ConViT: Improving Vision Transformers with Soft Convolutional Inductive Biases

Stéphane d'Ascoli [1 2] **Hugo Touvron** [2] **Matthew L. Leavitt** [2] **Ari S. Morcos** [2] **Giulio Biroli** [1 2] **Levent Sagun** [2]

Abstract

Convolutional architectures have proven extremely successful for vision tasks. Their hard inductive biases enable sample-efficient learning, but come at the cost of a potentially lower performance ceiling. Vision Transformers (ViTs) rely on more flexible self-attention layers, and have recently outperformed CNNs for image classification. However, they require costly pre-training on large external datasets or distillation from pre-trained convolutional networks. In this paper, we ask the following question: is it possible to combine the strengths of these two architectures while avoiding their respective limitations? To this end, we introduce *gated positional self-attention* (GPSA), a form of positional self-attention which can be equipped with a "soft" convolutional inductive bias. We initialize the GPSA layers to mimic the locality of convolutional layers, then give each attention head the freedom to escape locality by adjusting a *gating parameter* regulating the attention paid to position versus content information. The resulting convolutional-like ViT architecture, *ConViT*, outperforms the DeiT (Touvron et al., 2020) on ImageNet, while offering a much improved sample efficiency. We further investigate the role of locality in learning by first quantifying how it is encouraged in vanilla self-attention layers, then analyzing how it is escaped in GPSA layers. We conclude by presenting various ablations to better understand the success of the ConViT. Our code and models are released publicly at `https://github.com/facebookresearch/convit`.

[1]Department of Physics, Ecole Normale Supérieure, Paris, France [2]Facebook AI Research, Paris, France. Correspondence to: Stéphane d'Ascoli <stephane.dascoli@ens.fr>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

1. Introduction

The success of deep learning over the last decade has largely been fueled by models with strong inductive biases, allowing efficient training across domains (Mitchell, 1980; Goodfellow et al., 2016). The use of Convolutional Neural Networks (CNNs) (LeCun et al., 1998; 1989), which have become ubiquitous in computer vision since the success of AlexNet in 2012 (Krizhevsky et al., 2017), epitomizes this trend. Inductive biases are hard-coded into the architectural structure of CNNs in the form of two strong constraints on the weights: locality and weight sharing. By encouraging translation equivariance (without pooling layers) and translation invariance (with pooling layers) (Scherer et al., 2010; Schmidhuber, 2015; Goodfellow et al., 2016), the convolutional inductive bias makes models more sample-efficient and parameter-efficient (Simoncelli & Olshausen, 2001; Ruderman & Bialek, 1994). Similarly, for sequence-based tasks, recurrent networks with hard-coded memory cells have been shown to simplify the learning of long-range dependencies (LSTMs) and outperform vanilla recurrent neural networks in a variety of settings (Gers et al., 1999; Sundermeyer et al., 2012; Greff et al., 2017).

However, the rise of models based purely on attention in recent years calls into question the necessity of hard-coded inductive biases. First introduced as an add-on to recurrent neural networks for Sequence-to-Sequence models (Bahdanau et al., 2014), attention has led to a breakthrough in Natural Language Processing through the emergence of Transformer models, which rely solely on a particular kind of attention: Self-Attention (SA) (Vaswani et al., 2017). The strong performance of these models when pre-trained on large datasets has quickly led to Transformer-based approaches becoming the default choice over recurrent models like LSTMs (Devlin et al., 2018).

In vision tasks, the locality of CNNs impairs the ability to capture long-range dependencies, whereas attention does not suffer from this limitation. Chen et al. (2018) and Bello et al. (2019) leveraged this complementarity by augmenting convolutional layers with attention. More recently, Ramachandran et al. (2019) ran a series of experiments replacing some or all convolutional layers in ResNets with attention, and found the best performing models used convolu-

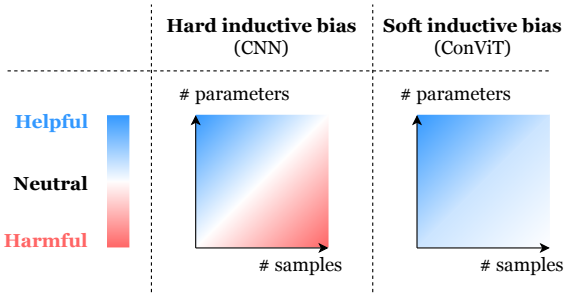


Figure 1. **Soft inductive biases can help models learn without being restrictive.** Hard inductive biases, such as the architectural constraints of CNNs, can greatly improve the sample-efficiency of learning, but can become constraining when the size of the dataset is not an issue. The soft inductive biases introduced by the ConViT avoid this limitation by vanishing away when not required.

tions in early layers and attention in later layers. The Vision Transformer (ViT), introduced by Dosovitskiy et al. (2020), entirely dispenses with the convolutional inductive bias by performing SA across embeddings of patches of pixels. The ViT is able to match or exceed the performance of CNNs but requires pre-training on vast amounts of data. More recently, the Data-efficient Vision Transformer (DeiT) (Touvron et al., 2020) was able to reach similar performances without any pre-training on supplementary data, instead relying on Knowledge Distillation (Hinton et al., 2015) from a convolutional teacher.

Soft inductive biases The recent success of the ViT demonstrates that while convolutional constraints can enable strongly sample-efficient training in the small-data regime, they can also become limiting as the dataset size is not an issue. In data-plentiful regimes, hard inductive biases can be overly restrictive and *learning* the most appropriate inductive bias can prove more effective. The practitioner is therefore confronted with a dilemma between using a convolutional model, which has a high performance floor but a potentially lower performance ceiling due to the hard inductive biases, or a self-attention based model, which has a lower floor but a higher ceiling. This dilemma leads to the following question: can one get the best of both worlds, and obtain the benefits of the convolutional inductive biases without suffering from its limitations (see Fig. 1)?

In this direction, one successful approach is the combination of the two architectures in "hybrid" models. These models, which interleave or combine convolutional and self-attention layers, have fueled successful results in a variety of tasks (Carion et al., 2020; Hu et al., 2018a; Ramachandran et al., 2019; Chen et al., 2020; Locatello et al., 2020; Sun et al., 2019; Srinivas et al., 2021; Wu et al., 2020). Another approach is that of Knowledge Distillation (Hinton et al., 2015), which has recently been applied to transfer the inductive bias of a convolutional teacher to a student transformer (Touvron et al., 2020). While these two methods offer an interesting compromise, they forcefully induce convolutional inductive biases into the Transformers, potentially affecting the Transformer with their limitations.

