, we reduce the patch size for such low resolution inputs. Specifically, we set a patch size of 8 and 4 for an input of size 64x64 and 32x32, respectively. Similarly, we adopt the original ViT designs for small datasets following [46]. Table 1 presents the high level details of these network architectures. Further ablations with different ViT attributes (*e.g.* depth, and heads) are provided in Appendix A.

3.2 Self-supervised View Prediction as Weight Initialization Scheme

As mentioned earlier, we learn to initialize wights for low resolution small-scale datasets via self-supervised training. Among many self-supervised learning methods [5, 6, 47], we adopt view prediction strategy based on [3]. Thus, our self-supervised approach does not require memory bank, large batch-size, or negative mining. The self-supervised weights are then used for initialization during fine-tuning stage directly from the low-resolution dataset. Our view prediction pre-training uses a student ($\mathcal{F}_s$) and teacher ($\mathcal{F}_t$) setup to predict different views of the same input sample from each other and thus follows the learning paradigm of knowledge self-distillation [18]. Both student and teacher represent the same network (Fig. 2) but process different views as explained below.

Self-supervised View Generation and Prediction: Consider a low resolution input $\boldsymbol{x}$ sampled from a small data distribution $\mathcal{Q}$. We define the height and width of the low-resolution input $\boldsymbol{x}$ by h and w, respectively. During pre-training, the input is distorted and augmented to generate global ($\boldsymbol{x}_g$) and local ($\boldsymbol{x}_l$) views. We use standard augmentations [3] which preserve the semantic information of each selected view. These augmentations include color jitter, gray scaling, solarization, random horizontal flip and gaussian blur. *Global views* are generated by randomly selecting regions in the input image covering more than 50% of the input portion, while *local views* are generated by randomly selecting regions covering around 20-50% portion of the input. The global and local views are further resized such that the ratio of area of local to global view is 1:4. As for instance, the global view generated for CIFAR sample is resized to a dimension of 32x32 and the local view is resized to 16x16. We generate 2 global and 8 local views in our case. *Dynamic Position Embeddings (DPE):* The number of input tokens vary based on the view size, so we use Dynamic Position Embeddings (DPE) [3, 47] which interpolates for the missing tokens of smaller views with height and width less than the original sample size $h \times w$. Both student and teacher networks process these multi-sized views and output the corresponding feature representations. *Self-supervised MLP Projection:* The features representation of each view is further processed by a 3-layer MLP of the student and teacher networks. The multi-layer projection performs better than a single layer MLP [3]. Thus, each low-resolution view is converted into 1024 dimensional feature vector. Ablative analysis on the effect of the output size of self-supervised MLP projection head is provided in Sec. 4.2.

The teacher network processes the global views to generate target features ($\mathcal{F}_{g_t}$) while all the local and global views are forward-passed through the student network to generate predicted features ($\mathcal{F}_{g_s}$) and ($\mathcal{F}_{l_s}$). These features are normalized [3, 47] to obtain $\tilde{\mathcal{F}}_{g_t}, \tilde{\mathcal{F}}_{g_s}$, and $\tilde{\mathcal{F}}_{l_s}$. We update the student's parameters by minimizing the following objective:

$$\mathcal{L} = -\tilde{\mathcal{F}}_{g_t} \cdot \log(\tilde{\mathcal{F}}_{g_s}) + \sum_{i=1}^{n} -\tilde{\mathcal{F}}_{g_t} \cdot \log(\tilde{\mathcal{F}}_{l_s}^{(i)}), \tag{1}$$

where n represent number of local views specifically set to 8. The teacher parameters are updated via exponential moving average of the student weights using: $\theta_t \leftarrow \lambda\theta_t + (1-\lambda\theta_s)$ where θ_t and θ_s denote the parameters of teacher and student network respectively and, λ follows the cosine schedule from 0.996 to 1 during training. Further, we apply centering and sharpening operations to the teacher output. This way our method avoid any mode collapse similar to BYOL [17] and converge to unique solution [3].

Our self-supervised view prediction objective (Eq. 1) on low resolution inputs induces locality in the Vision Transformer and encourages better intermediate feature representations which further aids during the fine-tuning stage on the same dataset.

