Machine Learning (2024) 113:6945–6975
https://doi.org/10.1007/s10994-024-06570-7

Spatial entropy as an inductive bias for vision transformers

Elia Peruzzo[1] · Enver Sangineto[2] · Yahui Liu[1,3] · Marco De Nadai[3] · Wei Bi[4] · Bruno Lepri[3] · Nicu Sebe[1]

Received: 30 May 2023 / Revised: 21 February 2024 / Accepted: 15 May 2024 /
Published online: 17 July 2024

Abstract
Recent work on Vision Transformers (VTs) showed that introducing a local inductive bias in the VT *architecture* helps reducing the number of samples necessary for training. However, the architecture modifications lead to a loss of generality of the Transformer backbone, partially contradicting the push towards the development of uniform architectures, shared, e.g., by both the Computer Vision and the Natural Language Processing areas. In this work, we propose a different and complementary direction, in which a local bias is introduced using *an auxiliary self-supervised task*, performed jointly with standard supervised training. Specifically, we exploit the observation that the attention maps of VTs, when trained with self-supervision, can contain a semantic segmentation structure which does not spontaneously emerge when training is supervised. Thus, we *explicitly* encourage the emergence of this spatial clustering as a form of training regularization. In more detail, we exploit the assumption that, in a given image, objects usually correspond to few connected regions, and we propose a spatial formulation of the information entropy to quantify this *object-based inductive bias*. By minimizing the proposed spatial entropy, we include an additional self-supervised signal during training. Using extensive experiments, we show that the proposed regularization leads to equivalent or better results than other VT proposals which include a local bias by changing the basic Transformer architecture, and it can drastically boost the VT final accuracy when using small-medium training sets. The code is available at https://github.com/helia95/SAR.

Keywords Self-supervision · Vision transformers · Transformers

Editor: Nathalie Japkowicz.

Enver Sangineto and Yahui Liu have contributed equally to this work.

✉ Marco De Nadai
work@marcodena.it

[1] University of Trento, Trento, Italy

[2] University of Modena and Reggio Emilia, Modena, Italy

[3] Bruno Kessler Foundation, Trento, Italy

[4] Tencent AI Lab, Shenzhen, China

1 Introduction

Vision Transformers (VTs) are increasingly emerging as the dominant Computer Vision architecture, alternative to standard Convolutional Neural Networks (CNNs). VTs are inspired by the Transformer network (Vaswani et al., 2017), which is the *de facto* standard in Natural Language Processing (NLP) (Kenton & Toutanova, 2019; Radford & Narasimhan, 2018) and it is based on multi-head attention layers transforming the input *tokens* (e.g., language words) into a set of final embedding tokens. Dosovitskiy et al. (2021) proposed an analogous processing paradigm, called ViT.[1], where word tokens are replaced by image patches, and self-attention layers are used to model global pairwise dependencies over all the input tokens. As a consequence, differently from CNNs, where the convolutional kernels have a spatially limited receptive field, ViT has a dynamic receptive field, which is given by its attention maps (Naseer et al., 2021). However, ViT heavily relies on huge training datasets (e.g., JFT-300 M (Dosovitskiy et al., 2021), a proprietary dataset of 303 million images), and underperforms CNNs when trained on ImageNet-1K ($\sim$ 1.3 million images) (Russakovsky et al., 2015) or using smaller datasets (Dosovitskiy et al., 2021; Raghu et al., 2021). The main reason why Transformers are more data hungry than CNNs is that they lack some inductive biases embedded into the CNN architecture, such as translation equivariance and locality, obtained using local filters, parameter sharing and spatial pooling. Hence, Transformers do not generalize well when trained on insufficient amounts of data (Dosovitskiy et al., 2021). To alleviate this problem and mitigate the need for a huge quantity of training data, a recent line of research is exploring the possibility of reintroducing typical CNN mechanisms in VTs (Yuan et al., 2021; Liu et al., 2021; Wu et al., 2021; Yuan et al., 2021; Xu et al., 2021; Li et al., 2021; Hudson & Zitnick, 2021; Hassani et al., 2023). The main idea behind these "hybrid" VTs is that convolutional layers, mixed with the VT self-attention layers, help to embed a *local inductive bias* in the VT *architecture*, i.e., to encourage the network to focus on local properties of the image domain. However, the disadvantage of this paradigm is that it requires drastic architectural changes to the original ViT, where the latter has now become a *de facto* standard in different vision and vision-language tasks (Sect. 2). Moreover, as emphasised by Li et al. (2022), one of the main advantages of CNNs is the large independence of the pre-training from the downstream tasks, which allows to use a uniform backbone, pre-trained only once for different vision tasks (e.g., classification, object detection, etc.). Conversely, the adoption of specific VT architectures breaks this independence and makes it difficult to use the same pre-trained backbone for different downstream tasks (Li et al., 2022).

In this paper, we follow an orthogonal (and relatively simpler) direction: rather than changing the ViT architecture, we propose to include a local inductive bias using an additional *pretext task* during training, which implicitly "teaches" the network the connectedness property of objects in an image. Specifically, we maximize the probability of producing attention maps whose highest values are clustered in few local regions (of *variable* size), based on the idea that, most of the time, an object is represented by one or very few spatially connected regions in the input image. This pretext task exploits a locality principle, characteristic of the natural images, and extracts additional (self-supervised) information from images without the need of architectural changes.

[1] In this paper, we use VT to refer to generic Vision Transformer archiectures, and ViT to refer to the specific architecture proposed in (Dosovitskiy et al., 2021).

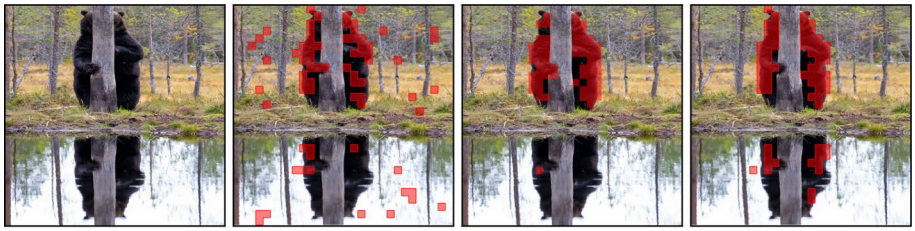

Fig. 1 ViT attention maps obtained using the `[CLS]` token query and thresholded to keep 60% of the mass following (Caron et al., 2021). **a** Original image. **b** Standard supervised learning. **c** DINO. **d** Training using SAR. In (**a**), the two largest connected components correspond to the main object (a bear, occluded by a tree), while the third and the fourth largest connected components correspond to the specular reflection of the bear into the river

Our work is inspired by the findings presented in (Caron et al., 2021; Bao et al., 2022; Naseer et al., 2021), in which the authors show that VTs, trained using self-supervision (Caron et al., 2021; Bao et al., 2022) or shape-distillation (Naseer et al., 2021), can spontaneously develop attention maps with a semantic segmentation structure. For instance, Caron et al. (2021) show that the last-layer attention maps of ViT, when this is trained with their self-supervised DINO algorithm, can be thresholded and used to segment the most important foreground objects of the input image without any pixel-level annotation during training (see Fig. 1c). Similar findings are shown in (Bao et al., 2022; Naseer et al., 2021). Interestingly, however, Caron et al. (2021) show that the same ViT architectures, when trained with supervised methods, produce much more spatially disordered attention maps (Fig. 1b). This is confirmed by Naseer et al. (2021), who observed that the attention maps of ViT, trained with supervised protocols, have a widely spread structure over the whole image. The reason why "blob"-like attention maps *spontaneously* emerge when VTs are trained with some algorithms but not with others, is still unclear. However, in this paper we build on top of these findings and we propose a *spatial entropy* loss function which *explicitly* encourages the emergence of locally structured attention maps (Fig. 1d), independently of the main algorithm used for training. Note that our goal is *not* to use the so obtained attention maps to extract segmentation structure from images or for other post-processing steps. Instead, we use the proposed spatial entropy loss to introduce an *object-based local inductive bias* in VTs. Since real life objects usually correspond to one or very few connected image regions, then also the corresponding attention maps of a VT head should focus most of their highest values on spatially clustered regions. The possible discrepancy between this inductive bias (an object in a given image is a mostly connected structure) and the actual spatial entropy measured in each VT head, provides a *self-supervised* signal which alleviates the need for huge supervised training datasets, without changing the ViT architecture.

The second contribution of this paper is based on the empirical results presented by Raghu et al. (2021), who showed that VTs are more influenced by the skip connections than CNNs and, specifically, that in the last blocks of ViT, the *patch tokens* (see Sect. 3) representations are mostly influenced by the skip connection path. This means that, in the last blocks of ViT, the self-attention layers have a relatively small influence on the final token embeddings. Since our spatial entropy is measured on the last-block attention maps, we propose to remove the skip connections in the last layer (only). We empirically show

that this minor architectural change is beneficial for ViT, both when used jointly with our spatial entropy loss, and when used with a standard training procedure.

Our regularization method, which we call SAR (Spatial Attention-based Regularization), can be easily plugged into existing VTs (including hybrid VTs) without drastic architectural changes of their architecture. SAR can be applied to different scenarios, jointly with a main-task loss function. For instance, when used in a supervised classification task, the main loss is the (standard) cross entropy, used jointly with our spatial entropy loss. We empirically show that SAR is sample-efficient: it can significantly boost the accuracy of ViT in different downstream tasks such as classification, object detection or segmentation, and it is particularly useful when training with small-medium datasets.

Our experiments show also that SAR is beneficial when plugged on hybrid VT architectures, especially when the training data are scarce.

In summary, our main contributions are the following: (1) We propose to embed a local inductive bias in ViT using spatial entropy as an alternative to re-introducing convolutional mechanisms in the VT architecture. (2) We propose to remove the last-block skip connections, empirically showing that this is beneficial for the patch token representations. (3) Using extensive experiments, we show that SAR improves the accuracy of different VT architectures, and it is particularly helpful when supervised training data are scarce.

2 Related work

Vision Transformers One of the very first fully-Transformer architectures for Computer Vision is iGPT (Chen et al., 2020), in which each image pixel is represented as a token. However, due to the quadratic computational complexity of Transformer networks (Vaswani et al., 2017), iGPT can only operate with very small-resolution images. This problem has been largely alleviated by ViT (Dosovitskiy et al., 2021), where the input tokens are $p \times p$ image patches (Sect. 3). The success of ViT has inspired several similar Vision Transformer (VT) architectures in different application domains, such as image classification (Dosovitskiy et al., 2021; Touvron et al., 2021; Yuan et al., 2021; Liu et al., 2021; Wu et al., 2021; Yuan et al., 2021; Li et al., 2021; Xu et al., 2021; d'Ascoli et al., 2021), object detection (Li et al., 2022; Carion et al., 2020; Zhu et al., 2021; Dai et al., 2021), segmentation (Strudel et al., 2021; Rao et al., 2022), human pose estimation (Zheng et al., 2021), object tracking (Meinhardt et al., 2022), video processing (Neimark et al., 2021; Li et al., 2022), image generation (Jiang et al., 2021; Hudson & Zitnick, 2021; Ramesh et al., 2022; Chang et al., 2022), point cloud processing (Guo et al., 2021; Zhao et al., 2021), vision-language foundation models (Bao et al., 2022; Alayrac et al., 2022; Wang et al., 2022) and many others. However, the lack of the typical CNN local inductive biases makes VTs to need more data for training (Dosovitskiy et al., 2021; Raghu et al., 2021). For this reason, many recent works are addressing this problem by proposing hybrid architectures, which reintroduce typical convolutional mechanisms into the VT design (Yuan et al., 2021; Liu et al., 2021; Wu et al., 2021; Yuan et al., 2021; Xu et al., 2021; Li et al., 2021; d'Ascoli et al., 2021; Hudson & Zitnick, 2021; Li et al., 2022; Hassani et al., 2023). In contrast, we propose a different and simpler solution, in which, rather than changing the VT architecture, we introduce an object-based local inductive bias (Sect. 1) by means of a pretext task based on the spatial entropy minimization.