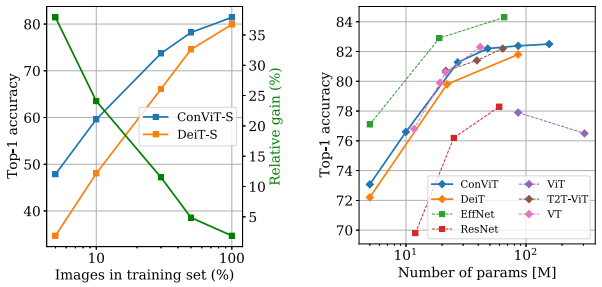
Contribution In this paper, we take a new step towards bridging the gap between CNNs and Transformers, by presenting a new method to "softly" introduce a convolutional inductive bias into the ViT. The idea is to let each SA layer decide whether to behave as a convolutional layer or not, depending on the context. We make the following contributions:

1. We present a new form of SA layer, named *gated positional self-attention* (GPSA), which one can initialize as a convolutional layer. Each attention head then has the freedom to recover expressivity by adjusting a *gating parameter*.
2. We then perform experiments based on the DeiT (Touvron et al., 2020), with a certain number of SA layers replaced by GPSA layers. The resulting Convolutional Vision Transformer (ConViT) outperforms the DeiT while boasting a much improved sample-efficiency (Fig. 2).
3. We analyze quantitatively how local attention is naturally encouraged in vanilla ViTs, then investigate the inner workings of the ConViT and perform ablations to investigate how it benefits from the convolution initialization.

Overall, our work demonstrates the effectiveness of "soft" inductive biases, especially in the low-data regime where the learning model is highly underspecified (see Fig. 1), and motivates the exploration of further methods to induce them.

Related work Our work is motivated by combining the recent success of pure Transformer models (Dosovitskiy et al., 2020) with the formalized relationship between SA and convolution. Indeed, Cordonnier et al. (2019) showed that a SA layer with N_h heads can express a convolution of kernel size $\sqrt{N_h}$, if each head focuses on one of the pixels in the kernel patch. By investigating the qualitative aspect of attention maps of models trained on CIFAR-10, it is shown that SA layers with relative positional encodings naturally converge towards convolutional-like configurations, suggesting that some degree of convolutional inductive bias is desirable.

Conversely, the restrictiveness of hard locality constraints has been proven by Elsayed et al. (2020). A breadth of approaches have been taken to imbue CNN architectures with nonlocality (Hu et al., 2018b;c; Wang et al., 2018; Wu



(a) Sample efficiency (b) Parameter efficiency

Figure 2. **The ConViT outperforms the DeiT both in sample and parameter efficiency**. *Left:* we compare the sample efficiency of our ConViT-S (see Tab. 1) with that of the DeiT-S by training them on restricted portions of ImageNet-1k, where we only keep a certain fraction of the images of each class. Both models are trained with the hyperparameters reported in (Touvron et al., 2020). We display the the relative improvement of the ConViT over the DeiT in green. *Right:* we compare the top-1 accuracies of our ConViT models with those of other ViTs (diamonds) and CNNs (squares) on ImageNet-1k. The performance of other models on ImageNet are taken from (Touvron et al., 2020; He et al., 2016; Tan & Le, 2019; Wu et al., 2020; Yuan et al., 2021).

et al., 2020). Another line of research is to induce a convolutional inductive bias is different architectures. For example, Neyshabur (2020) uses a regularization method to encourage fully-connected networks (FCNs) to learn convolutions from scratch throughout training.

Most related to our approach, d'Ascoli et al. (2019) explored a method to initialize FCNs networks as CNNs. This enables the resulting FCN to reach much higher performance than achievable with standard initialization. Moreover, if the FCN is initialized from a partially trained CNN, the recovered degrees of freedom allow the FCN to outperform the CNN it stems from. This method relates more generally to "warm start" approaches such as those used in spiked tensor models (Anandkumar et al., 2016), where a smart initialization, containing prior information on the problem, is used to ease the learning task.

Reproducibility We provide an open-source implementation of our method as well as pretrained models at the following address: `https://github.com/facebookresearch/convit`.

2. Background

We begin by introducing the basics of SA layers, and show how positional attention can allow SA layers to express convolutional layers.

Multi-head self-attention The attention mechanism is based on a trainable associative memory with (key, query) vector pairs. A sequence of L_1 "query" embeddings $\boldsymbol{Q} \in \mathbb{R}^{L_1 \times D_h}$ is matched against another sequence of L_2 "key" embeddings $\boldsymbol{K} \in \mathbb{R}^{L_2 \times D_h}$ using inner products. The result is an attention matrix whose entry (ij) quantifies how semantically "relevant" $\boldsymbol{Q}_i$ is to $\boldsymbol{K}_j$:

$$\boldsymbol{A} = \mathrm{softmax}\left(\frac{\boldsymbol{Q}\boldsymbol{K}^\top}{\sqrt{D_h}}\right) \in \mathbb{R}^{L_1 \times L_2}, \tag{1}$$

where $(\mathrm{softmax}\,[\boldsymbol{X}])_{ij} = \boldsymbol{e}^{\boldsymbol{X}_{ij}} / \sum_k e^{\boldsymbol{X}_{ik}}$.

Self-attention is a special case of attention where a sequence is matched to itself, to extract the semantic dependencies between its parts. In the ViT, the queries and keys are linear projections of the embeddings of 16×16 pixel patches $\boldsymbol{X} \in \mathbb{R}^{L \times D_{\mathrm{emb}}}$. Hence, we have $\boldsymbol{Q} = \boldsymbol{W}_{qry}\boldsymbol{X}$ and $\boldsymbol{K} = \boldsymbol{W}_{key}\boldsymbol{X}$, where $\boldsymbol{W}_{key}, \boldsymbol{W}_{qry} \in \mathbb{R}^{D_{\mathrm{emb}} \times D_h}$.

