Where are my Neighbors? Exploiting Patches Relations in Self-Supervised Vision Transformer

Guglielmo Camporese
guglielmo.camporese@phd.unipd.it
Elena Izzo
elena.izzo@phd.unipd.it
Lamberto Ballan
lamberto.ballan@unipd.it

Visual Intelligence and Machine
Perception (VIMP) Group
University of Padova
Padova, Italy

Abstract

Vision Transformers (ViTs) enabled the use of the transformer architecture on vision tasks showing impressive performances when trained on big datasets. However, on relatively small datasets, ViTs are less accurate given their lack of inductive bias. To this end, we propose a simple but still effective Self-Supervised Learning (SSL) strategy to train ViTs, that without any external annotation or external data, can significantly improve the results. Specifically, we define a set of SSL tasks based on relations of image patches that the model has to solve before or jointly the supervised task. Differently from ViT, our RelViT model optimizes all the output tokens of the transformer encoder that are related to the image patches, thus exploiting more training signals at each training step. We investigated our methods on several image benchmarks finding that RelViT improves the SSL state-of-the-art methods by a large margin, especially on small datasets. Code is available at: https://github.com/guglielmocamporese/relvit.

1 Introduction

Vision Transformer (ViT) [11] is a model that has been recently developed to address computer vision tasks, such as image classification and object detection. It builds on the Transformer architecture [31], the state of the art in natural language processing, considering patches as parts of the image such as words are parts of a sentence. Although ViT is a valid convolution neural networks (CNNs) competitor, showing comparable and even better results [29], a high-level performance is obtained only when training the model on huge amounts of data. To deal with this issue, aiming at better generalization levels, some recent works modified the attention backbone of ViT introducing hierarchical feature representation [22], progressively tokenization of the image [38] or a shrinking pyramid backbone [33]. Other works, inspired by BERT [8], used a Self-Supervised Learning (SSL) paradigm by firstly pre-training ViT on a massive amount of unlabelled data and, subsequently, training a linear classifier over the frozen feature of the model or fine-tuning the pre-trained model to a

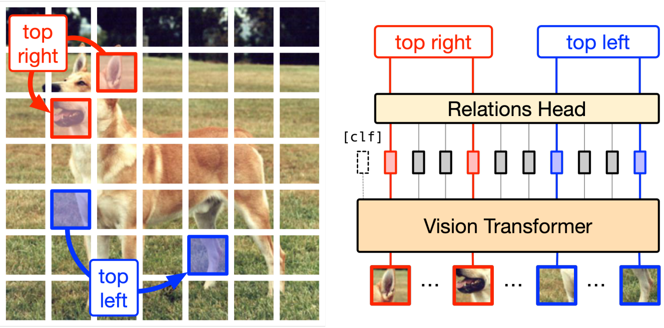

Figure 1: ViT splits the image into patches, and only the output classifier token that aggregates the global information of the image is directly trained with the label annotation. On the other hand, all the other tokens related to the input patches of the image are not used. Our work proposes several self-supervised tasks based on image patches used by ViT that can be beneficial for building strong internal features of the model. An example of such a task is the recognition of spatial relations among all the patches (in the picture above, only two relations examples are reported), whereas other tasks are described in Section 3.

downstream task [3, 5]. These variants outperform similar-size ResNet trained on ImageNet, although experiments carried out on smaller datasets are still limited.

In this paper, we investigate ViTs on small datasets by introducing SSL tasks that face the problem of learning spatial relations among pairs of patches.In particular: *(i)* We propose and investigate various SSL tasks based on image patches, naturally used by ViTs, showing the improvement optimizing all input tokens, not only the classification one; *(ii)* We show that our SSL tasks are beneficial for the model under two different settings: when the model is pre-trained from scratch on our SSL tasks and subsequently fine-tuned on classification, and when the model is jointly trained from scratch for classification and SSL tasks; *(iii)* We show that on small datasets, both ViT and its variants jointly trained from scratch on our SSL tasks reach better performance with respect to similar state-of-the-art models, and they overcome the same architectures trained with full supervision.

2 Related Work

Our work focuses on the recently proposed Vision Transformer [11], and it introduces several self-supervised strategies to learn internal representations for downstream tasks. Here we highlight some of the most relevant recent efforts to move beyond fully supervised learning of vision models by exploiting self-supervised learning and pre-training strategies.