Note that our proposal is different from object-centric learning (Locatello et al., 2020; Goyal et al., 2020; Didolkar et al., 2021; Engelcke et al., 2021; Sajjadi et al.,

2022; Herzig et al., 2022; Kang et al., 2022), where the goal is to use discrete objects (usually obtained using a pre-trained object detector or a segmentation approach) for object-based reasoning and modular/causal inference. In fact, although the attention maps produced by SAR can potentially be thresholded and used as discrete objects, our goal is not to segment the image patches or to use the patch clusters for further processing steps, but to exploit the clustering process as an additional self-supervised loss function which helps to reduce the need of labeled training samples (Sect. 1).

Self-supervised learning Most of the self-supervised approaches with still images and ResNet backbones (He et al., 2016) impose a semantic consistency between different views of the same image, where the views are obtained with data-augmentation techniques. This work can be roughly grouped in contrastive learning (van den Oord et al., 2018; Hjelm et al., 2019; Chen et al., 2020; He et al., 2020; Tian et al., 2020; Wang & Isola, 2020; Dwibedi et al., 2021), clustering methods (Bautista et al., 2016; Zhuang et al., 2019; Ji et al., 2019; Caron et al., 2018; Asano et al., 2020; Gansbeke et al., 2020; Caron et al., 2020, 2021), asymmetric networks (Grill et al., 2020; Chen & He, 2021) and feature-decorrelation methods (Ermolov et al., 2021; Zbontar et al., 2021; Bardes et al., 2022; Hua et al., 2021).

Recently, different works use VTs for self-supervised learning. For instance, Chen et al. (2021) have empirically tested different representatives of the above categories using VTs, and they also proposed MoCo-v3, a contrastive approach based on MoCo (He et al., 2020) but without the queue of the past-samples. DINO (Caron et al., 2021) is an on-line clustering method which is one of the current state-of-the-art self-supervised approaches using VTs. BEiT (Bao et al., 2022) adopts the typical "masked-word" NLP pretext task (Kenton & Toutanova, 2019), but it needs to pre-extract a vocabulary of visual words using the discrete VAE pre-trained in (Ramesh et al., 2021). Other recent works which use a "masked-patch" pretext task are: (He et al., 2022; Xie et al., 2022; Wei et al., 2022; Dong et al., 2023; Hua et al., 2023; Chen et al., 2024; Bachmann et al., 2022; El-Nouby et al., 2021; Zhou et al., 2022; Kakogeorgiou et al., 2022).

Yun (2022) use the assumption that adjacent patches usually belong to the same object in order to collect positive patches for a contrastive learning approach. Our inductive bias shares a similar intuitive idea but, rather than a contrastive method where positive pairs are compared with negative ones, our self-supervised loss is based on the proposed spatial entropy which groups together patches even when they are not pairwise adjacent. Generally speaking, in this paper, we do not propose a fully self-supervised algorithm, but we rather use self-supervision (we extract information from samples without additional manual annotation) to speed-up the convergence in a supervised scenario and decrease the quantity of annotated information needed for training. In the Appendix, we also show that SAR can be plugged on top of both MoCo-v3 and DINO, boosting the accuracy of both of them.

Similarly to this paper, Liu et al. (2021) propose a VT regularization approach based on predicting the geometric distance between patch tokens. In contrast, we use the highest-value connected regions in the VT attention maps to extract additional unsupervised information from images and the two regularization methods can potentially be used jointly. Li et al. (2020) compute the gradients of a ResNet with respect to the image pixels to get an attention (saliency) map. This map is thresholded and used to mask-out the most salient pixels. Minimizing the classification loss on this masked image encourages the attention on the non-masked image to include most of the useful information. Our approach is radically different and much simpler, because we do not need to manually set the thresholding value and we require only one forward and one backward pass per image.

Spatial entropy There are many definitions of spatial entropy (Razlighi & Kehtarnavaz, 2009; Altieri et al., 2018). For instance, Batty (1974) normalizes the probability of an event occurring in a given zone by the area of that zone, this way accounting for unequal space partitions. In (Shah et al., 2020), the Shannon entropy, defined over an histogram of a gray-scale image, is extended to include different hyperspectral bands. Ceci et al. (2019) use a Parzen-Window Density Estimation to include spatial information in their spatial entropy loss which accounts for the vicinity of different spatially-located sensors. Li et al. (2020) propose an entropy measure to evaluate the spatiotemporal regularity on tensor data which is based on the conditional probability of the similarities between tensor sub-blocks. In (Tupin et al., 2000), spatial entropy is defined over a Markov Random Field describing the image content, but its computation is very expensive (Razlighi & Kehtarnavaz, 2009). In contrast, our spatial entropy loss can be efficiently computed and it is differentiable, thus it can be easily used as an auxiliary regularization task in existing VTs.

3 Background

Given an input image I, ViT (Dosovitskiy et al., 2021) splits I in a grid of $K \times K$ non-overlapping patches, and each patch is linearly projected into a (learned) input embedding space. The input of ViT is this set of $n = K^2$ *patch tokens*, jointly with a special token, called `[CLS]` token, which is used to represent the whole image. Following a standard Transformer network (Vaswani et al., 2017), ViT transforms these $n + 1$ tokens in corresponding final $n + 1$ token embeddings using a sequence of L Transformer blocks. Each block is composed of LayerNorm (LN), Multiheaded Self Attention (MSA) and MLP layers, plus skip connections. Specifically, if the token embedding sequence in the $(l-1)$-th layer is $\boldsymbol{z}^{l-1} = [\boldsymbol{z}_{CLS}; \boldsymbol{z}_1; ... \boldsymbol{z}_n]$, then:

$$\boldsymbol{z}' = \text{MSA}\,(\,\text{LN}\,(\boldsymbol{z}^{l-1})) + \boldsymbol{z}^{l-1}, l = 1, \ldots, L \tag{1}$$

$$\boldsymbol{z}^{l} = \text{MLP}\,(\,\text{LN}\,(\boldsymbol{z}')) + \boldsymbol{z}', l = 1, \ldots, L \tag{2}$$

where the addition (+) denotes a skip (or "identity") connection, which is used both in the MSA (Eq. 1) and in the MLP (Eq. 2) layer. The MSA layer is composed of H different heads, and, in the h-th head ($1 \leq h \leq H$), each token embedding $\boldsymbol{z}_i \in \mathbb{R}^d$ is projected into a query ($\boldsymbol{q}_i^h$), a key ($\boldsymbol{k}_i^h$) and a value ($\boldsymbol{v}_i^h$). Given query (Q^h), key (K^h) and value (V^h) matrices containing the corresponding elements, the h-th self-attention matrix (A^h) is given by:

$$A^h = \text{softmax}\left(\frac{Q^h (K^h)^T}{\sqrt{d}}\right). \tag{3}$$

Using A^h, each head outputs a weighted sum of the values in V^h. The final MSA layer output is obtained by concatenating all the head outputs and then projecting each token embedding into a d-dimensional space. Finally, the last-layer (L) class token embedding $\boldsymbol{z}_{CLS}^L$ is fed to an MLP head, which computes a posterior distribution over the set of the target classes and the whole network is trained using a standard cross-entropy loss ($\mathcal{L}_{ce}$). Some hybrid VTs (see Sect. 2) such as CvT (Wu et al., 2021) and PVT (Wang et al., 2021), progressively subsample the number of patch tokens, leading to a final $k \times k$ patch token grid ($k \leq K$). In the rest of this paper, we generally refer to a spatially arranged grid of final patch token embeddings with a $k \times k$ resolution.

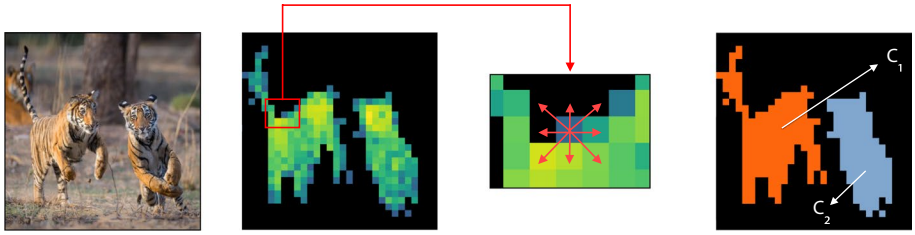

Fig. 2 A schematic illustration of the spatial entropy. **a** The original image. **b** The thresholded similarity map B^h (zero values shown in black) for a specific head. **c** The 8-connectivity relation used to group non-zero elements in B^h. (d) The resulting two connected components (C_1 and C_2), each component shown with a different colour. In this case, $C^h = \{C_1, C_2\}$

4 Method

Generally speaking, an object usually corresponds to one or very few connected regions of a given image. For instance, the bear in Fig. 1, despite being occluded by a tree, occupies only 2 distinct connected regions of the image. Our goal is to exploit this natural image inductive bias and penalize those attention maps which do not lead to a spatial clustering of their largest values. Intuitively, if we compare Fig. 1b with Fig. 1c, we observe that, in the latter case (in which DINO was used for training), the attention maps are more "spatially ordered", i.e. there are *less* and *bigger* "blobs" (obtained after thresholding the map values (Caron et al., 2021)). Since an image is usually composed of a few main objects, each of which typically corresponds to one or very few connected regions of tokens, during training we penalize those attention maps which produce a large number of small blobs. We use this as an auxiliary pretext task which extracts information from images without additional annotation, by exploiting the assumption that spatially close tokens should preferably belong to the same cluster.

4.1 Spatial entropy loss

For each head of the last Transformer block, we compute a similarity map S^h ($1 \leq h \leq H$, see Sect. 3) by comparing the `[CLS]` token query ($\boldsymbol{q}^h_{CLS}$) with all the patch token keys ($\boldsymbol{k}^h_{x,y}$, where $(x, y) \in \{1, ..., k\}^2$):

$$S^h_{x,y} = < \boldsymbol{q}^h_{CLS}, \boldsymbol{k}^h_{x,y} > / \sqrt{d}, \quad (x, y) \in \{1, ..., k\}^2, \tag{4}$$

where $< \boldsymbol{a}, \boldsymbol{b} >$ is the dot product between $\boldsymbol{a}$ and $\boldsymbol{b}$. S^h is extracted from the self-attention map A^h by selecting the `[CLS]` token as the only query and before applying the softmax (see Sect. 4.3 for a discussion about this choice). S^h is a $k \times k$ matrix corresponding to the final $k \times k$ spatial grid of patches (Sect. 3), and (x, y) corresponds to the "coordinates" of a patch token in this grid.