Multi-head SA layers use several self-attention heads in parallel to allow the learning of different kinds of interdependencies. They take as input a sequence of L embeddings of dimension $D_{\mathrm{emb}} = N_h D_h$, and output a sequence of L embeddings of the same dimension through the following mechanism:

$$\mathrm{MSA}(\boldsymbol{X}) := \underset{h \in [N_h]}{\mathrm{concat}} \left[\mathrm{SA}_h(\boldsymbol{X})\right] \boldsymbol{W}_{out} + \boldsymbol{b}_{out}, \tag{2}$$

where $\boldsymbol{W}_{out} \in \mathbb{R}^{D_{\mathrm{emb}} \times D_{\mathrm{emb}}}, b_{out} \in \mathbb{R}^{D_{\mathrm{emb}}}$. Each self-attention head h performs the following operation:

$$\mathrm{SA}_h(\boldsymbol{X}) := \boldsymbol{A}^h \boldsymbol{X} \boldsymbol{W}^h_{val}, \tag{3}$$

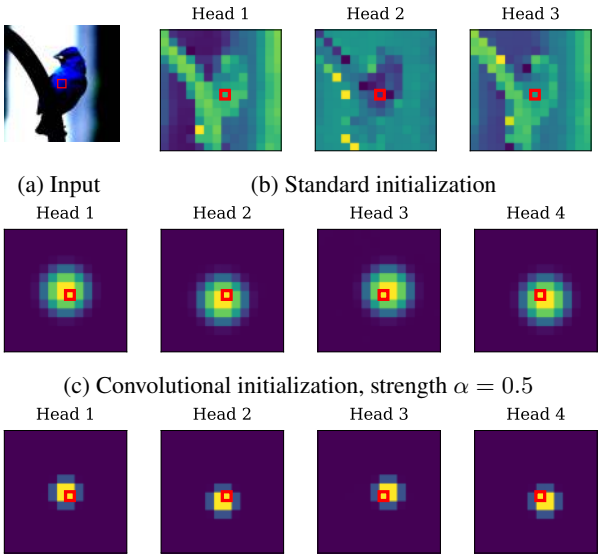
where $\boldsymbol{W}^h_{val} \in R^{D_{\mathrm{emb}} \times D_h}$ is the *value* matrix.

However, in the vanilla form of Eq. 1, SA layers are position-agnostic: they do not know how the patches are located according to each other. To incorporate positional information, there are several options. One is to add some positional information to the input at embedding time, before propagating it through the SA layers: (Dosovitskiy et al., 2020) use this approach in their ViT. Another possibility is to replace the vanilla SA with positional self-attention (PSA), using encodings $\boldsymbol{r}_{ij}$ of the relative position of patches i and j (Ramachandran et al., 2019):

$$\boldsymbol{A}^h_{ij} := \mathrm{softmax}\left(\boldsymbol{Q}^h_i \boldsymbol{K}^{h\top}_j + \boldsymbol{v}^{h\top}_{pos} \boldsymbol{r}_{ij}\right) \tag{4}$$

Each attention head uses a trainable embedding $\boldsymbol{v}^h_{pos} \in \mathbb{R}^{D_{\mathrm{pos}}}$, and the relative positional encodings $\boldsymbol{r}_{ij} \in \mathbb{R}^{D_{\mathrm{pos}}}$ only depend on the distance between pixels i and j, denoted denoted by a two-dimensional vector $\boldsymbol{\delta}_{ij}$.

Self-attention as a generalized convolution Cordonnier et al. (2019) show that a multi-head PSA layer with N_h



(a) Input (b) Standard initialization

(c) Convolutional initialization, strength $\alpha = 0.5$

(d) Convolutional initialization, strength $\alpha = 2$

Figure 3. **Positional self-attention layers can be initialized as convolutional layers.** (a): Input image from ImageNet, where the query patch is highlighted by a red box. (b),(c),(d): attention maps of an untrained SA layer (b) and those of a PSA layer using the convolutional-like initialization scheme of Eq. 5 with two different values of the locality strength parameter, α (c, d). Note that the shapes of the image can easily be distinguished in (b), but not in (c) or (d), when the attention is purely positional.

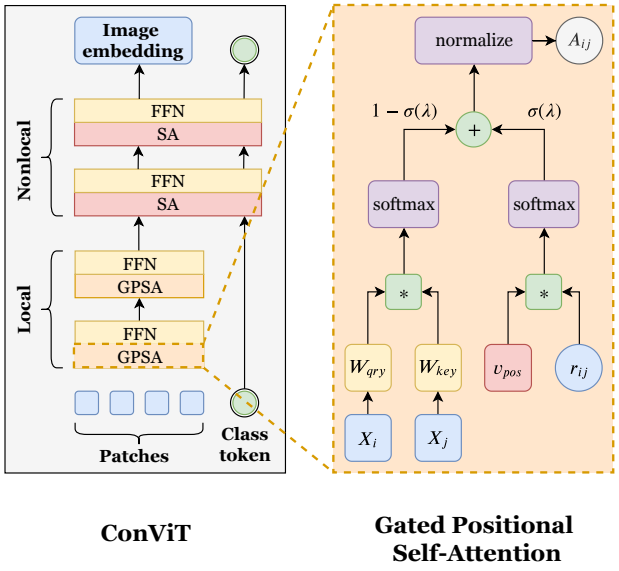
heads and learnable relative positional encodings (Eq. 4) of dimension $D_{\text{pos}} \geq 3$ can express any convolutional layer of filter size $\sqrt{N_h} \times \sqrt{N_h}$, by setting the following:

$$\begin{cases} \boldsymbol{v}_{pos}^h := -\alpha^h \left(1, -2\Delta_1^h, -2\Delta_2^h, 0, \dots 0\right) \\ \boldsymbol{r}_{\boldsymbol{\delta}} := \left(\|\boldsymbol{\delta}\|^2, \delta_1, \delta_2, 0, \dots 0\right) \\ \boldsymbol{W}_{qry} = \boldsymbol{W}_{key} := \boldsymbol{0}, \quad \boldsymbol{W}_{val} := \boldsymbol{I} \end{cases} \tag{5}$$

In the above,

- The *center of attention* $\boldsymbol{\Delta}^h \in \mathbb{R}^2$ is the position to which head h pays most attention to, relative to the query patch. For example, in Fig. 3(c), the four heads correspond, from left to right, to $\boldsymbol{\Delta}^1 = (-1, 1), \boldsymbol{\Delta}^2 = (-1, -1), \boldsymbol{\Delta}^3 = (1, 1), \boldsymbol{\Delta}^4 = (1, -1)$.
- The *locality strength* $\alpha^h > 0$ determines how focused the attention is around its center $\boldsymbol{\Delta}^h$ (it can also by understood as the "temperature" of the softmax in Eq. 1). When α^h is large, the attention is focused only on the patch(es) located at $\boldsymbol{\Delta}^h$, as in Fig. 3(d); when α^h is small, the attention is spread out into a larger area, as in Fig. 3(c).