Self-Supervised Learning in Vision. Semi-supervised learning methods for image classification share a common workflow that foresees the definition of a pretext task on unlabelled images and the subsequent evaluation of the learned features through the training of a linear classifier over the frozen representations or by the fine-tuning of the pre-trained model to a downstream task. Previous works on Self-Supervised Learning (SSL) investigate discriminative strategies, for instance classification [4, 6, 10, 15, 35], where each image has its own different class and the model is trained for discriminating them up to data perturbations. However, such modeling frameworks do not scale with the increasing number of input samples as the model is trained to discriminate between all the images. Other works on self-supervised learning focus on recognizing known perturbation functions on augmented images [10, 13], by using image patches for predicting their relative positions [9, 21, 26], counting features [27], image colorization [39], denoising autoencoders, and context encoders [28, 32, 40]. Other recent ideas focus on contrastive methods [14] where the model is trained to attract positive sample pairs and at the same time to repulse negative sample pairs [16, 18, 30, 35, 37]. More recent methods explore siamese architectures [2, 3, 4, 5] where the representation of the student branch is compared and pushed towards the representation of the teacher that sees a different view of the input image.

Vision Transformers. Transformers [31] are architectures designed for processing sequential data and, through the attention layers, they enable global dependencies among the input tokens. Originally, the transformer architecture shined on NLP-related tasks raising the hypothesis that even on vision could have potentially been beneficial. Despite the substantial difference between text and images, a recent breakthrough model developed for image classification, dubbed Vision Transformer [11], adopted the transformers for images by treating the image as a sequence of patches like words are considered for phrases. This work highlights the high potential of ViTs when they are trained on large datasets. However, the high performances are not maintained when dealing with small datasets due to the lack of inductive bias of the architecture. As result, in the last few months many ViT variants emerged adapting the transformer architecture for the computer vision tasks. Some works replaced the rectangular backbone characterizing standard ViT with a pyramidal structure [12, 17, 22, 33, 34, 38] obtaining an improvement in accuracy at small/medium scale but increasing complexity and hyperparameters. Other works introduced effective design principals of CNNs into ViT [7, 34, 36] in order to improve performance and robustness.

3 Our Method

The ViT model takes an image $\boldsymbol{x} \in \mathbb{R}^{C \times H \times W}$ as input, divides it into N patches $x_p \in \mathbb{R}^{N \times (P^2 \cdot C)}$ of C channels and size (P, P) and makes the output prediction as follows:

$$\boldsymbol{z}_0 = \left[\boldsymbol{x}_{\text{class}}, \boldsymbol{x}_p^1 \boldsymbol{E}, \ldots, \boldsymbol{x}_p^N \boldsymbol{E}\right] + \boldsymbol{E}_{\text{pos}} \tag{1}$$

$$\boldsymbol{z}'_\ell = \text{MSA}(\text{LN}(\boldsymbol{z}_{\ell-1})) + \boldsymbol{z}_{\ell-1}, \quad \ell = 1 \ldots L \tag{2}$$

$$\boldsymbol{z}_\ell = \text{MSA}(\text{LN}(\boldsymbol{z}'_\ell)) + \boldsymbol{z}'_\ell, \quad \ell = 1 \ldots L \tag{3}$$

$$\boldsymbol{y} = \text{LN}(\boldsymbol{z}_L^0) \tag{4}$$

where $\boldsymbol{z}_\ell^i$ is the i-th token of the ℓ-th layer, $\boldsymbol{E}$ is the projection matrix of the input patches, $\boldsymbol{E}_{\text{pos}} \in \mathbb{R}^{(N+1) \times D}$ is the learned positional embedding matrix, $\text{MSA}(\cdot)$ is the multi-head self-attention layer [31], and $\text{LN}(\cdot)$ is the layer normalization block [1]. As done in BERT [8], the model uses a randomly initialized classifier token $\boldsymbol{x}_{\text{class}}$ that, at the end, is taken as the reference for making the prediction $\boldsymbol{y}$.