In order to extract a set of connected regions containing the largest values in S^h, we zero-out those elements of S^h which are smaller than the mean value $m = 1/n \sum_{(x,y) \in \{1,...,k\}^2} S^h_{x,y}$:

$$B^h_{x,y} = \text{ReLU}\,(S^h_{x,y} - m), \quad (x, y) \in \{1, ..., k\}^2, \tag{5}$$

where thresholding using m corresponds to retain half of the total "mass" of Eq. 4. We can now use a standard algorithm (Grana et al., 2010) to extract the connected components[2] from B^h, obtained using an 8-connectivity relation between non-zero elements in B^h (see Fig. 2):

$$C^h = \{C_1, ..., C_{h_r}\} = \text{ ConnectedComponents } (B^h). \tag{6}$$

C_j $(1 \leq j \leq h_r)$ in C^h is the set of coordinates $(C_j = \{(x_1, y_1), ..., (x_{n_j}, y_{n_j})\})$ of the j-th connected component, whose cardinality (n_j) is variable, and such is the total number of components (h_r). Given C^h, we define the spatial entropy of S^h $(\mathcal{H}(S^h))$ as follows:

$$\mathcal{H}(S^h) = -\sum_{j=1}^{h_r} P^h(C_j) \log P^h(C_j), \tag{7}$$

$$P^h(C_j) = \frac{1}{|B^h|} \sum_{(x,y)\in C_j} B^h_{x,y}, \tag{8}$$

where $|B^h| = \sum_{(x,y)\in\{1,...,k\}^2} B^h_{x,y}$. Importantly, in Eq. 8, the probability $(P^h(C_j))$ of each region C_j is computed using all its elements, and this makes the difference with respect to a non-spatial entropy which is directly computed over all the individual elements in S^h, without considering the connectivity relation. Note that the less the number of components h_r or the less uniformly distributed the probability values $P^h(C_1), ...P^h(C_{h_r})$, the lower $\mathcal{H}(S^h)$. Using Eq. 7, the spatial entropy loss is defined as:

$$\mathcal{L}_{se} = \frac{1}{H} \sum_{h=1}^{H} \mathcal{H}(S^h). \tag{9}$$

$\mathcal{L}_{se}$ is used jointly with the main task loss. For instance, in case of supervised training, we use: $\mathcal{L}_{tot} = \mathcal{L}_{ce} + \lambda \mathcal{L}_{se}$, where λ is a weight used to balance the two losses.

4.2 Removing the skip connections

Raghu et al. (2021) empirically showed that, *in the last blocks* of ViT, the patch token representations are mostly propagated from the previous layers using the skip connections (Sect. 1). We presume this is (partially) due to the fact that only the `[CLS]` token is used as input to the classification MLP head (Sect. 3), thus, during training, the last-block patch token embeddings are usually neglected. Moreover, Raghu et al. (2021) show that the effective receptive field[3] (Luo et al., 2016) of each block, when computed *after* the MSA skip connections, is much smaller than the effective receptive field computed *before* the MSA

[2] In the rest of this article we will use the terms "connected components" or "connected regions" interchangeably. Specifically, a connected region $C = \{(x_1, y_1), ..., (x_n, y_n)\}$ in a binary map B is a set of points in B such that: (1) $\forall (x, y) \in C, B(x, y) = 1$, (2) $\forall (x_1, y_1), (x_2, y_2) \in C$, there exists a path connecting (x_1, y_1) with (x_2, y_2), and this path is included in C. The path is composed of a sequence of pairwise adjacent points.

[3] The *effective receptive field* is defined in (Luo et al., 2016) as: "How much each input pixel in a receptive field can impact the output of a unit n layers up the network".

skip connections. Both empirical observations lead to the conclusion that the MSA skip connections in the last blocks may be detrimental for the *representation* capacity of the final patch token embeddings. This problem is emphasized when using our spatial entropy loss, since it is computed using the attention maps of the last-block MSA (Sect. 4.1). For these reasons, we propose to remove the MSA skip connections in the last block (L). Specifically, in the L-th block, we replace Eq. 1-2 with:

$$\boldsymbol{z}' = \text{MSA}\,(\,\text{LN}\,(\boldsymbol{z}^{L-1})), \tag{10}$$

$$\boldsymbol{z}^{L} = \text{MLP}\,(\boldsymbol{z}') + \boldsymbol{z}'. \tag{11}$$

Note that, in addition to removing the MSA skip connections (Eq. 10), we also remove the subsequent LN (Eq. 11), because we empirically observed that this further improves the VT accuracy (see Sect. 5.1).

4.3 Discussion

In this section, we discuss and motivate the choices made in Sect. 4.1 and Sect. 4.2. First, we use S^h, extracted before the softmax (Eq. 3) because, using the softmax, the network can "cheat", by increasing the norm of the vectors $\boldsymbol{q}_{CLS}$ and $\boldsymbol{k}_{x,y}$ ($(x, y) \in \{1, ..., k\}^2$). As a result, the dot product $< \boldsymbol{q}_{CLS}, \boldsymbol{k}_{x,y} >$ also largely increases, and the softmax operation (based on the exponential function) enormously exaggerates the difference between the elements in S^h, generating a very peaked distribution, which zeros-out non-maxima (x, y) elements. We observed that, when using the softmax, the VT is able to minimize Eq. 9 by producing single-peak similarity maps which have a 0 entropy, each being composed of only one connected component with only one single token (i.e., $h_r = 1$ and $n_j = 1$).

Second, the spatial entropy (Eq. 7) is computed for each head separately and then averaged (Eq. 9) to allow each head to focus on *different* image regions. Note that, although computing the connected components (Eq. 6) is a non-differentiable operation, C^h is only used to "pool" the values of B^h (Eq. 8), and each C_j can be implemented as a binary mask (more details in the Appendix, where we also compare $\mathcal{L}_{se}$ with other solutions). It is also important to note that, although a smaller number of connected components (h_r) can decrease $\mathcal{L}_{se}$, this does not force the VT to always produce a *single* connected component (i.e., $h_r = 1$) because of the contribution of the main task loss (e.g., $\mathcal{L}_{ce}$). For instance, Fig. 1d shows 4 big connected components which correctly correspond to the non-occluded parts of the bear and their reflections into the river, respectively.

Finally, we remove the MSA skip connections only in the last block (Eqs. 10, 11) because, according to the results reported in (Raghu et al., 2021), removing the skip connections in the ViT intermediate blocks brings to an accuracy drop. In contrast, in Sect. 5.1 we show that our strategy, which keeps the ViT architecture unchanged apart from the last block, is beneficial even when used *without* our spatial entropy loss. Similarly, in preliminary experiments in which we used the spatial entropy loss also in other intermediate layers ($l < L$), we did not observe any significant improvement. In the rest of this paper, we refer to our full method SAR as composed of the spatial entropy loss (Sect. 4.1) and the last-block MSA skip connection and LN removal (Sect. 4.2).

Table 1 IN-100. (a) Influence of the spatial entropy loss weight λ (100 training epochs). (b) Influence of the number of epochs. (c) Analysis of the different components of SAR (100 epochs)

λ	Top-1 Acc.	
(a)		
0	75.72	
0.001	75.82	
0.005	76.16	
0.01	**76.72**	
0.05	76.22	
0.1	75.88	
Model	Top-1 Acc.	
	100 ep.	300 ep.
(b)		
ViT-S/16	74.22	80.82
ViT-S/16+SAR	**76.72** (+2.5)	**85.24** (+4.42)
Model	Top-1 Acc.	
(c)		
A: Baseline	74.22	
B: A + no MSA skip connections	74.64 (+0.42)	
C: B + no LN	75.72 (+1.5)	
D: C + no MLP skip connections	73.76 (-0.46)	
E: A + spatial entropy loss	74.78 (+0.56)	
F: A + SAR	76.72 (+2.5)	

The best results are given in bold

Table 2 Training from scratch on small-medium datasets. The baseline (ViT-S/16) results are reported from (El-Nouby et al., 2021)

Dataset	Train samples	Test samples	ViT-S/16	ViT-S/16+SAR
Stanford Cars	8,144	8,041	35.3	**64.65** (+29.35)
Clipart	34,019	14,818	41.0	**64.95** (+23.95)
Painting	52,867	22,892	38.4	**57.11** (+18.17)
Sketch	49,115	21,271	37.2	**62.98** (+30.78)

The best results are given in bold

Table 3 IN-100 experiments with different sampling ratios (100 epochs). † These results were obtained by us using the publicly available code taken from (Liu et al., 2021)

Model	Top-1 Acc.			
	0.25	0.50	0.75	1.00
ViT-S/16	21.66	29.86	35.62	74.22
ViT-S/16+DRLoc†	25.32 (+3.66)	34.9 (+5.04)	39.22 (+3.6)	75.81 (+1.59)
ViT-S/16+SAR	**29.06** (+7.4)	**38.02** (+8.16)	**46.12** (+10.5)	**76.72** (+2.5)

The best results are given in bold

5 Experiments

In Sect. 5.1 we analyse the contribution of the spatial entropy loss and the skip connection removal. In Sect. 5.2 we show that SAR improves ViT in different training–testing scenarios and with different downstream tasks. In Sect. 5.3 we analyse the properties of the attention maps generated using SAR. In the Appendix, we provide additional experiments using multi-label classification and other tasks, and we show how SAR can be used jointly with fully self-supervised learning approaches. We train the models using a maximum of 8 NVIDIA V100 32GB GPUs for the most computationally intensive experiments. For other experiments (e.g. transfer learning), we scale down to lower resources. In the Appendix we report a detailed list of the computational hardware utilized in each experimental setting.

5.1 Ablation study

In this section, we analyse the influence of the λ value (Sect. 4.1), the removal of the skip connections and the LN in the last ViT block (Sect. 4.2), and the use of the spatial entropy loss (Sect. 4.1). In all the ablation experiments, we use ImageNet-100 (IN-100) (Tian et al., 2020; Wang & Isola, 2020), which is a subset of 100 classes of ImageNet, and ViT-S/16, a 22 million parameter ViT (Dosovitskiy et al., 2021) trained with 224×224 resolution images and 14×14 patches tokens ($k = 14$) with a patch resolution of 16×16 (Touvron et al., 2021). Moreover, in all the experiments in this section, we adopt the training protocol and the data-augmentations described in (Liu et al., 2021). Note that these data-augmentations include, among other things, the use of Mixup (Zhang et al., 2018) and CutMix (Yun et al., 2019) (which are also used in all the supervised classification experiments of Sect. 5.2), and this shows that our entropy loss can be used jointly with "image-mixing" techniques.

Table 4 IN-1K experiments (300 training epochs)

Method	Top-1 Acc.
ViT-S/16 (Touvron et al., 2021)	79.8
ViT-S/16 + DRLoc†	80.2 (+0.4)
ViT-S/16 + SAR	**80.7** (+0.9)

The best result is given in bold

†This result has been obtained by us using the publicly available code taken from (Liu et al., 2021)

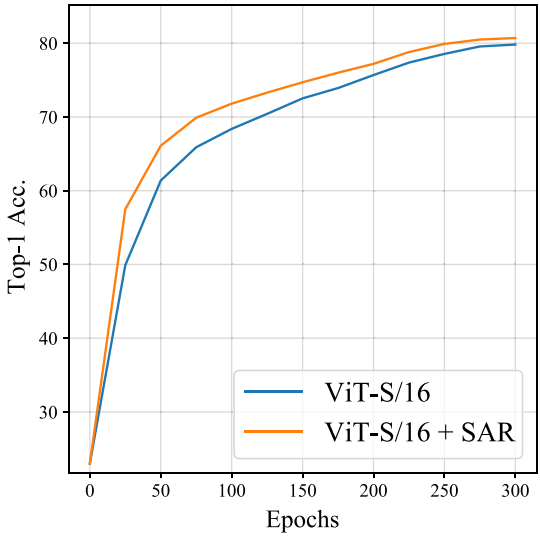

Fig. 3 IN-1K, validation set accuracy with respect to the number of training epochs

In Table 1a, we train from scratch all the models using 100 epochs and we show the impact on the test set accuracy using different values of λ. In the experiments of this table, we use our loss function ($\mathcal{L}_{tot} = \mathcal{L}_{ce} + \lambda\mathcal{L}_{se}$) and we remove both the skip connections and the LN in the last block (Eqs. 10, 11), thus the column $\lambda = 0$ corresponds to the result reported in Table 1c, Row "C" (see below). In the rest of the paper, we use the results obtained with this setting (IN-100, 100 epochs, etc.) and the best λ value ($\lambda = 0.01$) for **all** the other datasets, training scenarios (e.g., training from scratch, fine-tuning, fully self-supervised learning, etc.) and VT architectures (e.g., ViT, CvT, PVT, etc.). In fact, although a higher accuracy can very likely be obtained by tuning λ, our goal is to show that SAR is an easy-to-use regularization approach, even without tuning its only hyperparameter.