Thus, the PSA layer can achieve a strictly convolutional attention map by setting the centers of attention $\boldsymbol{\Delta}^h$ to each of the possible positional offsets of a $\sqrt{N_h} \times \sqrt{N_h}$ convolutional kernel, and sending the locality strengths α^h to some large value.



ConViT **Gated Positional Self-Attention**

Figure 4. **Architecture of the ConViT.** The ConViT (left) is a version of the ViT in which some of the self-attention (SA) layers are replaced with gated positional self-attention layers (GPSA; right). Because GPSA layers involve positional information, the class token is concatenated with hidden representation after the last GPSA layer. In this paper, we typically take 10 GPSA layers followed by 2 vanilla SA layers. FFN: feedforward network (2 linear layers separated by a GeLU activation); W_{qry}: query weights; W_{key}: key weights; v_{pos}: attention center and span embeddings (learned); r_{qk}: relative position encodings (fixed); λ: gating parameter (learned); σ: sigmoid function.

3. Approach

Building on the insight of (Cordonnier et al., 2019), we introduce the ConVit, a variant of the ViT (Dosovitskiy et al., 2020) obtained by replacing some of the SA layers by a new type of layer which we call *gated positional self-attention* (GPSA) layers. The core idea is to enforce the "informed" convolutional configuration of Eqs. 5 in the GPSA layers at initialization, then let them decide whether to stay convolutional or not. However, the standard parameterization of PSA layers (Eq. 4) suffers from two limitations, which lead us two introduce two modifications.

Adaptive attention span The first caveat in PSA is the vast number of trainable parameters involved, since the number of relative positional encodings $\boldsymbol{r}_\delta$ is quadratic in the number of patches. This led some authors to restrict the attention to a subset of patches around the query patch (Ramachandran et al., 2019), at the cost of losing long-range information.

To avoid this, we leave the relative positional encodings $\boldsymbol{r}_\delta$ fixed, and train only the embeddings $\boldsymbol{v}_{pos}^h$ which determine the center and span of the attention heads; this approach relates to the *adaptive attention span* introduced in Sukhbaatar et al. (2019) for Language Transformers. The initial values of $\boldsymbol{r}_\delta$ and $\boldsymbol{v}_{pos}^h$ are given by Eq. 5, where we take $D_{\text{pos}} = 3$ to get rid of the useless zero components. Thanks to $D_{\text{pos}} \ll D_h$, the number of parameters involved in the positional attention is negligible compared to the number of parameters involved in the content attention. This makes sense, as content interactions are inherently much simpler to model than positional interactions.

Positional gating The second issue with standard PSA is the fact that the content and positional terms in Eq. 4 are potentially of different magnitudes, in which case the softmax will ignore the smallest of the two. In particular, the convolutional initialization scheme discussed above involves highly concentrated attention scores, i.e. high-magnitude values in the softmax. In practice, we observed that using a convolutional initialization scheme on vanilla PSA layers gives a boost in early epochs, but degrades late-time performance as the attention mechanism lazily ignores the content information (see SM. A).

To avoid this, GPSA layers sum the content and positional terms *after* the softmax, with their relative importances governed by a learnable *gating* parameter λ_h (one for each attention head). Finally, we normalize the resulting sum of matrices (whose terms are positive) to ensure that the resulting attention scores define a probability distribution. The resulting GPSA layer is therefore parametrized as follows (see also Fig. 4):

$$\text{GPSA}_h(\boldsymbol{X}) := \text{normalize}\left[\boldsymbol{A}^h\right]\boldsymbol{X}\boldsymbol{W}_{val}^h \tag{6}$$

$$\boldsymbol{A}_{ij}^h := (1-\sigma(\lambda_h))\,\text{softmax}\left(\boldsymbol{Q}_i^h\boldsymbol{K}_j^{h\top}\right) + \sigma(\lambda_h)\,\text{softmax}\left(\boldsymbol{v}_{pos}^{h\top}\boldsymbol{r}_{ij}\right), \tag{7}$$

where $\left(\text{normalize}\left[\boldsymbol{A}\right]\right)_{ij} = \boldsymbol{A}_{ij}/\sum_k \boldsymbol{A}_{ik}$ and $\sigma : x \mapsto 1/(1+e^{-x})$ is the sigmoid function. By setting the gating parameter λ_h to a large positive value at initialization, one has $\sigma(\lambda_h) \simeq 1$: the GPSA bases its attention purely on position, dispensing with the need of setting $\boldsymbol{W}_{qry}$ and $\boldsymbol{W}_{key}$ to zero as in Eq. 5. However, to avoid the ConViT staying stuck at $\lambda_h \gg 1$, we initialize $\lambda_h = 1$ for all layers and all heads.

Architectural details The ViT slices input images of size 224 into 16×16 non-overlapping patches of 14×14 pixels and embeds them into vectors of dimension $D_{\text{emb}} = 64N_h$ using a convolutional stem. It then propagates the patches through 12 blocks which keep their dimensionality constant. Each block consists in a SA layer followed by a 2-layer Feed-Forward Network (FFN) with GeLU activation, both equipped with residual connections. The ConViT is simply a ViT where the first 10 blocks replace the SA layers by GPSA layers with a convolutional initialization.

Similar to language Transformers like BERT (Devlin et al., 2018), the ViT uses an extra "class token", appended to the sequence of patches to predict the class of the input. Since this class token does not carry any positional information, the SA layers of the ViT do not use positional attention: the positional information is instead injected to each patch before the first layer, by adding a learnable positional embedding of dimension D_{emb}. As GPSA layers involve positional attention, they are not well suited for the class token approach. We solve this problem by appending the class token to the patches after the last GPSA layer, similarly to what is done in (Touvron et al., 2021b) (see Fig. 4)[1].