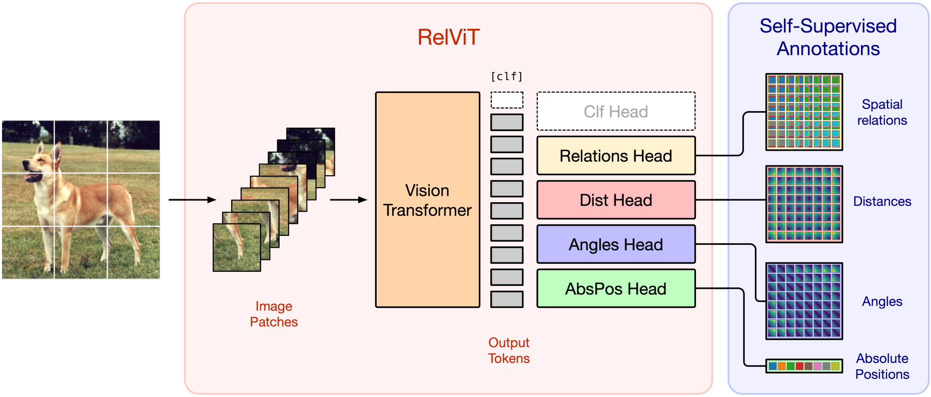

Figure 2: Our RelViT architecture. The image is partitioned into non-overlapping patches and fed to the vision transformer that produces the output tokens used by the SSL tasks.

3.1 Learning From All Tokens

We believe that during the learning process, extending the optimization of the classification token to all the output tokens of ViT could make the model more accurate and the training more efficient. To this end, inspired by [9, 21, 26], we define a list of tasks over the output tokens $\boldsymbol{z}_L^i$ with $i = 1 \dots N$ of the model: *spatial relations* (**SpRel**), to face the problem of recognizing the spatial relation class of pairs of image patches, *distances* (**Dist**) and *angles* (**Ang**), to learn a measure of distance and angle, respectively, between the input patches locations in the original image, and, finally, *absolute positions* (**AbsPos**) for recognizing the 2-d location of the input image patch. More formally, referring to $\boldsymbol{z}_i$ with $i = 1 \dots N$ as the i-th token of the L-th layer, given a pair of tokens $(\boldsymbol{z}_i, \boldsymbol{z}_j)$ where $\boldsymbol{z}_i$ is in position $\boldsymbol{p}_i = (x_i, y_i)$ and $\boldsymbol{z}_j$ is in position $\boldsymbol{p}_j = (x_j, y_j)$, we let the model to learn the spatial relation r_{ij} defined as:

$$r_{ij} = (s_x, s_y), \quad \text{where} \quad \begin{array}{l} s_x \in \{\mathrm{L}, \mathrm{C}, \mathrm{R}\} \\ s_y \in \{\mathrm{T}, \mathrm{C}, \mathrm{B}\} \end{array} \tag{5}$$

where the partial spatial relations {L, C, R, T, B} stand for {“left", “center", “right", “top", “bottom"}. In practice, we let the model to solve the relations classification task over the set of possible spatial relations $r \in \mathcal{R}$ with $|\mathcal{R}| = 9$.
We define the distance between the patch locations d_{ij} and the relative angle α_{ij} as:

$$d_{ij} = \|\boldsymbol{p}_i - \boldsymbol{p}_j\| \tag{6}$$

$$\alpha_{ij} = \cos^{-1}\left(\frac{\boldsymbol{p}_i \cdot \boldsymbol{p}_j}{\|\boldsymbol{p}_i\|\|\boldsymbol{p}_j\| + \varepsilon}\right) \tag{7}$$

To avoid numerical issues, in Eq. 7 we add a small positive constant ε in the denominator and shift $\boldsymbol{p}_i$ and $\boldsymbol{p}_j$ by a constant $\boldsymbol{s}$. Afterward, we normalize both d_{ij} and α_{ij} in $[-1, 1]$ and train the model to learn the normalized values. Finally, we define the absolute position of each token z_i simply as i and we let the model solve the classification task over the absolute positions $\{1, \dots, N\}$.

3.2 The Relational Vision Transformer

Our model is depicted in Figure 2. We aim at training all the output tokens processed by the transformer backbone for all the tasks defined in 3.1. Our architecture, dubbed RelViT, uses the standard ViT backbone equipped with specialized heads designed for solving the different tasks which take the output tokens $\boldsymbol{z} \in \mathbb{R}^{N \times D}$ of the transformer encoder as inputs. Specifically, *Relations*, *Dist* and *Angles* heads, which process pairs of tokens, are characterized by a MSA layer that produces the attention scores $\phi(\boldsymbol{z},\boldsymbol{z}) \in \mathbb{R}^{N \times N \times K}$ described as follows:

$$\phi(\boldsymbol{z},\boldsymbol{z}) = \left[\boldsymbol{a}_1, \ldots, \boldsymbol{a}_K\right], \quad \boldsymbol{a}_i = \frac{\boldsymbol{z}\boldsymbol{W}_i\boldsymbol{z}^T}{\sqrt{D}} \tag{8}$$

where $\boldsymbol{a}_i \in \mathbb{R}^{N \times N}$ is the attention scores of the single attention head, $\boldsymbol{W}_i \in \mathbb{R}^{D \times D}$ is a learnable projection matrix, K is the output dimension of the head, and D is the token dimension. Instead, the *AbsPos* head uses a fully connected (FC) layer that produces the logits $\phi(\boldsymbol{z}) \in \mathbb{R}^{N \times K}$ as follows:

$$\phi(\boldsymbol{z}) = \boldsymbol{z}\boldsymbol{W} + \boldsymbol{b} \tag{9}$$

where $\boldsymbol{W} \in \mathbb{R}^{D \times K}$ and $\boldsymbol{b} \in \mathbb{R}^{K}$ are respectively the weights and bias of the FC layer.
Given the corresponding logits, the SSL losses are computed as in the following equations:

$$\mathcal{L}_{\text{sp-rel}}(\boldsymbol{z}) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \text{CE}\left[\phi_{\text{sp-rel}}(\boldsymbol{z}_i,\boldsymbol{z}_j), r_{ij}\right] \tag{10}$$

$$\mathcal{L}_{\text{abs-pos}}(\boldsymbol{z}) = \frac{1}{N}\sum_{i=1}^{N} \text{CE}\left[\phi_{\text{abs-pos}}(\boldsymbol{z}_i), i\right] \tag{11}$$

$$\mathcal{L}_{\text{dist}}(\boldsymbol{z}) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \left\|\phi_{\text{dist}}(\boldsymbol{z}_i,\boldsymbol{z}_j) - d_{ij}\right\|^2 \tag{12}$$

$$\mathcal{L}_{\text{angle}}(\boldsymbol{z}) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} \left\|\phi_{\text{angle}}(\boldsymbol{z}_i,\boldsymbol{z}_j) - \alpha_{ij}\right\|^2 \tag{13}$$

where $\text{CE}(\cdot)$ is the cross-entropy with logits loss function. Finally, during training, we minimize the sum of all the losses of the tasks we want to solve. It is worth noticing that with this formulation, for each training step, we optimize all the combinations of pairs of patches at the same time in parallel.

3.2.1 Mega-Patches

Our SSL heads provide the spatial annotations thanks to a self-attention mechanism, where any output token of the ViT encoder interacts with itself and all the others in order to acquire knowledge about the spatial relations among pairs of patches. Consequently, we believe that the increase of the number of tokens, obtained from the non-overlapping patches on which the input image is divided, could doubly impact on the global process. First of all, there is the well-know computational effort which has a quadratic complexity with respect to the number of tokens. Secondly, a small patch size could reduce the improvement obtained thanks to our SSL approach since the contextual information in a patch is reduced.

Thus, we propose to divide the input image into $M \times M$ mega-patches which are non-overlapping rectangles whose height and width are integer multiples of the patch size P. More specifically, given an image $\boldsymbol{x} \in \mathbb{R}^{C \times H \times W}$, and the number of mega-patch per side M, we uniformly sample $M-1$ random indices that cut the image horizontally and vertically, producing rectangular mega-patches of different dimensions that are resized to (P,P) before feeding them to our RelViT model. A qualitative difference between patches and mega-patches of an image is illustrated in Figure 3. As shown, the mega-patches are groups of adjacent patches and we believe that their use could be beneficial for representing the image content, resulting in a better input for our RelViT model during the pre-training on the SSL tasks.

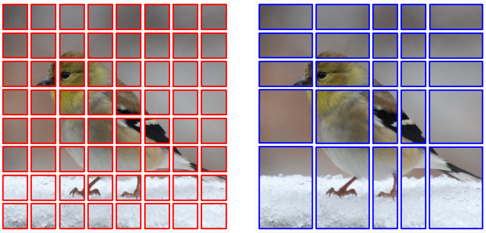

Figure 3: Comparison between using standard patches and our proposed mega-patches. On the left, the image is divided into 64 patches, on the right it is randomly divided into 25 mega-patches using $M = 5$.