In Table 1 (c), we train from scratch all the models using 100 epochs and Row "A" corresponds to our run of the original ViT-S/16 (Eq. 1-2). When we remove the MSA skip connections (Row "B"), we observe a +0.42 points improvement, which becomes +1.5 if we also remove the LN (Row "C"). This experiment confirms that the last block patch tokens can learn more useful representations if we inhibit the MSA identity path (Eq. 10-11). However, if we also remove the skip connections in the subsequent MLP layer (Row "D"), the results are inferior to the baseline. Finally, when we use the spatial entropy loss with the original architecture (Row "E"), the improvement is marginal, but using $\mathcal{L}_{se}$ jointly with Eq. 10-11 (full model, Row "F"), the accuracy boost with respect to the baseline is

Table 5 Object detection on PASCAL VOC 2007, evaluated using mean Average Precision (mAP)

Model	mAP
ViT-S/16	77.1
ViT-S/16 + SAR	**79.2** (+2.1)

The best result is given in bold

Table 6 Semantic segmentation on PASCAL VOC 2007, evaluated using mean Intersection over Union (mIoU)

Model	mIoU
ViT-S/16	66.17
ViT-S/16 + SAR	**68.68** (+2.51)

The best result is given in bold

much stronger. Table 1 (b) compares training with 100 and 300 epochs and shows that, in the latter case, SAR can reach a much higher relative improvement with respect to the baseline (+ 4.42).

5.2 Main results

Sample efficiency In order to show that SAR can alleviate the need of large labeled datasets (Sect. 1), we follow a recent trend of works (Liu et al., 2021; El-Nouby et al., 2021; Cao & Wu, 2021) where VTs are trained from scratch on small-medium datasets (without pre-training on ImageNet). Specifically, we strictly follow the training protocol proposed by El-Nouby et al. (2021), where 5,000 epochs are used to train ViT-S/16 directly on each target dataset. The results are shown in Table 2, which also provides the number of training and testing samples of each dataset, jointly with the accuracy values of the baseline (ViT-S/16, trained in a standard way, without SAR), both reported from El-Nouby et al. (2021). Table 2 shows that SAR can *drastically* improve the ViT-S/16 accuracy on these small-medium datasets, with an improvement ranging from + 18.17 to + 30.78 points. These results, jointly with the results obtained on IN-100 (Table 1 (b)), show that SAR is particularly effective in boosting the performance of ViT when labeled training data are scarce.

We further analyze the impact of the amount of training data using different subsets of IN-100 with different sampling ratios (ranging from 25 to 75%, with images randomly selected). We use the same training protocol of Table 1 (b) (e.g., 100 training epochs, etc.) and we test on the *whole* IN-100 validation set. Table 3 shows the results, confirming that, with less data, the accuracy boost obtained using SAR can significantly increase (e.g., with 75% of the data we have a 10.5 points improvement). In the same table, we compare SAR with Dense Relative Localization (DRLoc) loss (Liu et al., 2021), which, similarly to SAR, is based on an auxiliary self-supervised task used to regularize VTs training (Sect. 2). DRLoc encourages the VT to learn spatial relations within an image by predicting the relative distance between the (x, y) positions of randomly sampled output embeddings from the $k \times k$ grid of the last layer L. Table 3 shows that SAR largely outperforms DRLoc, especially in a low-data regime (e.g., with 75% of the data, the difference between SAR and DRLoc is 6.9 points).

Training on ImageNet-1K We extend the previous results training ViT on ImageNet-1K (IN-1K), and comparing SAR with the baseline (ViT-S/16, trained in a standard

Table 7 Transfer learning results (100 epochs fine-tuning). The first row corresponds to a standard fine-tuning protocol, while the other configurations include SAR either in the pre-training or in the fine-tuning stage

SAR Pre-training	SAR Fine-tuning	CIFAR-10	CIFAR-100	Flowers	Pets
✗	✗	98.59	88.95	95.07	92.21
✓	✗	98.69 (+0.1)	89.19 (+0.24)	96.05 (+0.98)	92.7 (+0.49)
✗	✓	98.72 (+0.13)	88.95	95.1 (+0.03)	92.34 (+0.13)
✓	✓	98.65 (+0.06)	89.21 (+0.26)	96.1 (+1.03)	92.7 (+0.49)

Table 8 Out-of-distribution testing on ImageNet-A (IN-A) and ImageNet-C (IN-C)

Model	IN-A (Acc. ↑)	IN-C (mCE ↓)
ViT-S/16	19.2	52.8
ViT-S/16 + SAR	**22.39** (+3.19)	**51.6** (-1.2)

The best results are given in bold

way, without SAR), and with DRLoc. Table 4 shows that SAR can boost the accuracy of ViT of almost 1 point *without any additional learnable parameters* or drastic architectural changes, and this gain is higher than DRLoc. The reason why the relative improvement is smaller with respect to what was obtained with smaller datasets is likely due to the fact that, usually, regularization techniques are mostly effective with small(er) datasets (Balestriero et al., 2022). Nevertheless, Fig. 3 shows that SAR can be used jointly with large datasets to significantly speed-up training. For instance, ViT-S/16 + SAR, with 100 epochs, achieves almost the same accuracy as the baseline trained with 150 epochs, while we surpass the final baseline accuracy (79.8% at epoch 300) with only 250 training epochs (79.9% at epoch 250). From a computational point of view, reducing the number of epochs needed for convergence by one sixth on a large dataset may be a significant acceleration, also considering that, on average, the overall computational overhead of SAR (with non-optimized code) is only + 2.9% (further details on Sect. A). Finally, the two regularization approaches (SAR and DRLoc) can potentially be combined, but we leave this for future work.

Transfer learning with object detection and image segmentation tasks We further analyze the quality of the models pre-trained on IN-1K using object detection and semantic segmentation downstream tasks. Specifically, we use ViTDet (Li et al., 2022), a recently proposed object detection/segmentation framework in which a (standard) pre-trained ViT backbone is adapted *only at fine-tuning time* in order to generate a feature pyramid to be used for multi-scale object detection or image segmentation. Note that, as mentioned in Sect. 1, hybrid approaches which are based on ad hoc architectures are not suitable from this framework, because they need to redesign their backbone and introduce a feature pyramid also in the pre-training stage (Li et al., 2022; Wang et al., 2021). Conversely, we use the pre-trained networks whose results are reported in Table 4, where the baseline is ViT-S/16 and our approach corresponds to ViT-S/16 + SAR. For the object detection task, following (Girshick, 2015), we use the trainval set of PASCAL VOC 2007 and 2012 (Everingham et al., 2010) (16.5K training images) to fine-tune the two models using ViTDet, and the test set of PASCAL VOC 2007 for evaluation. The

Table 9 IN-100 experiments with different VTs (100 training epochs). The results of the baselines have been obtained by us using the corresponding publicly available code

Model	Top-1 Acc.
ViT-S/16	74.22
ViT-S/16+SAR	**76.72** (+2.5)
T2T-14	82.42
T2T-14+SAR	**83.96** (+1.54)
PVT-S	76.57
PVT-S+SAR	**77.78** (+1.21)
CvT-13	83.38
CvT-13+SAR	**85.20** (+1.82)

The best results are given in bold

Table 10 IN-1K experiments with different VTs (300 training epochs). All results but ours are reported from the corresponding paper. The number of parameters is in millions

Model	Param.	Acc.
ViT-S/16	22	79.8
ViT-S/16+SAR	22	**80.7** (+0.9)
T2T-14	21.5	81.5
T2T-14+SAR	21.5	**81.9** (+0.4)
PVT-S	24.5	79.8
PVT-S+SAR	24.5	79.84 (+0.04)
CvT-13	20	81.6
CvT-13+SAR	20	**81.8** (+0.2)

The best results are given in bold

results, reported in Table 5, show that the model pre-trained using SAR outperforms the baseline of more than 2 points, which is an increment even larger that the boost obtained in the classification task used during pre-training (Table 4). Similarly, for the segmentation task, we use PASCAL VOC-12 trainval for fine-tuning and PASCAL VOC 2007 test for evaluation. Table 6 shows that the model pre-trained with SAR achieves an improvement of more than 2.5 mIoU points compared to the baseline. These detection and segmentation improvements confirm that the local inductive bias introduced in ViT using SAR can be very useful for localization tasks, especially when the fine-tuning data are scarce like in PASCAL VOC.

Transfer learning with different fine-tuning protocols In this battery of experiments, we evaluate SAR in a transfer learning scenario with classification tasks. We adopt the four datasets used in Dosovitskiy et al. (2021); Touvron et al. (2021); Chen et al. (2021); Caron et al. (2021): CIFAR-10 and CIFAR-100 (Krizhevsky, 2009), Oxford Flowers102 (Nilsback & Zisserman, 2008), and Oxford-IIIT-Pets (Everingham et al., 2010). The standard transfer learning protocol consists in pre-training on IN-1K, and then fine-tuning on each dataset. This corresponds to the first row in Table 7, where the IN-1K pre-trained model is ViT-S/16 in Table 4. The next three rows show different pre-training/fine-tuning configurations, in which we use SAR in one of the two phases or in both (see the Appendix for more details). All the configurations lead to an overall improvement of the accuracy with respect to the baseline, and show that SAR can be

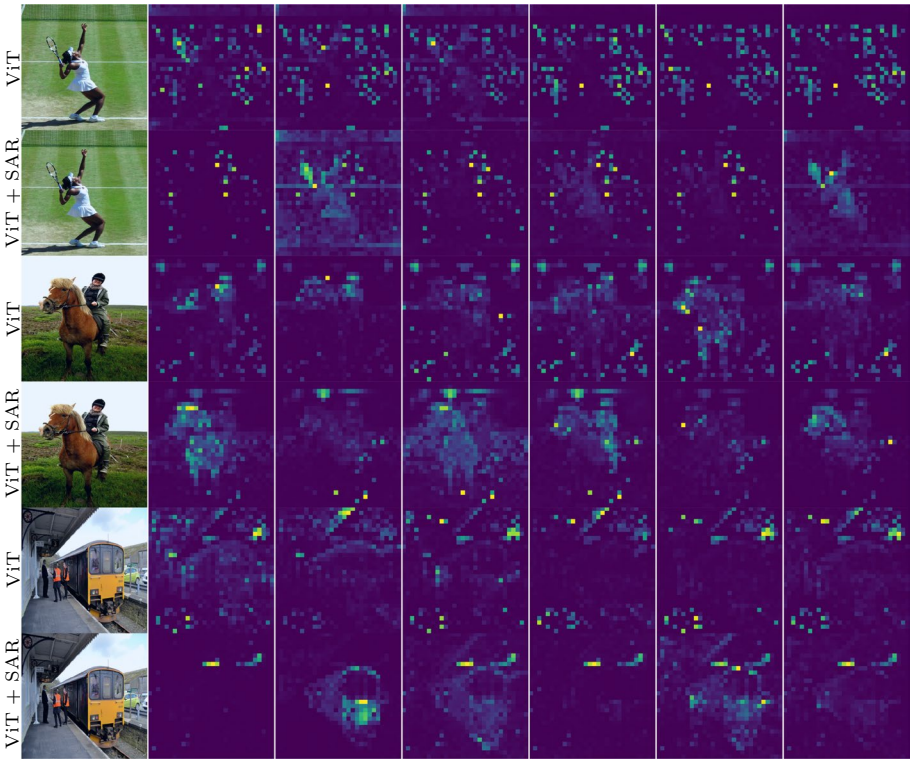

Fig. 4 A qualitative comparison between the attention maps generated by ViT-S/16 and ViT-S/16 + SAR. For each image, we show all the 6 attention maps (A^h) corresponding to the 6 last-block heads, computed using only the `[CLS]` token query

used flexibly. For instance, SAR can be used when fine-tuning a VT trained in a standard way, without the need to re-train it on ImageNet.