For fairness, and since they are computationally cheap, we keep the absolute positional embeddings of the ViT active in the ConViT. However, as shown in SM. F, the ConViT relies much less on them, since the GPSA layers already use relative positional encodings. Hence, the absolute positional embeddings could easily be removed, dispensing with the need to interpolate the embeddings when changing the input resolution (the relative positional encodings simply need to be resampled according to Eq. 5, as performed automatically in our open-source implementation).

Training details We based our ConVit on the DeiT (Touvron et al., 2020), a hyperparameter-optimized version of the ViT which has been open-sourced[2]. Thanks to its ability to achieve competitive results without using any external data, the DeiT both an excellent baseline and relatively easy to train: the largest model (DeiT-B) only requires a few days of training on 8 GPUs.

To mimic 2×2, 3×3 and 4×4 convolutional filters, we consider three different ConViT models with 4, 9 and 16 attention heads (see Tab. 1). Their number of heads are slightly larger than the DeiT-Ti, ConViT-S and ConViT-B of Touvron et al. (2020), which respectively use 3, 6 and 12 attention heads. To obtain models of similar sizes, we use two methods of comparison.

- To establish a direct comparison with Touvron et al. (2020), we lower the embedding dimension of the ConViTs to $D_{\text{emb}}/N_h = 48$ instead of 64 used for the DeiTs. Importantly, *we leave all hyperparameters (scheduling, data-augmentation, regularization) presented in (Touvron et al., 2020) unchanged* in order to

[1] We also experimented incorporating the class token as an extra patch of the image to which all heads pay attention to at initialization, but results were worse than concatenating the class token after the GPSA layers (not shown).

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

Name	Model	N_h	D_{emb}	Size	Flops	Speed	Top-1	Top-5
Ti	DeiT	3	192	6M	1G	1442	72.2	-
	ConViT	4	192	6M	1G	734	**73.1**	**91.7**
Ti+	DeiT	4	256	10M	2G	1036	75.9	93.2
	ConViT	4	256	10M	2G	625	**76.7**	**93.6**
S	DeiT	6	384	22M	4.3G	587	79.8	-
	ConViT	9	432	27M	5.4G	305	**81.3**	**95.7**
S+	DeiT	9	576	48M	10G	480	79.0	94.4
	ConViT	9	576	48M	10G	382	**82.2**	**95.9**
B	DeiT	12	768	86M	17G	187	81.8	-
	ConViT	16	768	86M	17G	141	**82.4**	**95.9**
B+	DeiT	16	1024	152M	30G	114	77.5	93.5
	ConViT	16	1024	152M	30G	96	**82.5**	**95.9**

Table 1. **Performance of the models considered, trained from scratch on ImageNet.** Speed is the number of images processed per second on a Nvidia Quadro GP100 GPU at batch size 128. Top-1 accuracy is measured on ImageNet-1k test set without distillation (see SM. B for distillation). The results for DeiT-Ti, DeiT-S and DeiT-B are reported from (Touvron et al., 2020).

Train size	**Top-1** DeiT	ConViT	Gap	**Top-5** DeiT	ConViT	Gap
5%	34.8	**47.8**	37%	57.8	**70.7**	22%
10%	48.0	**59.6**	24%	71.5	**80.3**	12%
30%	66.1	**73.7**	12%	86.0	**90.7**	5%
50%	74.6	**78.2**	5%	91.8	**93.8**	2%
100%	79.9	**81.4**	2%	95.0	**95.8**	1%

Table 2. **The convolutional inductive bias strongly improves sample efficiency.** We compare the top-1 and top-5 accuracy of our ConViT-S with that of the DeiT-S, both trained using the original hyperparameters of the DeiT (Touvron et al., 2020), as well as the relative improvement of the ConViT over the DeiT. Both models are trained on a subsampled version of ImageNet-1k, where we only keep a variable fraction (leftmost column) of the images of each class for training.

achieve a fair comparison. The resulting models are named ConViT-Ti, ConViT-S and ConViT-B.

- We also trained DeiTs and ConViTs using the same number of heads and $D_{\text{emb}}/N_h = 64$, to ensure that the improvement due to ConViT is not simply due to the larger number of heads (Touvron et al., 2021b). This leads to slightly larger models denoted with a "+" in Tab. 1. To maintain stable training while fitting these models on 8 GPUs, we lowered the learning rate from 0.0005 to 0.0004 and the batch size from 1024 to 512. These minimal hyperparameter changes lead the DeiT-B+ to perform less well than the DeiT-S+, which is not the case for the ConViT, suggesting a higher stability to hyperparameter changes.

Performance of the ConViT In Tab. 1, we display the top-1 accuracy achieved by these models evaluated on the ImageNet test set after 300 epochs of training, alongside their number of parameters, number of flops and throughput. Each ConViT outperforms its DeiT of same size and same number of flops by a margin. Importantly, although the positional self-attention does slow down the throughput of the ConViTs, they also outperform the DeiTs at equal throughput. For example, The ConViT-S+ reaches a top-1 of 82.2%, outperforming the original DeiT-B with less parameters and higher throughput. Without any tuning, the ConViT also reaches high performance on CIFAR100, see SM. C where we also report learning curves.

Note that our ConViT is compatible with the distillation methods introduced in Touvron et al. (2020) at no extra cost. As shown in SM. B, hard distillation improves performance, enabling the hard-distilled ConViT-S+ to reach 82.9% top-1 accuracy, on the same footing as the hard-distilled DeiT-B with half the number of parameters. However, while distillation requires an additional forward pass through a pre-trained CNN at each step of training, ConViT has no such requirement, providing similar benefits to distillation without additonal computational requirements.

Sample efficiency of the ConViT In Tab. 2, we investigate the sample-efficiency of the ConViT in a systematic way, by subsampling each class of the ImageNet-1k dataset by a fraction $f = \{0.05, 0.1, 0.3, 0.5, 1\}$ while multiplying the number of epochs by $1/f$ so that the total number images presented to the model remains constant. As one might expect, the top-1 accuracy of both the DeiT-S and its ConViT-S counterpart drops as f decreases. However, the ConViT suffers much less: while training on only 10% of the data, the ConVit reaches 59.5% top-1 accuracy, compared to 46.5% for its DeiT counterpart.