3.3 Self-supervised to Supervised Label Prediction

We initialize a given model with weights learned via our self-supervised approach on the target dataset and then fine-tune the model on the same corresponding dataset. This is in contrast to existing practices of initializing the models with different initialization schemes [45] or ImageNet pre-trained weights. We transfer weights from the teacher network (Fig. 2) and replace the self-supervised MLP projection head with a randomly initialized MLP classifier. The model is then trained via supervised objective as follows:

$$\mathcal{L}_{CE} = -\sum_{i=1}^{k} y_i \cdot log(\mathcal{F}(\boldsymbol{x})_i), \tag{2}$$

where k is the output dimension of the final classifier and y represents the one-hot encoded ground-truth. We note that teacher provides high quality target features during pre-training and hence prove useful for the fine-tuning stage [3]. Further, the ablation on the effect self-supervised weights is provided in Table 7.

4 Experimental Protocols

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

Datasets: We validate our approach on five small-scale, low-resolution datasets including Tiny-Imagenet [34], CINIC10 [10], CIFAR10, CIFAR100 [32], SVHN [15] and two fine-grained datasets including Aircraft [42] and Cars [31]. Details about the dataset size, sample resolution and the number of classes are provided in Table 2.

Self-supervised Training Setup: We train all models with the Adam optimizer [30] and a batch size of 256 via distributed learning over 4 Nvidia V100 32GB GPUs . We linearly ramp up the learning rate during the first 10 epochs using: $lr = 0.0005 * \frac{\text{Batch size}}{256}$. After first 10 epochs, learning rate follows the cosine schedule [41]. We re-scale the student and teacher

Dataset	Train Size	Test Size	Dimensions	# Classes
Tiny-Imagenet [34]	100,000	10,000	64x64	200
CIFAR10 [32]	50,000	10,000	32x32	10
CIFAR100 [32]	50,000	10,000	32x32	100
CINIC10 [10]	90,000	90,000	32x32	10
SVHN [15]	73,257	26,032	32x32	10
Aircraft [42]	6,667	3533	224x224	102
Cars [31]	8,144	8,041	224x224	196

Table 2: Details of datasets in terms of sample size and resolution used in our experiments. We propose to learn Self-supervised initialization directly from small datasets with default resolution.

Model	Params(M)	Tiny-Imagenet	CIFAR10	CIFAR100	CINIC10	SVHN
ResNet56 [21]	0.9	56.51	94.65	74.44	85.34	97.61
ResNet110 [21]	1.7	59.77	95.27	76.18	86.81	97.82
EfficientNet B0 [53]	4.0	55.48	88.38	61.64	75.64	96.06
ResNet18 [21]	11.6	53.32	90.44	64.49	77.79	96.78
ViT (scratch)	2.8	57.07	93.58	73.81	83.73	97.82
SL-ViT [35](Arxiv'21)	2.9	61.07	94.53	76.92	84.48	97.79
ViT-Drloc [39] (NeurIPS'21)	3.15	42.33	81.00	58.29	71.50	94.02
ViT (Ours)	2.8	**63.36**	**96.41**	**79.15**	**86.91**	**98.03**
Swin (scratch)	7.1	60.05	93.97	77.32	83.75	97.83
SL-Swin [35] (Arxiv'21)	10.2	64.95	94.93	79.99	87.22	97.92
Swin-Drloc [39] (NeurIPS'21)	7.7	48.66	86.07	65.32	77.25	95.77
Swin (Ours)	7.1	**65.13**	**96.18**	**80.95**	**87.84**	**98.01**
CaiT (scratch)	7.7	64.37	94.91	76.89	85.44	98.13
SL-CaiT [35] (Arxiv'21)	9.2	67.18	95.81	80.32	86.97	**98.28**
CaiT-DRLoc [39] (NeurIPS'21)	8.5	45.95	82.20	56.32	73.85	19.59
CaiT (Ours)	7.7	**67.46**	**96.42**	**80.79**	**88.27**	98.18

Table 3: Our approach performs favorably well against different ViT baselines [35, 39] as well as CNNs without adding any additional parameters or requiring changes to architecture or loss functions. Note that all methods are trained on the original input resolution as provided in Table 1.

Model	Method	Input Resolution	Patch-size	No. of Tokens	Params(M)	CIFAR10	CIFAR100	SVHN
Swin	Drloc [39]	224x224	4	3136	28	83.89	66.23	94.23
Swin (Ours)	Drloc [39]	32x32	2	64	7.7	86.07	65.32	95.77
Swin (Ours)	Ours	224x224	4	3136	7.7	92.04	73.46	96.86

Table 4: In comparison to [39], the performance of our approach improves significantly on higher resolution. Thus our approach proves effective on both low as well as high input resolutions.