Out-of-distribution testing We test the robustness of our ViT trained with SAR when the testing distribution is different from the training distribution. Specifically, following (Bai et al., 2021), we use two different testing sets: (1) ImageNet-A (Hendrycks et al., 2021), which are real-world images but collected from challenging scenarios (e.g., occlusions, fog scenes, etc.), and (2) ImageNet-C (Hendrycks & Dietterich, 2019), which is designed to measure the model robustness against common image corruptions.

Note that training is done only on IN-1K. Thus, in Table 8, ViT-S/16 and ViT-S/16 + SAR correspond to the models we trained on IN-1K whose results on the IN-1K standard validation set are reported in Table 4. ImageNet-A and ImageNet-C are used *only for testing*, hence they are useful to assess the behaviour of a model when evaluated on a distribution different from the training distribution (Bai et al., 2021). The results reported in Table 8 show that SAR can significantly improve the robustness of ViT (note that, with the mCE metric, the lower the better (Bai et al., 2021)). We presume that this is a side-effect of our spatial entropy loss minimization, which leads to heads usually focusing on the foreground objects and, therefore, reducing the dependence with respect to the background appearance variability distribution.

Table 11 A comparison of the segmentation properties of the attention maps on PASCAL VOC-12, measuring the Jaccard similarity

Model	Jaccard (↑)
ViT-S/16	19.18
ViT-S/16+SAR	**31.19** (+12.01)

The best result is given in bold

Different VT architectures Finally, we show that SAR can be used with VTs of different capacities and with architectures different from ViT. For this purpose, we plug SAR into the following VT architectures: ViT-S/16 (Touvron et al., 2021), T2T (Yuan et al., 2021), PVT (Wang et al., 2021) and CvT (Wu et al., 2021). Specifically, T2T, PVT and CvT are hybrid architectures, which use typical CNN mechanisms to introduce a local inductive bias into the VT training (Sects. 1 and 2). We omit other common frameworks such as, for instance, Swin (Liu et al., 2021) because of the lack of a `[CLS]` token in their architecture. Although the `[CLS]` token used, e.g, in Sect. 4.1 to compute S^h, can potentially be replaced by a vector obtained by average-pooling all the patch embeddings, we leave this for future investigations. Moreover, for computational reasons, we focus on small-medium capacity VTs (see Table 10 for details on the number of parameters of each VT). Importantly, for each tested method, we use the original training protocol developed by the corresponding authors, including, e.g., the learning rate schedule, the batch size, the VT-specific hyperparameter values and the data-augmentation type used to obtain the corresponding published results, both when we train the baseline and when we train using SAR. Moreover, as usual (Sect. 5.1), we keep fixed the only SAR hyperparameter ($\lambda = 0.01$). Although better results can likely be obtained by adopting the common practice of hyperparameter tuning (including the VT-specific hyperparameters), our goal is to show that SAR can be easily used in different VTs, increasing their final testing accuracy. The results reported in Table 9 and Table 10 show that SAR improves *all* the tested VTs, independently of their specific architecture, model capacity or training protocol. Note that both PVT and CvT have a final grid resolution of 7×7, which is smaller than the 14×14 grid used in ViT and T2T, and this probably has a negative impact on our spatial based entropy loss.

Overall, the results reported in Tables 9 and 10: (1) Confirm that SAR is mostly useful with smaller datasets (being the relative improvements on IN-100 significantly larger than those obtained on IN-1K). (2) Show that the object-based inductive bias introduced when training with SAR is (partially) complementary with respect to the local bias embedded in the hybrid VT architectures, as witnessed by the positive boost obtained when these VTs are used jointly with SAR. (3) Show that, on IN-1K, the accuracy of ViT-S/16 + SAR is comparable with the hybrid VTs (without SAR). However, the advantage of ViT-S/16 + SAR is its simplicity, which does not need drastic architectural changes to the original ViT architecture, where the latter is quickly becoming a *de facto* standard in many vision and vision-language tasks (Sect. 2).

```
# bs    : batch size
# H     : number of heads
# k X k : size of the embedding grid
# S     : defined in Eq. 4 (main paper),
#         tensor with shape [bs, H, k, k]

def SpatialEntropyLoss(S, eps=1e-9):
    bs, H, k, _ = S.size()
    m = torch.mean(S, dim=[2, 3], keepdim=True)
    B = torch.relu(S - m) + eps

    with torch.no_grad():
        batch_idxs, head_idxs, M = connected_components(B)

    p_u = torch.sum(B[batch_idxs, head_idxs] * M, dim=[2, 3])
    p_n = p_u / torch.sum(B, dim=[2, 3])
    SE = -torch.sum((p_n * torch.log(p_n)) / (bs * H)
    return SE
```

Fig. 5 PyTorch-like pseudocode of the Spatial Entropy loss

5.3 Attention map analysis

This section qualitatively and quantitatively analyses the attention maps obtained using SAR. Note that, as mentioned in Sects. 1 and 2, we do not directly use the attention map clusters for segmentation tasks or as input to a post-processing step. Thus, the goal of this analysis is to show that the spatial entropy loss minimization effectively results in attention maps with spatial clusters, leaving their potential use for a segmentation-based post-processing as a future work.

Figure 4 visually compares the attention maps obtained with ViT-S/16 and ViT-S/16 + SAR. As expected, standard training generates attention maps with a widely spread structure. Conversely, using SAR, a semantic segmentation structure clearly emerges. In the Appendix, we show additional results.

For a quantitative analysis, we follow the protocol used in (Caron et al., 2021; Naseer et al., 2021), where the Jaccard similarity is used to compare the ground-truth segmentation masks of the objects in PASCAL VOC-12 with the thresholded attention masks of the last ViT block. Specifically, the attention maps of all the heads are thresholded to keep 60% of the mass, and the head with the highest Jaccard similarity with the ground-truth is selected (Caron et al., 2021; Naseer et al., 2021). Table 11 shows that SAR significantly improves the segmentation results, quantitatively confirming the qualitative analysis in Fig. 4.

6 Conclusions

In this paper we proposed SAR, a regularization method which exploits the connectedness property of the objects to introduce a local inductive bias into the VT training. By penalizing spatially disordered attention maps, an additional self-supervised signal can be extracted from the sample images, thereby reducing the reliance on large numbers of labeled training samples. Using different downstream tasks and training–testing protocols (including fine-tuning and out-of-distribution testing), we showed tha SAR can

Table 12 Object detection on PASCAL VOC 2007. We report the mean and the standard deviation ($\pm\sigma$) computed over 5 runs

Model	mAP
ViT-S/16	77.18 ± 0.10
ViT-S/16+SAR	**79.16** ± 0.05 (+1.98)

The best result is given in bold

significantly boost the accuracy of a ViT backbone, especially when the training data are scarce. Although SAR can also be used jointly with hybrid VTs, its main advantage over the latter is the possibility to be easily plugged into the original ViT backbone, whose architecture is widely adopted in many vision and vision language tasks.

Future work SAR can be extended to VT for videos. In fact, a "temporal inductive bias" contained in videos is that natural objects usually move smoothly and, thus, they can be represented by few connected *3D* regions in, e.g., a sequence of T consecutive frames. Thus, Equation (4) can be extended e.g., by comparing the `[CLS]` token query with all the patch token keys contained in these T frames, keeping the rest of the algorithm unchanged.

Another promising direction for a future work is combining SAR with DRLoc (Liu et al., 2021): they are both training regularization approaches for VTs and their joint use can lead to a further improvement of the sample efficiency.

Limitations Since training VTs is very computationally expensive, in our experiments we used only small/medium capacity VTs. We leave the extension of our empirical analysis to larger capacity VTs for the future. For the same computational reasons, we have not tuned hyperparameters on the datasets. However, we believe that the SAR accuracy improvement, obtained in all the tested scenarios without hyperparameter tuning, further shows its robustness and ease to use.

Appendix

Pseudo-code of the spatial entropy loss and computational analysis

Figure 5 shows the pseudo-code for computing $\mathcal{L}_{se}$ (Eq. 9). The goal is twofold: to show how easy is to compute $\mathcal{L}_{se}$, and how to make it differentiable. Specifically, Eq. 6 is based on a connected component algorithm which is not differentiable. However, once C^h is computed, each element $C_j \in C^h$ ($C_j = \{(x_1, y_1), ..., (x_{n_j}, y_{n_j})\}$) can be represented as a binary mask M_j, defined as:

$$M_j[x, y] = \begin{cases} 1, & \text{if } (x, y) \in C_j \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

Using Eq. 12, we compute the probability $P^h(C_j)$ of the component C_j using matrix multiplication, which is done efficiently on GPUs. In practice, Eq. 8 is computed as:

$$P^h(C_j) = \frac{1}{|B^h|} \sum_{(x,y)} B^h_{x,y} \odot M_j[x, y], \tag{13}$$

Table 13 Semantic segmentation on PASCAL VOC 2007. We report the mean and the standard deviation ($\pm\sigma$) computed over 5 runs

Model	mIoU
ViT-S/16	65.94 ± 0.33
ViT-S/16+SAR	**68.62** ± 0.11 (+2.68)

The best result is given in bold

Table 14 Multi-label classification on PASCAL VOC 2007 with models trained from scratch. We report the mean and the standard deviation over 5 runs

Model	mAP
ViT-S/16	72.37 ± 0.07
ViT-S/16+SAR	**75.16** ± 0.13 (+2.79)

The best result is given in bold

Table 15 Multi-label classification on PASCAL VOC 2007 with models pre-trained on ImageNet-1K. We report the mean and the standard deviation over 5 runs

Model	mIoU
ViT-S/16	95.21 ± 0.14
ViT-S/16+SAR	**96.77** ± 0.10 (+1.56)

The best result is given in bold

where $\odot$ is the element-wise product. This implementation makes it is possible to back-propagate the spatial entropy loss even if the connected component algorithm is not differentiable.

The algorithm shown in Fig. 5 takes as input S^h, which is extracted from the h-th head of the last MSA layer, without the need to be recomputed. On the other hand, *connected_components*(B^h) is $O(k^2)$, where B^h is a $k \times k$ matrix. Since all the other operations of the algorithm are negligable from a computational point of view, the overall overhead introduced by SAR is $O(Hk^2)$, being H the number of heads of the last MSA layer. Empirically, *using non-optimized code* and 224×224 resolution images, we have that, on average, the run time for processing a single sample is 5.84 ms (ViT-S/16) and 6.01 ms (ViT-S/16 + SAR), corresponding to an overhead of 2.9%.