This result can be directly compared to (Zhai et al., 2019), which after testing several thousand convolutional models

reaches a top-1 accuracy of 56.4%; the ConViT is therefore highly competitive in terms of sample efficiency. These findings confirm our hypothesis that the convolutional inductive bias is most helpful on small datasets, as depicted in Fig. 1.

4. Investigating the role of locality

In this section, we demonstrate that locality is naturally encouraged in standard SA layers, and examine how the ConViT benefits from locality being imposed at initialization.

SA layers are pulled towards locality We begin by investigating whether the hypothesis that PSA layers are naturally encouraged to become "local" over the course of training (Cordonnier et al., 2019) holds for the vanilla SA layers used in ViTs, which do not benefit from positional attention. To quantify this, we define a measure of "nonlocality" by summing, for each query patch i, the distances $\|\boldsymbol{\delta}_{ij}\|$ to all the key patches j weighted by their attention score $\boldsymbol{A}_{ij}$. We average the number obtained over the query patch to obtain the nonlocality metric of head h, which can then be averaged over the attention heads to obtain the nonlocality of the whole layer ℓ:

$$D_{loc}^{\ell,h} := \frac{1}{L}\sum_{ij} \boldsymbol{A}_{ij}^{h,\ell}\|\boldsymbol{\delta}_{ij}\|,$$
$$D_{loc}^{\ell} := \frac{1}{N_h}\sum_{h} D_{loc}^{\ell,h} \tag{8}$$

Intuitively, D_{loc} is the number of patches between the center of attention and the query patch: the further the attention heads look from the query patch, the higher the nonlocality.

In Fig. 5 (left panel), we show how the nonlocality metric evolves during training across the 12 layers of a DeiT-S trained for 300 epochs on ImageNet. During the first few epochs, the nonlocality falls from its initial value in all layers, confirming that the DeiT becomes more "convolutional". During the later stages of training, the nonlocality metric stays low for lower layers, and gradually climbs back up for upper layers, revealing that the latter capture long range dependencies, as observed for language Transformers (Sukhbaatar et al., 2019).

These observations are particularly clear when examining the attention maps (Fig. 15 of the SM), and point to the beneficial effect of locality in lower layers. In Fig. 10 of the SM., we also show that the nonlocality metric is lower when training with distillation from a convolutional network as in Touvron et al. (2020), suggesting that the locality of the teacher is partly transferred to the student (Abnar et al., 2020).

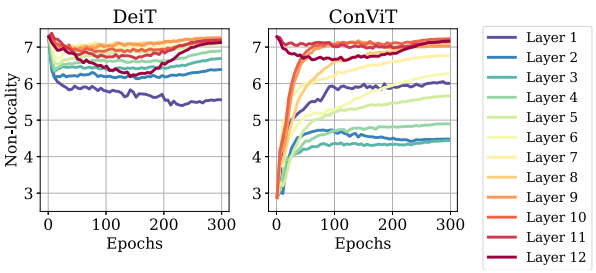


Figure 5. **SA layers try to become local, GPSA layers escape locality.** We plot the nonlocality metric defined in Eq. 8, averaged over a batch of 1024 images: the higher, the further the attention heads look from the query pixel. We trained the DeiT-S and ConViT-S for 300 epochs on ImageNet. Similar results for DeiT-Ti/ConViT-Ti and DeiT-B/ConViT-B are shown in SM. D.

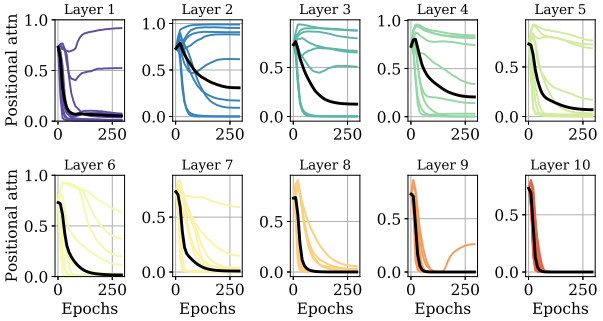


Figure 6. **The gating parameters reveal the inner workings of the ConViT.** For each layer, the colored lines (one for each of the 9 attention heads) quantify how much attention head h pays to positional information versus content, i.e. the value of $\sigma(\lambda_h)$, see Eq. 7. The black line represents the value averaged over all heads. We trained the ConViT-S for 300 epochs on ImageNet. Similar results for ConViT-Ti and ConViT-B are shown in SM D.

GPSA layers escape locality In the ConViT, strong locality is imposed at the beginning of training in the GPSA layers thanks to the convolutional initialization. In Fig. 5 (right panel), we see that this local configuration is escaped throughout training, as the nonlocality metric grows in all the GPSA layers. However, the nonlocality at the end of training is lower than that reached by the DeiT, showing that some information about the initialization is preserved throughout training. Interestingly, the final nonlocality does not increase monotonically throughout the layers as for the DeiT. The first layer and the final layers strongly escape locality, whereas the intermediate layers (particularly the second layer) stay more local.

To gain more understanding, we examine the dynamics of the gating parameters in Fig. 6. We see that in all layers, the average gating parameter $\mathbb{E}_h\sigma(\lambda_h)$ (in black), which reflects the average amount of attention paid to positional information versus content, decreases throughout training.

This quantity reaches 0 in layers 6-10, meaning that positional information is practically ignored. However, in layers 1-5, some of the attention heads keep a high value of $\sigma(\lambda_h)$, hence take advantage of positional information. Interestingly, the ConViT-Ti only uses positional information up to layer 4, whereas the ConViT-B uses it up to layer 6 (see App. D), suggesting that larger models - which are more under-specified - benefit more from the convolutional prior. These observations highlight the usefulness of the gating parameter in terms of interpretability.

The inner workings of the ConViT are further revealed by the attention maps of Fig. 7, which are obtained by propagating an embedded input image through the layers and selecting a query patch at the center of the image[3]. In layer 10, (bottom row), the attention maps of DeiT and ConViT look qualitatively similar: they both perform content-based attention. In layer 2 however (top row), the attention maps of the ConViT are more varied: some heads pay attention to content (heads 1 and 2) whereas other focus mainly on position (heads 3 and 4). Among the heads which focus on position, some stay highly localized (head 4) whereas others broaden their attention span (head 3). The interested reader can find more attention maps in SM. E.