Method	Aircraft	Cars
ViT-Drloc [38]	10.40	13.82
ViT (Ours)	**66.04**	**43.89**

Table 5: Our approach outperforms the existing SOTA approach on the finegrained datasets (Top-1 accuracy).

outputs using a temperature parameter [12] which is set to 0.1 for the student network, while it follows a linear warm-up from 0.04 to 0.07 for the teacher network.

Supervised Training Setup: We use the training framework of [35] for supervised learning and apply standard data augmentations for consistency. Specifically, we use cutmix [63], mixup [63], auto-augment [7], and repeated augment [8]. Further, we also use label smoothing [52], stochastic depth [25], and random erasing [65]. We train all models for 100 epochs with a batch size of 256 on a single Nvidia V100 32GB GPU. We use Adam optimizer [30] with a learning rate of 0.002 and learning decayed rate of 5e-2 with cosine scheduling.

Data	ViT (scratch)	SL-ViT	ViT (Ours)	Swin (scratch)	SL-Swin	Swin (Ours)
CIFAR10	39.93	26.42	**26.01**	36.13	26.28	**25.38**
CIFAR100	65.04	48.56	**48.10**	53.83	47.29	**45.10**

Table 6: Mean corruption error (*lower is better*) is reported against 18 natural corruptions [23]. Our training method improves model robustness against such real world corruptions such as fog, rain, etc.

4.1 Results

Generalization: We observe generalization of different methods with a comparative analaysis presented in Table 3 across 3 different ViT architectures (Table 1). We keep a patch size of 8 for Tiny-Imagenet to generate 16 number of input tokens for ViT and CaiT architectures. We reduce the patch size to 4 so that the resultant number of tokens become 64 for all other datasets such as CIFAR, SVHN, and CINIC10. Similarly, for Swin architecture, we use a patch size of 4 for Tiny-Imagenet to obtain 64 tokens, while for other datasets, we use a patch size of 2 which produces 256 number of input tokens. We consistently follow these architectural settings for all the baselines (Table 3). Our approach consistently performs better as compared to recent state-of-the-art methods ([35, 39]) for ViTs training on small-scale datasets (Table 3). Particularly, we observe a significant gain for the difficult cases where the ratio of number of classes vs. input samples is higher *e.g.* CIFAR100 and Tiny-ImageNet (Table 3). In this manner, our approach paves the way to adopt ViTs to small datasets that also outperforms CNN based models. The effect of our self-supervised wight initialization on convolutional networks is provided in Appendix B.

Performance on fine-grained datasets: We test our approach on two fine-grained datasets such as Aircarft and Cars, and present the top-1 accuracy results in Table 5. We observe that our proposed approach performs well on fine-grained datasets compared to [39].

Robustness to Input Resolution and Patch Sizes: A recent method [39] projects the input samples to higher resolution to train Vision Transformer *e.g.* input resolution of 32x32 for CIFAR is re-scaled to 224x224 during training. This significantly increases the number of input tokens and hence the quadratic complexity within self-attention (Table. 4). In comparison, our approach successfully trains ViTs on low resolution inputs while being computationally efficient. Our method scales well on high resolution inputs and outperforms [39] by notable margins (Table 4).

Robustness to Natural Corruptions: We analyse the mean corruption error on CIFAR10 and CIFAR100 in Table 6. Our training approach increases the model robustness against 18 natural corruptions such as fog, rain, noise, and blur, etc. [23].

Attention to Salient Regions: We visualize self-attention in Fig. 4. The attention scores of the class token computed across the attention heads for the last ViT block is projected onto the unseen test samples of Tiny-ImageNet. Our proposed approach is able to capture the shape of the salient objects more efficiently with minimal or no attention to the background as compared to the baseline approaches where the attention is more spread out in the background and they completely fail to capture the shape of the salient object in the image.