Comparing the spatial entropy loss with other solutions

In our preliminary experiments, we replaced the spatial entropy loss with a different loss based on a *total variation denoising* (Rudin et al., 1992) criterion, which we formulated as:

$$\mathcal{L}_{tv} = \frac{1}{H} \sum_{h=1}^{H} \sum_{(x,y) \in \{1,\dots,k\}^2} |S^h_{x,y} - S^h_{x,y+1}| + |S^h_{x,y} - S^h_{x+1,y}|. \tag{14}$$

Table 16 Self-supervised training on IN-100 (300 epochs). (a) Accuracy on IN-100 evaluated through linear probing and a single testing crop. (b) Segmentation properties of the corresponding attention maps evaluated on PASCAL VOC-12. The results of the baselines have been obtained by us using the corresponding publicly available code

Method	Top-1 Acc.
(a)	
MoCo-v3	77.60
MoCo-v3 + SAR	**78.88** (+1.28)
DINO	77.42
DINO + SAR	**79.90** (+2.48)
Method	Jaccard similarity
(b)	
MoCo-v3	27.40
MoCo-v3 + SAR	**33.10** (+6.17)
DINO	35.06
DINO + SAR	**35.28** (+0.22)

The best results are given in bold

However, $\mathcal{L}_{tv}$ drastically underperforms $\mathcal{L}_{se}$ and tends to produce blurred (more uniform) attention maps. The main difference between $\mathcal{L}_{tv}$ and $\mathcal{L}_{se}$ is that the thresholding (Eq. 5) and the adjacency-based clustering (Eq. 6) operations in $\mathcal{L}_{se}$ group together image regions of variable size and shape which have in common high attention scores for a specific head, and, therefore, they presumably represent the same semantics (e.g., a specific object). In contrast, $\mathcal{L}_{tv}$ compares patch tokens which are adjacent to each other (e.g., $S^h_{x,y}$ and $S^h_{x,y+1}$) but which do not necessarily share the same semantics (e.g., $S^h_{x,y}$ may belong to the background while $S^h_{x,y+1}$ belongs to a foreground object). Thus, the implicit local inductive bias of the two losses is different: in case of $\mathcal{L}_{se}$, the inductive bias is that the image regions corresponding to the highest attention scores (for a specific head) should be spatially grouped in few, big "blobs", while, in the case of $\mathcal{L}_{tv}$, the inductive bias is that generic adjacent regions should have similar attention scores. Note that a similar result would presumably be obtained using any other binary relation, even one different from the adjacency relation (such as, for example, the Mutual Information used in (Isola et al., 2015).

Finally, we have empirically tested slightly different similarity metrics. For instance, replacing Eq. 4 with a cosine similarity computed as:

$$\hat{S}^h_{x,y} = \frac{< \boldsymbol{q}^h_{CLS}, \boldsymbol{k}^h_{x,y} >}{||\boldsymbol{q}^h_{CLS}|| \cdot ||\boldsymbol{k}^h_{x,y}||}, \quad (x,y) \in \{1,...,k\}^2, \tag{15}$$

and keeping all the rest unchanged (e.g., Eqs. 5–9), we get a slightly lower accuracy when training on IN-100 (76.25 top-1 accuracy versus 76.72 in Table 1 (c)). This is likely due to the fact that $\hat{S}^h$ in Eq. 15 does not correspond to the formula used to compute the attention for the main task loss (Eq. 3). Thus, merging the main task loss (e.g., the cross entropy loss) with the spatial entropy loss may be more difficult.

Fig. 6 A qualitative comparison between the thresholded attention maps generated by ViT-S/16 and ViT-S/16+SAR using the protocol described in (Caron et al., 2021). The ground truth segmentation maps are shown in the first column

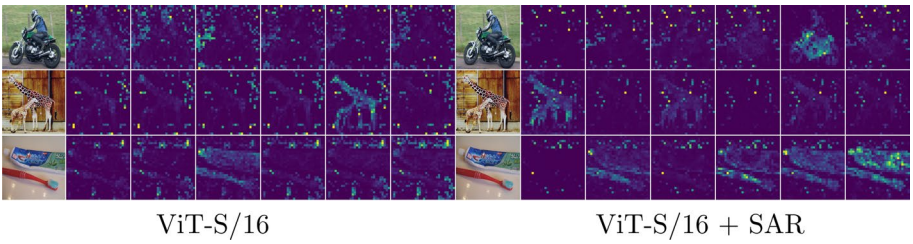

Fig. 7 A qualitative comparison between the attention maps generated by ViT-S/16 and ViT-S/16 + SAR (supervised case, training on IN-1K). Zoom in to see details

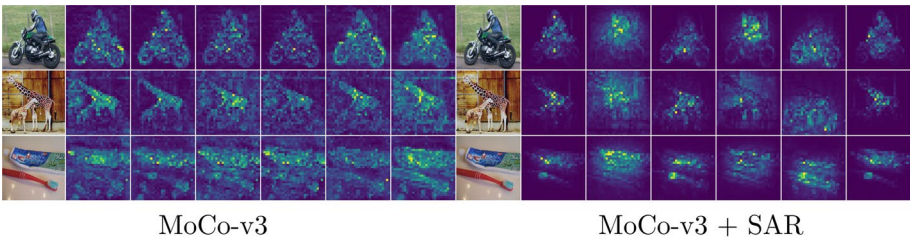

Fig. 8 A qualitative comparison between the attention maps generated by MoCo-v3 and MoCo-v3 + SAR with the ViT-S/16 backbone (training on IN-100). For each image, we show all the 6 attention maps (A^h) corresponding to the 6 last-block heads, computed using only the `[CLS]` token query. Zoom in to see details

Multi-object tasks

In this section, we analyze the performance of SAR with multi-object target tasks. We use the PASCAL VOC dataset (Everingham et al., 2010), in which images contain multiple annotated objects, and the usual training–testing split (see Sect. 5.2). Given the medium size of the dataset (16.5K training samples), we repeated each experiment 5 times with random seeds, reporting the corresponding mean and standard deviation. Specifically, we evaluate the models on object detection, segmentation, and multi-label classification

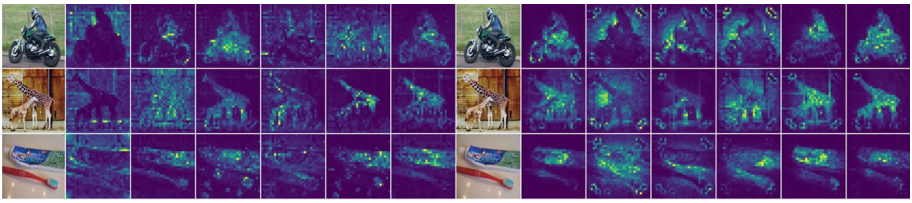

Fig. 9 A qualitative comparison between the attention maps generated by DINO and DINO + SAR with the ViT-S/16 backbone (training on IN-100). Zoom in to see details

tasks. In Table 12 Table 13 we respectively repeat the detection and the segmentation task described in Sect. 5.2, and we report the statistics averaged over the 5 runs.

The multi-label classification task consists in predicting the presence of an object class independently of other classes possibly present in the same image, where each ground-truth image is labeled with multiple labels. Since multiple objects may coexist in the same image, as a standard practice for multi-label classification, we replace the softmax operation on the classification MLP head with a sigmoid function and we use the binary cross entropy for training. We perform two sets of experiments, starting from random weights (Table 14) or fine-tuning a model pre-trained on ImageNet-1K (Table 15). Specifically, in case of the fine-tuning experiments, we use SAR both in the pre-training and in the fine-tuning phase (see Sect. 5.2, "Transfer learning with different fine-tuning protocols"). All the experiments are repeated 5 times and we report both the mean and the standard deviation.

The experiments of this section show that SAR is effective also when multiple objects coexist in the same image and the task involves classifying, detecting or segmenting all these objects.

Self-supervised experiments

In this section, we use SAR in a fully self-supervised scenario. Analogously to all the other experiments of this paper, we keep fixed $\lambda = 0.01$ and we use the hyperparameters of each tested VT-based method without any hyperparameter tuning. Since self-supervised learning algorithms are very time consuming, we use IN-100, which is a medium-size dataset. We plug SAR on top of two state-of-the-art VT-based self-supervised learning algorithms: MoCo-v3 (Chen et al., 2021) and DINO (Caron et al., 2021) (Sect. 2). When we use MoCo-v3, in $\mathcal{L}_{tot}$ (Sect. 4.1), we replace the cross-entropy loss ($\mathcal{L}_{ce}$) with the contrastive loss used in (Chen et al., 2021). Similarly, when we use DINO, we use as the main task loss the "self-distillation" proposed in (Caron et al., 2021), jointly with its multi-crop strategy. We use the official code of MoCo-v3 and DINO and we strictly follow the algorithms and the training protocols of the baseline methods, including all the default hyperparameters suggested by the corresponding authors. However, for computational reasons, we used a 1024 batch size for MoCo-v3 and MoCo-v3 + SAR, and a batch size of 512 for DINO and DINO + SAR. The VT backbone is ViT-S/16 for all the methods. More details in Sect. F.

We evaluate all models (with and without SAR) using the standard self-supervised evaluation protocol, consisting in freezing the network after training and then training a linear classifier on top of the frozen features (Caron et al., 2021; Chen et al., 2021). The results are reported in Table 16 (a), and show that, on IN-100, SAR significantly improves these state-of-the-art algorithms, including DINO (which inspired our work).

We qualitatively compare the attention maps obtained with and without SAR in Fig. 8 (MoCo-v3) and Fig. 9 (DINO). Figure 8 shows that, in MoCo-v3 + SAR, the head-specific attention maps focus on slightly different aspects of the main object, while in MoCo-v3, the attention is much more "disordered" (spread over the whole image). On the other hand, when comparing DINO with DINO + SAR (Fig. 9), the attention map differences are more subtle. However, the higher inter-head variability in DINO + SAR is one interesting difference. For instance, while DINO's maps usually focus only on the main foreground object, in DINO + SAR, different heads cover different foreground objects (e.g., the cat and the sink in Row 11) or different background regions (e.g., the road and the sky in the "train" figure of Row 5). This difference is probably due to the difference in how DINO and DINO + SAR are optimized. In fact, in DINO, the only source of supervision is the comparison between two different views of the same image (Sect. 2), which likely encourages the network to focus on the objects most frequently in common. On the other hand, in DINO + SAR, the creation of connected regions with large attention values into each head's map is also encouraged using $\mathcal{L}_{se}$.

These qualitative observations are confirmed by the quantitative analysis reported in Table 16 (b), where we follow the protocol described in Sect. 5.3. SAR increases the Jaccard similarity of both self-supervised algorithms.

An additional qualitative analysis of the attention maps

In this section, we extend the analysis of Sect. 5.3 providing additional visualizations of the attention maps obtained using supervised training on IN-1K. In Fig. 6 we show the thresholded attention maps, obtained using the protocol described in (Caron et al., 2021) (see Sect. 5.3 for details) and the image samples used in Fig. 4. The results are compared with the ground-truth segmentation maps, and illustrate the experiments of Table 11. Note that the segmentation-like effects of these thresholded attention maps can potentially be used for model explainability, in the same fashion they were use in DINO.

Figure 7 shows the attention maps of ViT-S/16 and ViT-S/16 + SAR. These attention maps show that the ViT-S/16 attention scores are spread over all the image, while in ViT-S/16 + SAR they are much more spatially clustered and usually focused on the main object(s) of the input image. For example, the first row shows that the heads of ViT-S/16 focus on the upper part of the image, and only the keyboard of the laptop emerges. Vice versa, from the ViT-S/16 + SAR attention heads, it is possible to recognise the shape and size of the laptop precisely. Similarly, the second last row shows an example where the input image contains some elephants. While the different heads of ViT-S/16 seem to focus mainly on the background, the first head of ViT-S/16 + SAR is clearly focused on the elephants. Importantly, these visualizations show that different heads of ViT-S/16 + SAR usually focus on different semantic concepts, which shows that there are no collapse phenomena using the spatial entropy loss (see Eq. 7 and the corresponding discussion in Sect. 4.3).