Ref.	Train gating	Conv init	Train GPSA	Use GPSA	Full data	10% data
a (ConViT)	✓	✓	✓	✓	**82.2**	**59.7**
b	✗	✓	✓	✓	82.0	57.4
c	✓	✗	✓	✓	81.4	56.9
d	✗	✗	✓	✓	81.6	54.6
e (DeiT)	✗	✗	✗	✗	79.1	47.8
f	✗	✓	✗	✓	78.6	54.3
g	✗	✗	✗	✓	73.7	44.8

Table 3. **Gating and convolutional initialization play nicely together.** We ran an ablation study on the ConViT-S+ trained for 300 epochs on the full ImageNet training set and on 10% of the training data. From the left column to right column, we experimented freezing the gating parameters to 0, removing the convolutional initialization, freezing the GPSA layers and removing them altogether.

Strong locality is desirable We next investigate how the performance of the ConViT is affected by two important hyperparameters of the ConViT: the *locality strength*, α, which determines how focused the heads are around their center of attention, and the number of SA layers replaced by GPSA layers. We examined the effects of these hyperparameters on ConViT-S, trained on the first 100 classes of ImageNet. As shown in Fig. 8(a), final test accuracy increases both with the locality strength and with the number of GPSA layers; in other words, the more convolutional, the better.

[3]We do not show the attention paid to the class token in the SA layers

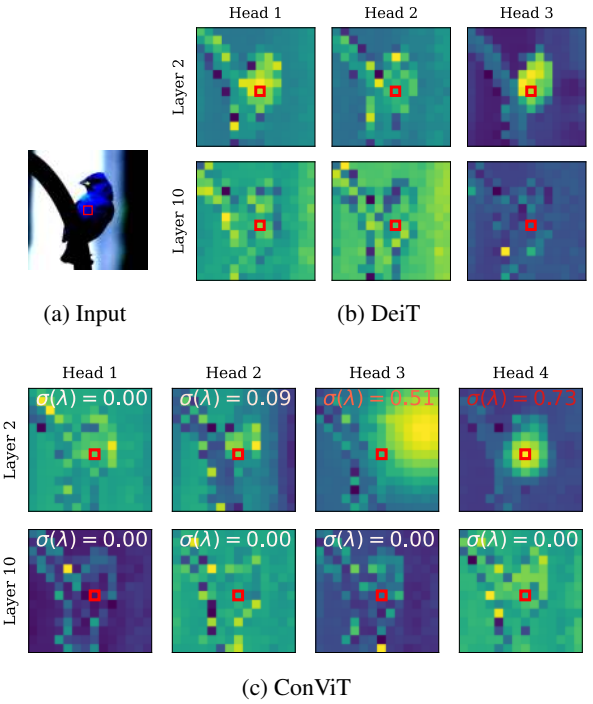


Figure 7. **The ConViT learns more diverse attention maps**. *Left:* input image which is embedded then fed into the models. The query patch is highlighted by a red box and the colormap is logarithmic to better reveal details. *Center:* attention maps obtained by a DeiT-Ti after 300 epochs of training on ImageNet. *Right:* Same for ConViT-Ti. In each map, we indicated the value of the gating parameter in a color varying from white (for heads paying attention to content) to red (for heads paying attention to position). Attention maps for more images and heads are shown in SM. E.

In Fig. 8(b), we show how performance at various stages of training is impacted by the presence of GPSA layers. We see that the boost due to GPSA is particularly strong during the early stages of training: after 20 epochs, using 9 GPSA layers leads the test-accuracy to almost double, suggesting that the convolution initialization gives the model a substantial "head start". This speedup is of practical interest in itself, on top of the boost in final performance.

Ablation study In Tab. 3, we present an ablation on the ConViT, denoted as **[a]**. We experiment removing the positional gating **[b]**[4], the convolutional initialization **[c]**, both gating and the convolutional initialization **[d]**, and the GPSA altogether (**[e]**, which leaves us with a plain DeiT).

Surprisingly, on full ImageNet, GPSA without gating **[d]** already brings a substantial benefit over the DeiT (+2.5), which is mildly increased by the convolutional initialization (**[b]**, +2.9). As for gating, it helps a little in presence

[4]To remove gating, we freeze all gating parameters to $\lambda = 0$ so that the same amount of attention is paid to content and position.

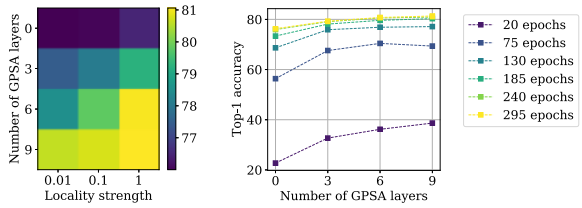


Figure 8. **The beneficial effect of locality**. *Left:* As we increase the locality strength (i.e. how focused each attention head is its associated patch) and the number of GPSA layers of a ConViT-S+, the final top-1 accuracy increases significantly. *Right:* The beneficial effect of locality is particularly strong in the early epochs.

of the convolutional initialization (**[a]**, +3.1), and is unhelpful otherwise. These mild improvements due to gating and convolutional initialization (likely due to performance saturation above 80% top-1) become much clearer in the low data regime. Here, GPSA alone brings +6.8, with an extra +2.3 coming from gating, +2.8 from convolution initialization and +5.1 with the two together, illustrating their complementarity.

We also investigated the performance of the ConViT with all GPSA layers frozen, leaving only the FFNs to be trained in the first 10 layers. As one could expect, performance is strongly degraded in the full data regime if we initialize the GPSA layers randomly (**[f]**, -5.4 compared to the DeiT). However, the convolutional initialization remarkably enables the frozen ConViT to reach a very decent performance, almost equalling that of the DeiT (**[e]**, -0.5). In other words, replacing SA layers by random "convolutions" hardly impacts performance. In the low data regime, the frozen ConViT even outperforms the DeiT by a margin (+6.5). This naturally begs the question: is attention really key to the success of ViTs (Dong et al., 2021; Tolstikhin et al., 2021; Touvron et al., 2021a)?