4.2 Ablative Analysis

Effect of Data Size on Self-supervised Weight Initialization: We study the effect of data size on our self-supervised learning for weight initialization (Fig. 5). We train the ViT models [54] on 25%, 50% and 75% of the training samples across 3 datasets: CIFAR10, CIFAR100 and Tiny-Imagenet datasets. In case of CIFAR10, our approach achieves more than

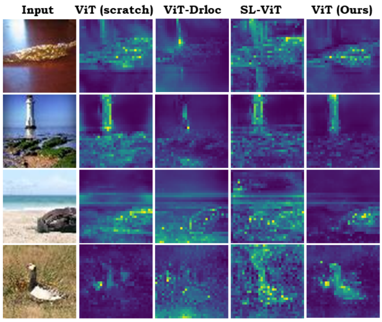

Figure 4: Our approach captures the salient objects in the image in comparison to baseline methods for which the attention is dispersed in the background.

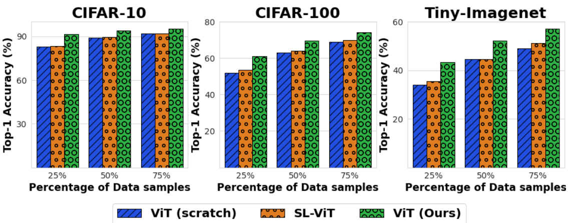

Figure 5: We demonstrate the efficiency of our approach by varying the training samples of CIFAR10, CIFAR100 and Tiny-Imagenet. Specifically we train the models with 25%, 50% and 75% of training data. *Our approach consistently performs better even with limited training data.*

Model	Weights	CIFAR100	Tiny-Imagenet
ViT	Student	77.27	61.02
ViT	Teacher	**79.15**	**63.36**

Table 7: Self-supervised teacher weights transfer well as compared to the student. This finding is consistent with [3].

Local View	Global View	CIFAR10		CIFAR100	
		ViT	Swin	ViT	Swin
(0.1 , 0.4)	(0.4 , 1.0)	77.36	67.82	79.52	74.10
(0.15 , 0.4)	(0.4 , 1.0)	78.90	68.12	78.14	74.06
(0.2 , 0.5)	(0.5 , 1.0)	77.48	67.90	79.53	74.05
(0.2 , 0.5)	(0.6 , 1.0)	78.87	65.52	79.51	74.09
(0.2 , 0.5)	**(0.7 , 1.0)**	**79.19**	**71.64**	**79.78**	**74.67**

Local View	Global View	Tiny-ImageNet	
		ViT	Swin
(0.1 , 0.3)	(0.3 , 1.0)	64.37	74.33
(0.15 , 0.45)	(0.45 , 1.0)	62.02	74.09
(0.2 , 0.4)	**(0.5 , 1.0)**	**64.82**	**75.13**
(0.2 , 0.4)	(0.6 , 1.0)	62.03	57.33
(0.3 , 0.5)	(0.5 , 1.0)	61.69	73.92

Table 8: We demonstrate the impact of local-global crop ratios chosen during self-supervised training on the Top-1 train accuracy scores of CIFAR10 and CIFAR100 (left), and Tiny-Imagenet (right).

90% top-1 accuracy with just 25% of data and outperforms other approaches with notable margins. We observe a similar trend with CIFAR100 and Tiny-Imagenet datasets.

Effect of Local-Global Crop Ratio: We generate local and global views by randomly cropping certain regions from the original input image. The cropped area of each generated view is chosen from a specified range of values w.r.t the original input size. We analyse the impact of the range of aspect ratios for local and global views w.r.t the original input size in Table 8. The original input size of Tiny-Imagenet is 2x times greater than the other datasets used in our experiments, therefore, we use a modified range of local-global aspect ratios as shown in Table 8 (right). We observe that range of aspect ratios between (0.2, 0.4) for local view and (0.5, 0.1) for global view works well for the Tiny-ImageNet. Similarly, for the other relatively lower-resolution datasets, the optimal aspect ratios are in the range of (0.2, 0.5) and (0.7, 1.0) for the local and global views, respectively (Table 8).

Effect of Self-supervised MLP Dimensions: Table 9 shows the effect of the output head dimension of our self-supervised projection MLP on the model generalization during supervised fine-tuning stage. We fix the local-global aspect ratio to their optimal values and ablate over a range of MLP head dimensions. Based on the top-1 accuracy results on train set (Table 9), we choose the dimension of size 1024 for all our experiments.

Effect of Teacher Vs. Student Weights Transfer: We compare the performance of the ViT initialized with our self-supervised weights from student and teacher networks (Table 7). We observe higher generalization (top-1 accuracy) with the teacher weights that corroborates our strategy of choosing teacher rather than student weights for the supervised training stage.