4 Experiments

We aim at demonstrating the effectiveness of learning Visual Transformers via a number of self-supervised tasks whose benefits are shown on both ViT and two of its variants: Swin [22] and T2T-ViT [38].

4.1 Datasets and Implementation Details

Model	Img Res.	Blocks	Heads	Dim	#Tokens	#Params
ViT-S/4	(32, 32)	12	6	384	65	21.3M
ViT-S/8	(64, 64)	12	6	384	65	21.4M
ViT-S/32	(224, 224)	12	6	384	50	22.5M
ViT-S/16	(224,224)	12	6	384	197	22.5M
ViT-S/14	(224,224)	12	6	384	257	22.5M

Table 1: The used ViTs model configurations; *Img Res.* is the image resolution, *Blocks* and *Heads* are the number of layers in the backbone and of heads in the multi-head self attention layer, *Dim* is the dimension of the token in the transformer encoder, *#Tokens* is the number of patches coming from the image and *#Params* is the number of learnable parameters of the model.

Datasets and RelViT Configurations. We conducted experiments on standard image benchmarks for classification focusing on relatively small datasets providing a simple and effective way of boosting ViTs performance. In particular, we tested RelViT on CIFAR-10 [19], SVHN [24], CIFAR-100 [19], Flower-102 [25], Tiny-Imagenet [20] and ImageNet-100 (a subset of 100 labels from ImageNet), which have a number of train samples from 1k for Flower-102 to 130k for ImageNet-100. We used ViT-S backbone configuration [3] (model details in Table 1), while the experiments on ViT variants use Swin-T and T2T-ViT-14 backbones with the same model configurations as in [21]. Since our method is built upon the backbone and, once the model is trained with our SSL tasks, the additional parameters related to the heads are negligible with respect to the backbone size (+2% on the spatial relation, distance, and angle tasks and +0.1% on the absolute position task), the computational complexity added by the RelViT is negligible with respect to the visual backbone computational cost.

Training Details. For all the experiments, we use the same configurations of the training parameters, except if differently mentioned. All the experiments, obtained pre-training on our SSL tasks, fine-tuning on classification or training jointly on classification and SSL tasks, are trained for 100 epochs. Following [3], we use the AdamW [23] optimizer with a cosine learning rate scheduling, linear warm-up of 10 epochs, and $lr = 0.0005$ as the maximum value for the learning rate. We found that RelViT does not need regularization, so we avoid the use of dropout, drop-path, weight decay, and label smoothing. The batch size is set to 256 for ViT backbone and 64 for its variants, and we use a single GPU for performing each experiment. The data augmentation for the classification problem follows the standard practice of resized crops and random horizontal flipping. Instead, when using ViT backbone and SSL tasks, each image patch is randomly resized-cropped, and following [5] a color shift with color-jittering is applied as well as random gray-scale perturbation.

4.2 Experimental Results

We investigated our RelViT model under two different scenarios: pre-training the model from scratch for our SSL tasks and subsequently fine-tuning on classification (from now on referred as to *pre-training* and *fine-tuning*, respectively) and training the model from scratch jointly on classification and SSL tasks (*downstream-only*). We used the spatial relations and the absolute positions tasks in both pre-training and downstream-only and we removed the positional encoding in the *pre-training* to avoid the shortcut learning problem, while they are used as aid whenever we want to solve the classification task, thus during both *fine-tuning* and *downstream-only* (more details in paragraph 4.3). Finally, we trained all the experiments over 5 different runs in order to prove the stability of our method.

	Backbone	**Supervised**	**RelViT**	**Improv.**
CIFAR-10	ViT-S/4	86.09 ±0.46	**90.23** ±0.09	**+4.14** ↑
SVHN	ViT-S/4	96.01 ±0.07	**97.14** ±0.03	**+1.13** ↑
CIFAR-100	ViT-S/4	59.19 ±0.84	**64.99** ±0.46	**+5.85** ↑
Flower-102	ViT-S/32	42.08 ±0.29	**45.78** ±0.75	**+3.70** ↑
TinyImagenet	ViT-S/8	43.19 ±0.78	**51.98** ±0.20	**+8.79** ↑
Imagenet100	ViT-S/32	58.04 ±0.91	**66.46** ±0.45	**+8.42** ↑

Table 2: Comparison between our RelVit model and the supervised ViT baselines on several small datasets. RelVit is pre-trained on our SSL tasks and subsequently finetuned for classification, whereas the baselines are trained on classification.