Implementation details

In the following, we list the implementation details. For computational reasons, all the experiments of this paper, except those presented in Sect. C, have been done only once, thus we cannot report standard deviation values computed over multiple runs. However,

this is the standard practice adopted by all the evaluation protocols used in this paper and by all the methods we have compared with.

ViT based models

We train our models using the public code of Touvron et al. (2021)[4] for ViT and we modify the original code when we use SAR, as described in Sect. 4.2. The models are trained with a batch size of 1024, using the AdamW optimizer (Loshchilov & Hutter, 2019) with initial learning rate of 0.001, a cosine learning rate schedule, a weight decay of 0.05 and using the original data-augmentation protocol, including the use of Mixup (Zhang et al., 2018) and CutMix (Yun et al., 2019) for the supervised classification tasks. We use 8 NVIDIA V100 32GB GPUs to train the models on IN-1K, and 4 GPUs for the set of experiments on IN-100.

Hybrid architectures

In all the supervised experiments, we used the officially released code for PVT (Wang et al., 2021),[5] T2T (Yuan et al., 2021)[6] and CvT (Wu et al., 2021),[7] strictly following the original training protocol for each architecture.

PVT and T2T are trained with batch size of 1024, using the AdamW optimizer with an initial learning rate of 0.001, momentum 0.9 and weight decay of 0.05. CvT is trained with a batch size of 2048 and an initial learning rate of 0.02, decayed with a cosine schedule. The data augumentations of the original articles are based on the DeiT protocol (Touvron et al., 2021). We refer the reader to the original papers for further details. When we use SAR, we plug it on top of the original public code, following Sect. 4.2. We use 8 NVIDIA V100 32GB GPUs to train the models on IN-1K, and 4 GPUs for the set of experiments on IN-100.

Object detection and semantic segmentation downstream tasks

Starting from the pretrained ViT-S/16 and ViT-S/16 + SAR models of Table 4, we follow (Li et al. (2022)) to obtain a feature pyramid and a Mask-RCNN detector (He et al., 2017). In both the object detection and the segmentation task, the models are fine-tuned for 25 epochs with the AdamW optimizer, learning rate of 0.0001 and linear warmup for the first 250 iterations. The input image is 512×512 during training, augmented with random cropping and flipping. We used a batch size of 8 distributed on 4 GPUs.

Multi-label classification

We train the models with the binary cross-entropy as the main loss. The resolution of the input image is 224 × 224 and, during training, we apply the standard augmentation techniques using Mixup (Zhang et al., 2018), which we found beneficial to further boost the

[4] https://github.com/facebookresearch/deit

[5] https://github.com/whai362/PVT/tree/v1

[6] https://github.com/yitu-opensource/T2T-ViT

[7] https://github.com/microsoft/CvT

performance of all the models (with and without SAR). We use the AdamW optimizer, with a learning rate of 0.0001, batch size of 128, and train the models for 100 epochs on 2 GPUs.

Transfer learning with different fine-tuning protocols

We fine-tune the ViT-S/16 models pretrained on IN-1K (see Table 4), always keeping unchanged the VT architecture used in the pre-training stage. This is done to avoid making the adaptation task more difficult, since each of the four datasets used in Table 7 is composed of a relatively small number of samples. For instance, when SAR is used during pre-training but removed during fine-tuning (second row of Table 7), in the fine-tuning stage we use only $\mathcal{L}_{ce}$ for training but we do *not* re-introduce skip connections or LN layers in the last block (i.e., we use Eq. 10-Eq. 11 when fine-tuning). Conversely, when pre-training is done without SAR, which is introduced only in the fine-tuning stage (third row of Table 7), when fine-tuning, we use $\mathcal{L}_{tot} = \mathcal{L}_{ce} + \lambda\mathcal{L}_{se}$, but keeping the standard skip connections and the LN layer in the last block (Eq. 1-Eq. 2).

The models are fine-tuned for 100 epochs with a batch size of 512, and an initial learning rate of 0.0005 decayed with a cosine schedule. For this set of experiments, we utilized 2 NVIDIA V100 GPUs.

Self-supervised experiments

In our self-supervised experiments, we adopt the original code for MoCo-v3 (Chen et al., 2021)[8] and DINO (Caron et al., 2021)[9] with a ViT-S/16 backbone. For computational reasons, we restrict our experiments training the models on IN-100 for 300 epochs. Moreover, to fit the available computational resources, we reduce the batch size to 1024 for MoCo-v3, while DINO is trained with the default multi-crop strategy ($2 \times 224^2 + 10 \times 96^2$), but with a batch size of 512. We thoroughly follow the authors' specifications for the other hyperparameters. The results in Table 16 are obtained using a standard linear evaluation protocol in which the pretrained backbone is frozen, and a linear classifier is trained on top of it, using SGD for 100 epochs on IN-100. Due to the high computational cost, the models are trained on 8 GPUs.

Dataset licensing details

CIFAR-10, CIFAR-100 are released to the public with a non-commercial research and/or educational use.[10] Oxford flower102 is released to the public with an unknown license through its website,[11] and we assume a non-commercial research and/or educational use. ImageNet annotations have a non-commercial research and educational license.[12] PASCAL VOC 2012 images abide by the Flickr Terms of Use.[13] Stanford Cars images have a non-commercial research and educational license[14] ClipArt, Painting and Sketches are part

[8] https://github.com/facebookresearch/moco-v3

[9] https://github.com/facebookresearch/dino

[10] https://www.cs.toronto.edu/~kriz/cifar.html

[11] https://www.robots.ox.ac.uk/~vgg/data/flowers/102/

[12] https://image-net.org/download

[13] http://host.robots.ox.ac.uk/pascal/VOC/voc2012/

[14] http://ai.stanford.edu/~jkrause/cars/car_dataset.html

of the DomainNet dataset which is released under a fair use license.[15] The ImageNet-A[16] and ImageNet-C[17] images are released with unknown licence, so we refer to the original authors to use these datasets.

Acknowledgements Enver Sangineto, Bruno Lepri and Nicu Sebe acknowledge funding by the European Union's Horizon Europe research and innovation program under grant agreement No. 101120237 (ELIAS). Bruno Lepri and Nicu Sebe also acknowledge the support of the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU.

Author Contributions E.P., Y.L. wrote the code. E.P., E.S., Y.L., M.D.N conducted the research. B.L., W.B., and N.S. provided the funding and supported the research. All authors wrote and reviewed the paper.

Funding Open access funding provided by Università degli Studi di Trento within the CRUI-CARE Agreement.

Data availability The data is public and available online.

Code availability The code is available at https://github.com/helia95/SAR.

Declarations

Conflict of interest Not applicable.

Ethics approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

References

Alayrac, J.-B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., et al. (2022). Flamingo: A visual language model for few-shot learning. *Advances in Neural Information Processing Systems, 35*, 23716–23736.

Altieri, L., Cocchi, D., & Roli, G. (2018). SpatEntropy: Spatial Entropy Measures in R. arxiv:1804.05521.

Asano, Y. M., Rupprecht, C., & Vedaldi, A. (2020). *A critical analysis of self-supervision, or what we can learn from a single image*. ICLR: OpenReview.net.

Bachmann, R., Mizrahi, D., Atanov, A., & Zamir, A. (2022). Multimae: Multimodal multi-task masked autoencoders. ECCV (37) (Vol. 13697, pp. 348–367). Springer.

Bai, Y., Mei, J., Yuille, A.L., & Xie, C. (2021). Are transformers more robust than cnns? Neurips (pp. 26831–26843).

[15] http://ai.bu.edu/DomainNet/#dataset

[16] https://github.com/hendrycks/natural-adv-examples

[17] https://github.com/hendrycks/robustness

Balestriero, R., Bottou, L., & LeCun, Y. (2022). The effects of regularization and data augmentation are class dependent. *Advances in Neural Information Processing Systems, 35*, 37878–37891.
Bao, H., Dong, L., Piao, S., & Wei, F. (2022). *Beit: BERT pre-training of image transformers*. ICLR: OpenReview.net.
Bao, H., Wang, W., Dong, L., & Wei, F. (2022). VL-BEiT: Generative visionlanguage pretraining. arxiv:2206.01127.
Bardes, A., Ponce, J., & LeCun, Y. (2022). *Vicreg: Variance-invariance-covariance regularization for self-supervised learning*. Iclr: OpenReview.net.
Batty, M. (1974). Spatial entropy. Geographical analysis (Vol. 6, pp. 1–31). Wiley Online Library.
Bautista, M.A., Sanakoyeu, A., Tikhoncheva, E., Ommer, B. (2016). Cliquecnn: Deep unsupervised exemplar learning. Advances in Neural Information Processing Systems, 29
Cao, Y., &Wu, J. (2021). Rethinking self-supervised learning: Small is beautiful. arxiv:2103.13559.
Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., & Zagoruyko, S. (2020). End-to-end object detection with transformers. ECCV (Vol. 12346, pp. 213–229). Springer.
Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), ECCV (Vol. 11218, pp. 139–156). Springer.
Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., & Joulin, A. (2020). Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems, 33*, 9912–9924.
Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. ICCV (pp. 9650–9660).
Ceci, M., Corizzo, R., Malerba, D., & Rashkovska, A. (2019). Spatial autocorrelation and entropy for renewable energy forecasting. Data min. knowl. discov.
Chang, H., Zhang, H., Jiang, L., Liu, C., & Freeman, W.T. (2022). Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11315–11325).
Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., & Sutskever, I. (2020). Generative pretraining from pixels. In *International conference on machine learning* (pp. 1691–1703).
Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In: *International conference on machine learning* (pp. 1597–1607).
Chen, X., Ding, M., Wang, X., Xin, Y., Mo, S., Wang, Y., & Wang, J. (2024). Context autoencoder for self-supervised representation learning. *International Journal of Computer Vision, 132*(1), 208–223.
Chen, X., & He, K. (2021). Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 15750–15758).
Chen, X., Xie, S., & He, K. (2021). An empirical study of training self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9640–9649).
Dai, Z., Cai, B., Lin, Y., Chen, J. (2021). Up-detr: Unsupervised pre-training for object detection with transformers. In *Proceedings of the IEEE/cvf conference on computer vision and pattern recognition* (pp. 1601–1610).
Didolkar, A., Goyal, A., Ke, N.R., Blundell, C., Beaudoin, P., Heess, N., Bengio, Y. (2021). Neural production systems. In *NIPS*.
Dong, X., Bao, J., Zhang, T., Chen, D., Zhang, W., Yuan, L., . . . Guo, B. (2023). Peco: Perceptual codebook for BERT pre-training of vision transformers. In *AAAI* (pp. 552–560). AAAI Press.
Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., . . . Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. In *9th international conference on learning representations*, ICLR 2021, virtual event, Austria, May 3–7, 2021. OpenReview.net. Retrieved from https://openreview.net/forum?id=YicbFdNTTy
Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P., & Zisserman, A. (2021). With a little help from my friends: Nearest-neighbor contrastive learning of visual representations. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9588–9597).
d'Ascoli, S., Touvron, H., Leavitt, M.L., Morcos, A.S., Biroli, G., & Sagun, L. (2021). Convit: Improving vision transformers with soft convolutional inductive biases. In *International conference on machine learning* (pp. 2286–2296).
El-Nouby, A., Izacard, G., Touvron, H., Laptev, I., Jégou, H., & Grave, E. (2021). Are large-scale datasets necessary for self-supervised pre-training? CoRR, abs/2112.10740.
Engelcke, M., Parker Jones, O., & Posner, I. (2021). Genesis-v2: Inferring unordered object representations without iterative refinement. *Advances in Neural Information Processing Systems, 34*, 8085–8094.
Ermolov, A., Siarohin, A., Sangineto, E., & Sebe, N. (2021). Whitening for selfsupervised representation learning. In *International conference on machine learning* (pp. 3015–3024).