Performance comparison with self-supervised learning based CNN: We provide a com-

SSL Head	CIFAR10		CIFAR100		Tiny-Imagenet	
	ViT	Swin	ViT	Swin	ViT	Swin
512	78.77	79.46	71.63	74.04	66.84	74.10
1024	**79.19**	**79.78**	**71.64**	**74.67**	**65.82**	**75.13**
2048	78.83	79.48	71.15	73.87	65.05	73.95
4096	78.92	79.50	71.03	74.21	66.48	74.03

Table 9: *Effect of Self-supervised MLP Projection Head:* We observe that MLP head dimension of 1024 gives better overall results on train set across 3 datasets using ViT and Swin architectures.

Model	Initialization	CIFAR100	Tiny-Imagenet
ResNet-18	SS (ours)	65.00	53.48
ViT	SS (ours)	**79.15**	**63.36**

Table 10: Top-1 accuracy comparison of self-supervised (SS) trained CNN with Ours.

Method	Tiny-Imagenet	CIFAR10	CIFAR100
SimCLR [5]	58.87	93.50	74.77
MOCO-V3 [6]	52.39	93.95	72.22
Ours	**63.36**	**96.41**	**79.15**

Table 11: Comparison of other existing self-supervised learning techniques with ours using basic ViT architecture.

Method	Epochs	CIFAR100
ViT-Drloc [39]	600 (supervised)	68.29
ViT (Ours)	200(self-supervised) + 100(supervised)	**76.08**

Table 12: Our approach is efficient in terms of epochs used and outperforms the current approach in terms of Top-1 accuracy.

Self-supervised MLP layers	CIFAR100	Tiny-Imagenet
1-Layer	76.69	60.54
3-Layer (Ours)	**79.15**	**63.36**

Table 13: Top-1 accuracy comparison of 3-Layer MLP with 1-Layer MLP training

Head size	CIFAR100	Tiny-Imagenet
65536 [3]	77.42	60.77
1024 (Ours)	**79.15**	**63.36**

Table 14: Top-1 accuracy comparison of the size of projection head used during self-supervised training

parison of ViT with ResNet18 (2.8 vs. 11.6 million parameters) in Table 10. ViT's performance improves significantly in comparison to self-supervised CNN. This finding is consistent with [3]. In addition, we compare our proposed self-supervised approach with different contrastive self-supervised methods [5, 6] that are mainly studied for CNNs (Table 11). Our method provides SOTA results for ViTs in comparison to self-supervised CNN.

Efficiency in terms of epochs: We further observe that our method trained for 300 epochs (200 for self-supervised view matching and 100 for supervised label prediction) outperforms the current SOTA approach [39] trained for even 600 epochs (Table 12).

Self-supervised MLP layers: We modify the self-supervised projection MLP which reduces complexity and increases generalization as shown in Table 13.

Analysis of MLP Head: We observe that larger the size of MLP such as 65536 [3], the lower is the performance on small-scale datasets (Table 9 and Table 14). This is because the large size of MLP head might result in overfitting the features of low resolution views.

5 Conclusion

In this work, we introduce an effective strategy to train Vision Transformers on small-scale low-resolution datasets without large-scale pre-training. We propose to learn self-supervised inductive biases directly from the small-scale datasets. We initialize the network with the weights learned through self-supervision and fine-tune it on the same dataset during the supervised training. We show through extensive experiments that our method can serve as a better initialization scheme and hence allows to train ViTs from scratch on small datasets while performs favorably well w.r.t the existing state-of-the-art methods. Further, our approach can be used in a plug-and-play manner for different ViT designs and training frameworks without any modifications to the architectures or loss functions.

References

[1] Sara Atito, Muhammad Awais, and Josef Kittler. Sit: Self-supervised vision transformer. *arXiv preprint arXiv:2104.03602*, 2021.

[2] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[3] 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.

[4] Shuo Chen, Tan Yu, and Ping Li. Mvt: Multi-view vision transformer for 3d object recognition. *arXiv preprint arXiv:2110.13083*, 2021.

[5] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[6] Xinlei Chen, Saining Xie, and Kaiming He. An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9640–9649, 2021.

[7] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[8] 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.

[9] Zhigang Dai, Bolun Cai, Yugeng Lin, and Junying Chen. Up-detr: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1601–1610, 2021.

[10] Luke N. Darlow, Elliot J. Crowley, Antreas Antoniou, and Amos J. Storkey. Cinic-10 is not imagenet or cifar-10, 2018. URL https://arxiv.org/abs/1810.03505.