Table 2 shows RelViT accuracy on classification task after the *fine-tuning*. Our model is compared with the standard supervised ViT, referred as *Supervised*, whose results are obtained by performing both a supervised pre-training and a fine-tuning only on classification task equipping both with PEs and using the same settings described in paragraph 4.1. Our approach leads to higher accuracy than the supervised one in all datasets, gaining +8.79% on Tiny-Imagenet, +5.85% on CIFAR-100 and +4.14% on CIFAR-10.

Backbone	**Method**	**CIFAR-10**	**SVHN**	**CIFAR-100**	**Flower-102**
		ViT-S/4	ViT-S/4	ViT-S/4	ViT-S/32
ViT [11]	Supervised	85.86 ±0.37	95.94 ±0.05	59.51 ±0.99	41.74 ±0.97
	RelViT (ours)	**89.27** ±0.23	**96.41** ±0.39	**61.92** ±0.19	**45.78** ±0.75
	(**Improv.**)	(**+3.41** ↑)	(**+0.47** ↑)	(**+2.41** ↑)	(**+4.04** ↑)
Swin [22]	Supervised [21]	59.47	71.6	53.28	34.51
	Swin+$\mathcal{L}_{drloc}$ [21]	83.89	94.23	66.23	39.37
	RelSwin (ours)	**92.33** ±0.22	**96.64** ±0.07	**68.69** ±0.91	**58.47** ±0.71
	(**Improv.**)	(**+8.44** ↑)	(**+2.41** ↑)	(**+2.46** ↑)	(**+19.10** ↑)
T2T-ViT [38]	Supervised [21]	84.19	95.36	65.16	31.73
	T2T-ViT+$\mathcal{L}_{drloc}$ [21]	87.56	96.49	**68.03**	34.35
	RelT2T-ViT (ours)	**90.82** ±0.16	**96.52** ±0.05	66.27 ±0.88	**50.53** ±1.45
	(**Improv.**)	(**+3.26** ↑)	(**+0.03** ↑)	(**-1.76** ↓)	(**+16.18** ↑)

Table 3: Comparison of our method with other works on multiple backbones. The results are obtained training the model from scratch jointly on our SSL tasks and classification.

The first row in Table 3 shows RelViT results on classification under the *downstream-only* scenario using ViT as backbone and standard ViT as supervised baseline. Also this configuration improves the results on all the datasets up to +4.04% on Flower102, +3.41% on CIFAR-10, and +2.41% on CIFAR-100. The improvements obtained in both scenarios demonstrate the effectiveness of our self-supervised tasks for image classification and the stability of our method's accuracy suggests that RelViT is not affected by large fluctuations.

Finally, we investigated how our SSL tasks generalize across different backbones. We chose Swin [22] and T2T-ViT [38] to compare our method with [21] that shares a similar setting. As reported in Table 3, we outperform the supervised baselines and also results reported in [21] up to +19.10% and +16.18% on Flower-102, with Swin and T2T-ViT backbones, respectively.

4.3 Ablations

SSL Multi-Task Ablation. We defined multiple SSL tasks over the image tokens and evaluated each combination of tasks in order to highlight both the importance of each sub-problem and the relations among them. Table 4 displays the impact of each combination of SSL tasks reporting the performance on the SSL tasks investigated during the *pre-training* (columns 2nd-6th) and the accuracy on the classification after the *fine-tuning* (last column). It is worth noticing that solving, individually, each SSL task leads to similar fine-tuning accuracy, improving the performance w.r.t the supervised baseline of +3.00% on average. Among SSL tasks, the *SpRel* one is the most performing since if we rank the combinations by the fine-tuning accuracy of the fine-tuned model, it is present in the top-3 entries, suggesting its importance for the fine-tuning classification task. Moreover, we find that the optimal combination of tasks is obtained by learning the *SpRel* and the *AbsPos* of the patches.

Positional Embeddings and Patches Permutations. We investigated the shortcut learning problem in RelViT using the standard learnable 1D positional embeddings (PEs) during the self-supervised pre-training.Moreover, to avoid trivial solutions and break the dependency between PEs and SSL labels, we investigated the shuffling of the input patches before adding the PEs, permuting the rows and columns of the label matrix with the same permutation used for shuffling the input tokens. Since we aim at focusing on the role of the PEs and the patch shuffling, for simplicity we used only the *SpRel* learning on the pre-training.