Everingham, M., Gool, L. V., Williams, C. K. I., Winn, J. M., & Zisserman, A. (2010). The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision, 88*(2), 303–338.

Gansbeke, W.V., Vandenhende, S., Georgoulis, S., Proesmans, M., & Gool, L.V. (2020). SCAN: Learning to classify images without labels. In *ECCV*.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 1440–1448).

Goyal, A., Lamb, A., Gampa, P., Beaudoin, P., Levine, S., Blundell, C., . . . Mozer, M. (2020). Object files and schemata: Factorizing declarative and procedural knowledge in dynamical systems. arxiv: 2006.16225.

Grana, C., Borghesani, D., & Cucchiara, R. (2010). Optimized block-based connected components labeling with decision trees. In *IEEE Transactions on Image Processing* (Vol. 19, pp. 1596–1609).

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., et al. (2020). Bootstrap your own latent-a new approach to selfsupervised learning. *Advances in Neural Information Processing Systems, 33*, 21271–21284.

Guo, M.-H., Cai, J.-X., Liu, Z.-N., Mu, T.-J., Martin, R. R., & Hu, S.-M. (2021). Pct: Point cloud transformer. *Computational Visual Media, 7*, 187–199.

Hassani, A., Walton, S., Li, J., Li, S., & Shi, H. (2023). Neighborhood attention transformer. In *CVPR* (pp. 6185–6194). IEEE.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R. (2022). Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 16000–16009).

He, K., Fan, H., Wu, Y., Xie, S., Girshick, R. (2020). Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9729–9738).

He, K., Gkioxari, G., Dollár, P., & Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp. 2961–2969).

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770–778).

Hendrycks, D., & Dietterich, T. G. (2019). *Benchmarking neural network robustness to common corruptions and perturbations*. ICLR (poster): OpenReview.net.

Hendrycks, D., Zhao, K., Basart, S., Steinhardt, J., & Song, D. (2021). Natural adversarial examples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 15262–15271).

Herzig, R., Ben-Avraham, E., Mangalam, K., Bar, A., Chechik, G., Rohrbach, A., . . . Globerson, A. (2022). Object-region video transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 3148–3159).

Hjelm, R.D., Fedorov, A., Lavoie-Marchildon, S., Grewal, K., Bachman, P., Trischler, A., & Bengio, Y. (2019). Learning deep representations by mutual information estimation and maximization. In *7th international conference on learning representations*, ICLR. OpenReview.net.

Hua, T., Tian, Y., Ren, S., Raptis, M., Zhao, H., & Sigal, L. (2023). *Self-supervision through random segments with autoregressive coding (randsac)*. ICLR: OpenReview.net.

Hua, T., Wang, W., Xue, Z., Ren, S., Wang, Y., Zhao, H. (2021). On feature decorrelation in self-supervised learning. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9598–9608).

Hudson, D.A., & Zitnick, L. (2021). Generative adversarial transformers. In *International conference on machine learning* (pp. 4487–4499).

Isola, P., Zoran, D., Krishnan, D., & Adelson, E.H. (2014). Crisp boundary detection using pointwise mutual information. In *ECCV* (3) (Vol. 8691, pp. 799–814). Springer.

Ji, X., Henriques, J.F., & Vedaldi, A. (2019). Invariant information clustering for unsupervised image classification and segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9865–9874).

Jiang, Y., Chang, S., & Wang, Z. (2021). Transgan: Two pure transformers can make one strong gan, and that can scale up. NIPS (pp. 14745–14758).

Kakogeorgiou, I., Gidaris, S., Psomas, B., Avrithis, Y., Bursuc, A., Karantzalos, K., & Komodakis, N. (2022). What to Hide from Your Students: Attention- Guided Masked Image Modeling. arxiv: 2203.12719.

Kang, H., Mo, S., & Shin, J. (2022). Remixer: Object-aware mixing layer for vision transformers and mixers. Iclr2022 workshop on the elements of reasoning: Objects, structure and causality.

Kenton, J.D.M.-W.C., & Toutanova, L.K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. Proceedings of naacl (Vol. 1, p. 2).

Krizhevsky, A. (2009). Learning multiple layers of features from tiny images. Retrieved from https://api.semanticscholar.org/CorpusID:18268744

Li, K., Wu, Z., Peng, K., Ernst, J., & Fu, Y. (2020). Guided attention inference network. In *IEEE Transactions on Pattern Analysis and Machine Intelligence* (Vol. 42, pp. 2996–3010).

Li, Y., Fujita, H., Li, J., Liu, C., & Zhang, Z. (2022). Tensor approximate entropy: An entropy measure for sleep scoring. *Knowledge-based Systems* (Vol. 245, p. 108503).

Li, Y., Mao, H., Girshick, R.B., & He, K. (2022). Exploring plain vision transformer backbones for object detection. ECCV (Vol. 13669, pp. 280–296). Springer.

Li, Y., Wu, C.-Y., Fan, H., Mangalam, K., Xiong, B., Malik, J., Feichtenhofer, C. (2022). Mvitv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 4804–4814).

Li, Y., Zhang, K., Cao, J., Timofte, R., & Gool, L.V. (2021). LocalViT: Bringing locality to vision transformers. arxiv:2104.05707.

Liu, Y., Sangineto, E., Bi, W., Sebe, N., Lepri, B., & Nadai, M. (2021). Efficient training of visual transformers with small datasets. *Advances in Neural Information Processing Systems, 34*, 23818–23830.

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., . . . Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 10012–10022).

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., . . . Kipf, T. (2020). Object-centric learning with slot attention. *NIPS* (Vol. 33, pp. 11525–11538).

Loshchilov, I., & Hutter, F. (2019). *Decoupled weight decay regularization*. ICLR (poster): OpenReview.net.

Luo, W., Li, Y., Urtasun, R., & Zemel, R.S. (2016). Understanding the effective receptive field in deep convolutional neural networks. *NIPS* (pp. 4898– 4906).

Meinhardt, T., Kirillov, A., Leal-Taixe, L., & Feichtenhofer, C. (2022). Trackformer: Multi-object tracking with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8844–8854).

Naseer, M. M., Ranasinghe, K., Khan, S. H., Hayat, M., Shahbaz Khan, F., & Yang, M.-H. (2021). Intriguing properties of vision transformers. *Advances in Neural Information Processing Systems, 34*, 23296–23308.

Neimark, D., Bar, O., Zohar, M., Asselmann, D. (2021). Video transformer network. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 3163–3172).

Nilsback, M.-E., & Zisserman, A. (2008). Automated flower classification over a large number of classes. In *Sixth indian conference on computer vision, graphics & image processing* (pp. 722–729).

Radford, A., & Narasimhan, K. (2018). Improving language understanding by generative pre-training.

Raghu, M., Unterthiner, T., Kornblith, S., Zhang, C., & Dosovitskiy, A. (2021). Do vision transformers see like convolutional neural networks? *Neurips* (pp. 12116–12128).

Ramesh, A., Dhariwal, P., Nichol, A., Chu, C., Chen, M. (2022). Hierarchical text-conditional image generation with CLIP latents. arxiv:2204.06125.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., . . . Sutskever, I. (2021). Zero-shot text-to-image generation. In *International conference on machine learning* (pp. 8821–8831).

Rao, Y., Zhao, W., Chen, G., Tang, Y., Zhu, Z., Huang, G., . . . Lu, J. (2022). Denseclip: Language-guided dense prediction with context-aware prompting. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 18082–18091).

Razlighi, Q., & Kehtarnavaz, N. (2009). A comparison study of image spatial entropy. In *Visual communications and image processing* (Vol. 7257, pp. 615–624).

Rudin, L.I., Osher, S., & Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena* (Vol. 60, pp. 259–268).

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision, 115*, 211–252.

Sajjadi, M. S., Duckworth, D., Mahendran, A., Van Steenkiste, S., Pavetic, F., Lucic, M., & Kipf, T. (2022). Object scene representation transformer. *Advances in Neural Information Processing Systems, 35*, 9512–9524.

Shah, D., Zaveri, T., Trivedi, Y.N., Plaza, A. (2020). Entropy-based convex set optimization for spatial–spectral endmember extraction from hyperspectral images. In *IEEE journal of selected topics in applied earth observations and remote sensing* (Vol. 13, pp. 4200–4213).

Strudel, R., Garcia, R., Laptev, I., Schmid, C. (2021). Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 7262–7272).

Tian, Y., Krishnan, D., Isola, P. (2020). Contrastive multiview coding. *ECCV* (pp. 776–794).

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., & Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. In *International conference on machine learning* (pp. 10347–10357).

Tupin, F., Sigelle, M., Maitre, H. (2000). Definition of a spatial entropy and its use for texture discrimination. *ICIP*.

van den Oord, A., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. arxiv:1807.03748.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., . . . Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, 30.

Wang, T., & Isola, P. (2020). Understanding contrastive representation learning through alignment and uniformity on the hypersphere. In *International conference on machine learning* (pp. 9929–9939).

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., . . . Shao, L. (2021). Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 568–578).

Wang, Z., Yu, J., Yu, A. W., Dai, Z., Tsvetkov, Y., & Cao, Y. (2022). *Simvlm: Simple visual language model pretraining with weak supervision*. ICLR: OpenReview.net.

Wei, C., Fan, H., Xie, S.,Wu, C.-Y., Yuille, A., & Feichtenhofer, C. (2022). Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 14668–14678).

Wu, H., Xiao, B., Codella, N., Liu, M., Dai, X., Yuan, L., & Zhang, L. (2021). Cvt: Introducing convolutions to vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 22–31).

Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., . . . Hu, H. (2022). Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 9653–9663).

Xu, W., Xu, Y., Chang, T., & Tu, Z. (2021). Co-scale conv-attentional image transformers. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9981–9990).

Yuan, K., Guo, S., Liu, Z., Zhou, A., Yu, F., Wu, W. (2021). Incorporating convolution designs into visual transformers. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 579–588).

Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.-H., . . . Yan, S. (2021). Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 558–567).

Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., & Yoo, Y. (2019). Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6023–6032).

Yun, S., Lee, H., Kim, J., & Shin, J. (2022). Patch-level representation learning for self-supervised vision transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 8354–8363).

Zbontar, J., Jing, L., Misra, I., LeCun, Y., & Deny, S. (2021). Barlow twins: Selfsupervised learning via redundancy reduction. In *International conference on machine learning* (pp. 12310–12320).

Zhang, H., Cissé, M., Dauphin, Y. N., & Lopez-Paz, D. (2018). *mixup: Beyond empirical risk minimization*. ICLR (poster): OpenReview.net.

Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V. (2021). Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 16259–16268).

Zheng, C., Zhu, S., Mendieta, M., Yang, T., Chen, C., & Ding, Z. (2021). 3d human pose estimation with spatial and temporal transformers. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 11656–11665).

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A.L., & Kong, T. (2022). iBOT: Image BERT Pre-Training with Online Tokenizer. ICLR.

Zhu, X., Su, W., Lu, L., Li, B., Wang, X., & Dai, J. (2021). *Deformable DETR: Deformable transformers for end-to-end object detection*. ICLR: OpenReview.net.

Zhuang, C., Zhai, A.L., & Yamins, D. (2019). Local aggregation for unsupervised learning of visual embeddings. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 6002–6012).