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

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

[13] 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. *arXiv preprint arXiv:2010.11929*, 2020.

[14] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR.

[15] 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, 2013. URL https://arxiv.org/abs/1312.6082.

[16] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Herv'e J'egou, and Matthijs Douze. Levit: a vision transformer in convnet's clothing for faster inference. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12239–12249, 2021.

[17] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in neural information processing systems*, 33:21271–21284, 2020.

[18] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent-a new approach to self-supervised learning. *Advances in Neural Information Processing Systems*, 33:21271–21284, 2020.

[19] Boris Hanin and David Rolnick. How to start training: The effect of initialization and architecture. *Advances in Neural Information Processing Systems*, 31, 2018.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[22] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[23] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2018.

[24] 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.

[25] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *European conference on computer vision*, pages 646–661. Springer, 2016.

[26] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017.

[27] Xiao Shi Huang, Felipe Perez, Jimmy Ba, and Maksims Volkovs. Improving transformer optimization through better initialization. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 4475–4483. PMLR, 13–18 Jul 2020.

[28] Aishwarya Kamath, Mannat Singh, Yann LeCun, Gabriel Synnaeve, Ishan Misra, and Nicolas Carion. Mdetr-modulated detection for end-to-end multi-modal understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1780–1790, 2021.

[29] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM Computing Surveys (CSUR)*, 2021.

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

[31] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *4th International IEEE Workshop on 3D Representation and Recognition (3dRR-13)*, Sydney, Australia, 2013.

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

[33] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012.

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

[35] Seung Hoon Lee, Seunghyun Lee, and Byung Cheol Song. Vision transformer for small-size datasets. *arXiv preprint arXiv:2112.13492*, 2021.

[36] 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. *arXiv preprint arXiv:2106.09785*, 2021.

[37] Yawei Li, K. Zhang, Jie Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *ArXiv*, abs/2104.05707, 2021.

[38] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021.

[39] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco Nadai. Efficient training of visual transformers with small datasets. *Advances in Neural Information Processing Systems*, 34, 2021.

[40] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021.

[41] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[42] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013.

[43] Muhammad Muzammal Naseer, Kanchana Ranasinghe, Salman H Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems*, 34:23296–23308, 2021.

[44] Namuk Park and Songkuk Kim. How do vision transformers work? In *International Conference on Learning Representations*, 2022.

[45] 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. *Advances in neural information processing systems*, 32, 2019.

[46] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34, 2021.

[47] Kanchana Ranasinghe, Muzammal Naseer, Salman Khan, Fahad Shahbaz Khan, and Michael Ryoo. Self-supervised video transformer. In *IEEE/CVF International Conference on Computer Vision and Pattern Recognition*, June 2022.

[48] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 12179–12188, 2021.

[49] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7262–7272, 2021.

[50] 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.

[51] 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.

[52] 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.

[53] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning*, pages 6105–6114. PMLR, 2019.

[54] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*, pages 10347–10357. PMLR, 2021.

[55] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 32–42, 2021.

[56] 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.

[57] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3349–3364, 2020.

[58] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 22–31, 2021.

[59] 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. *arXiv preprint arXiv:2111.09886*, 2021.

[60] Weijian Xu, Yifan Xu, Tyler Chang, and Zhuowen Tu. Co-scale conv-attentional image transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9981–9990, 2021.

[61] Kun Yuan, Shaopeng Guo, Ziwei Liu, Aojun Zhou, Fengwei Yu, and Wei Wu. Incorporating convolution designs into visual transformers. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 559–568, 2021.

[62] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Francis E. H. Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 538–547, 2021.

[63] 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.

[64] H. Zhao, L. Jiang, J. Jia, P. Torr, and V. Koltun. Point transformer. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16239–16248, Los Alamitos, CA, USA, oct 2021. IEEE Computer Society. doi: 10.1109/ICCV48922.2021.01595.

[65] 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.

[66] Jinghao Zhou, Chen Wei, Huiyu Wang, Wei Shen, Cihang Xie, Alan Yuille, and Tao Kong. ibot: Image bert pre-training with online tokenizer, 2021.

[67] Chen Zhu, Renkun Ni, Zheng Xu, Kezhi Kong, W. Ronny Huang, and Tom Goldstein. Gradinit: Learning to initialize neural networks for stable and efficient training. In *NeurIPS*, 2021.

[68] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020.