As shown in Figure 4, if the model uses the PEs without shuffling during the pre-training, it solves the task with 100% of *SpRel* accuracy but the corresponding fine-tuned model reaches a classification accuracy comparable to a randomly initialized network (red dashed line in Fig. 4). Even though the shuffling removes the trivial solution and improves the classification accuracy, we removed PEs from RelViT during *pre-training* on our SSL tasks since, in this way, we obtained the best results. Whereupon, the shuffling is optional due to the permutation equivariance property of the model. By contrast, it is important to highlight that we used PEs during both *fine-tuning* and *downstream-only* as well as to compute all the supervised baselines.

Method	**AbsPos (Acc ↑)**	**Ang (MSE ↓)**	**Dist (MSE ↓)**	**SpRel (Acc ↑)**	**Finetuning (Acc ↑)**
Supervised	✗	✗	✗	✗	85.38
RelViT	✗	✗	✗	48.53	89.02
RelViT	✗	✗	10.54	✗	89.27
RelViT	✗	13.08	✗	✗	88.75
RelViT	15.24	✗	✗	✗	89.39
RelViT	✗	✗	10.29	48.91	89.24
RelViT	✗	12.55	✗	48.66	90.12
RelViT	15.73	✗	✗	48.53	**90.13**
RelViT	✗	12.58	10.32	✗	89.34
RelViT	16.03	✗	10.49	✗	89.75
RelViT	15.51	12.82	✗	✗	89.90
RelViT	✗	12.49	10.28	48.98	88.81
RelViT	15.90	✗	10.39	48.67	89.80
RelViT	16.03	12.52	✗	48.80	89.73
RelViT	16.06	12.65	10.50	✗	89.67
RelViT	16.07	12.54	10.39	48.74	90.06

Table 4: Results using our SSL tasks on CIFAR-10 with ViT-S/4 backbone. The columns from the 2nd to the 6th report the performance on the SSL tasks whereas the last column is the accuracy of the fine-tuned models.

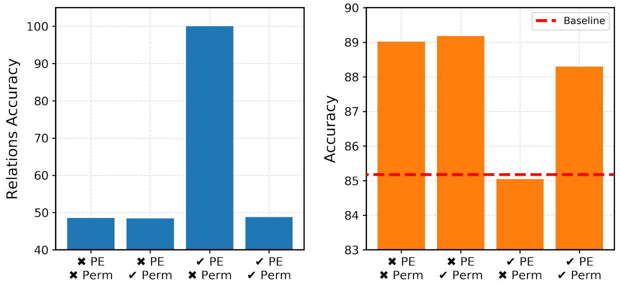

Figure 4: The role of the positional embedding (PE) and random permutations (Perm) of the input patches in RelViT during the SSL Pre-Training (left) and the Fine-Tuning (right).

Mega-Patches Ablation. We investigated our RelViT model using the mega-patches approach. To this end, the models are pre-trained from scratch using mega-patches for the spatial relations and the absolute positions tasks, and subsequently they are fine-tuned on classification using the standard patches. Table 5 shows RelViT accuracy after a supervised fine-tuning on ImageNet-100 and Flower-102 using three different backbones: ViT-S/32, ViT-S/16 and ViT-S/14 which differ in the patch size. *Supervised* and *No-Mega-Patches* are reported as references. *Supervised* is obtained training only on classification, whereas *No-Mega-Patches* is obtained pre-training for the spatial relations and the absolute positions without mega-patches and fine-tuning on classification. We tested several configurations by varying the parameter $M \in [3,10]$. We observe that our RelViT model always outperforms the supervised baselines regardless of the value of the parameter M underlining the effectiveness of our RelViT model. Moreover, comparing the results obtained with and without the use of the mega-patches, as expected, the mega-patches approach improves the accuracy levels as much as the patch size of the backbone is reduced suggesting that our SSL taks need more contextual information in each patch to express their potential.

	ImageNet-100			**Flower-102**		
M	ViT-S/32	ViT-S/16	ViT-S/14	ViT-S/32	ViT-S/16	ViT-S/14
Supervised	58.04	69.42	71.14	41.67	46.37	45.39
No-Mega-Patches	67.02	74.74	75.76	**45.49**	49.31	50.98
10	-	75.30	76.10	-	48.43	49.61
8	-	74.36	76.42	-	46.76	**52.25**
6	**67.26**	**75.36**	**76.60**	45.39	49.61	51.67
5	67.14	75.26	75.58	44.22	48.63	51.37
4	66.02	74.26	74.24	44.31	49.31	50.39
3	64.38	73.16	74.12	43.43	**50.78**	50.10

Table 5: Comparison between our RelViT model using our mega-patches approach and the supervised ViT baselines. The results are obtained pre-training the model from scratch for on SSL tasks using the mega-patches and then fine-tuning on the standard patches. M is the number of mega-patches per side (height or width) used during the SSL pre-training.

5 Conclusion

In this work, we have proposed and investigated several self-supervised tasks defined over image patches that are naturally used by ViT. Our proposed methods are straightforward to implement and effective as the RelViT shows significant improvement on all the tested datasets and in both scenarios: pre-training on SSL tasks and then fine-tuning and training jointly on SSL and SL tasks. All the reported experiments focused on relatively small datasets, showing a simple way of making ViTs stronger even on the small dataset regime.

Acknowledgements. This work was supported by the PRIN-17 PREVUE project from the Italian MUR (CUP E94I19000650001). We also acknowledge the HPC resources of UniPD – DM and CAPRI clusters – and NVIDIA for their donation of GPUs used in this research.

References

[1] Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *ArXiv*, abs/1607.06450, 2016.

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Herve Jegou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey E. Hinton. A simple framework for contrastive learning of visual representations. In *Proc. of the International Conference on Machine Learning (ICML)*, 2020.

[5] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[6] Xinlei Chen, Haoqi Fan, Ross B. Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *ArXiv*, abs/2003.04297, 2020.

[7] Zhengsu Chen, Lingxi Xie, Jianwei Niu, Xuefeng Liu, Longhui Wei, and Qi Tian. Visformer: The vision-friendly transformer. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of the North American Chapter of the Association for Comput. Linguistics (NAACL)*, 2019.

[9] Carl Doersch, Abhinav Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2015.

[10] Alexey Dosovitskiy, Jost Tobias Springenberg, Martin Riedmiller, and Thomas Brox. Discriminative unsupervised feature learning with convolutional neural networks. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2014.

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2021.

[12] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[13] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2018.

[14] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross B. Girshick. Momentum contrast for unsupervised visual representation learning. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[16] Olivier Henaff, Aravind Srinivas, Jeffrey De Fauw, Ali Razavi, Carl Doersch, S.M. Ali Eslami, and Aaron van den Oord. Data-efficient image recognition with contrastive predictive coding. In *Proc. of the International Conference on Machine Learning (ICML)*, 2020.

[17] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[18] R Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Phil Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2018.

[19] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[20] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. 2015. URL `http://cs231n.stanford.edu/tiny-imagenet-200.zip`.

[21] Yahui Liu, Enver Sangineto, Wei Bi, Nicu Sebe, Bruno Lepri, and Marco De Nadai. Efficient training of visual transformers with small-size datasets. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[22] 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 *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[23] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proc. of the International Conference on Learning Representations (ICLR)*, 2019.

[24] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[25] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. *Proc. of Indian Conference on Computer Vision, Graphics & Image Processing (ICVGIP)*, 2008.

[26] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2016.

[27] Mehdi Noroozi, Hamed Pirsiavash, and Paolo Favaro. Representation learning by learning to count. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2017.

[28] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[29] Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[30] Aäron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *ArXiv*, abs/1807.03748, 2018.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

[32] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proc. of the International Conference on Machine Learning (ICML)*, 2008.

[33] 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 *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[34] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. CvT: Introducing convolutions to vision transformers. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[35] Zhirong Wu, Yuanjun Xiong, Stella X. Yu, and Dahua Lin. Unsupervised feature learning via non-parametric instance discrimination. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[36] Tete Xiao, Mannat Singh, Eric Mintun, Trevor Darrell, Piotr Dollár, and Ross Girshick. Early convolutions help transformers see better. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

[37] Mang Ye, Xu Zhang, PongChi Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[38] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proc. of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021.

[39] Richard Zhang, Phillip Isola, and Alexei A. Efros. Colorful image colorization. In *Proc. of the European Conference on Computer Vision (ECCV)*, 2016.

[40] Richard Zhang, Phillip Isola, and Alexei A. Efros. Split-brain autoencoders: Unsupervised learning by cross-channel prediction. In *Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.