5. Conclusion and perspectives

The present work investigates the importance of initialization and inductive biases in learning with vision transformers. By showing that one can take advantage of convolutional constraints in a soft way, we merge the benefits of architectural priors and expressive power. The result is a simple recipe that improves trainability and sample efficiency, without increasing model size or requiring any tuning.

Our approach can be summarized as follows: instead of interleaving convolutional layers with SA layers as done in hybrid models, let the layers decide whether to be convolutional or not by adjusting a set of gating parameters. More generally, combining the biases of varied architectures and letting the model choose which ones are best for a given task could become a promising direction, reducing the need for greedy architectural search while offering higher interpretability.

Another direction which will be explored in future work is the following: if SA layers benefit from being initialized as random convolutions, could one reduce even more drastically their sample complexity by initializing them as pre-trained convolutions?

Acknowledgements We thank Hervé Jégou and Francisco Massa for helpful discussions. SD and GB acknowledge funding from the French government under management of Agence Nationale de la Recherche as part of the "Investissements d'avenir" program, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute).

References

Abnar, S., Dehghani, M., and Zuidema, W. Transferring inductive biases through knowledge distillation. *arXiv preprint arXiv:2006.00555*, 2020.

Anandkumar, A., Deng, Y., Ge, R., and Mobahi, H. Homotopy analysis for tensor pca. *arXiv preprint arXiv:1610.09322*, 2016.

Bahdanau, D., Cho, K., and Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Bello, I., Zoph, B., Vaswani, A., Shlens, J., and Le, Q. V. Attention augmented convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3286–3295, 2019.

Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., and Zagoruyko, S. End-to-end object detection with transformers. *arXiv preprint arXiv:2005.12872*, 2020.

Chen, Y., Kalantidis, Y., Li, J., Yan, S., and Feng, J. A2-nets: Double attention networks. *arXiv preprint arXiv:1810.11579*, 2018.

Chen, Y.-C., Li, L., Yu, L., El Kholy, A., Ahmed, F., Gan, Z., Cheng, Y., and Liu, J. Uniter: Universal image-text representation learning. In *European Conference on Computer Vision*, pp. 104–120. Springer, 2020.

Cordonnier, J.-B., Loukas, A., and Jaggi, M. On the relationship between self-attention and convolutional layers. *arXiv preprint arXiv:1911.03584*, 2019.

d'Ascoli, S., Sagun, L., Biroli, G., and Bruna, J. Finding the needle in the haystack with convolutions: on the benefits of architectural bias. In *Advances in Neural Information Processing Systems*, pp. 9334–9345, 2019.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dong, Y., Cordonnier, J.-B., and Loukas, A. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. *arXiv preprint arXiv:2103.03404*, 2021.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Elsayed, G., Ramachandran, P., Shlens, J., and Kornblith, S. Revisiting spatial invariance with low-rank local connectivity. In *International Conference on Machine Learning*, pp. 2868–2879. PMLR, 2020.

Gers, F. A., Schmidhuber, J., and Cummins, F. Learning to forget: Continual prediction with lstm. 1999.

Goodfellow, I., Bengio, Y., and Courville, A. *Deep Learning*. MIT Press, 2016.

Greff, K., Srivastava, R. K., Koutník, J., Steunebrink, B. R., and Schmidhuber, J. LSTM: A Search Space Odyssey. *IEEE Transactions on Neural Networks and Learning Systems*, 28(10):2222–2232, October 2017. ISSN 2162-2388. doi: 10.1109/TNNLS.2016.2582924. Conference Name: IEEE Transactions on Neural Networks and Learning Systems.

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

Hinton, G., Vinyals, O., and Dean, J. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Hu, H., Gu, J., Zhang, Z., Dai, J., and Wei, Y. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3588–3597, 2018a.

Hu, J., Shen, L., Albanie, S., Sun, G., and Vedaldi, A. Gather-Excite: Exploiting Feature Context in Convolutional Neural Networks. In Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 31*, pp. 9401–9411. Curran Associates, Inc., 2018b.

Hu, J., Shen, L., and Sun, G. Squeeze-and-Excitation Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, June 2018c. doi: 10.1109/CVPR.2018.00745. ISSN: 2575-7075.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W., and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Locatello, F., Weissenborn, D., Unterthiner, T., Mahendran, A., Heigold, G., Uszkoreit, J., Dosovitskiy, A., and Kipf, T. Object-centric learning with slot attention. *arXiv preprint arXiv:2006.15055*, 2020.

Mitchell, T. M. *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research . . . , 1980.

Neyshabur, B. Towards learning convolutions from scratch. *Advances in Neural Information Processing Systems*, 33, 2020.

Radosavovic, I., Kosaraju, R. P., Girshick, R., He, K., and Dollár, P. Designing network design spaces. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 10428–10436, 2020.

Ramachandran, P., Parmar, N., Vaswani, A., Bello, I., Levskaya, A., and Shlens, J. Stand-alone self-attention in vision models. *arXiv preprint arXiv:1906.05909*, 2019.

Ruderman, D. L. and Bialek, W. Statistics of natural images: Scaling in the woods. *Physical review letters*, 73(6):814, 1994.

Scherer, D., Müller, A., and Behnke, S. Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition. In Diamantaras, K., Duch, W., and Iliadis, L. S. (eds.), *Artificial Neural Networks – ICANN 2010*, Lecture Notes in Computer Science, pp. 92–101, Berlin, Heidelberg, 2010. Springer. ISBN 978-3-642-15825-4. doi: 10.1007/978-3-642-15825-4_10.

Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, January 2015. ISSN 0893-6080. doi: 10.1016/j.neunet.2014.09.003. URL http://www.sciencedirect.com/science/article/pii/S0893608014002135.

Simoncelli, E. P. and Olshausen, B. A. Natural image statistics and neural representation. *Annual review of neuroscience*, 24(1):1193–1216, 2001.

Srinivas, A., Lin, T.-Y., Parmar, N., Shlens, J., Abbeel, P., and Vaswani, A. Bottleneck Transformers for Visual Recognition. *arXiv e-prints*, art. arXiv:2101.11605, January 2021.

Sukhbaatar, S., Grave, E., Bojanowski, P., and Joulin, A. Adaptive attention span in transformers. *arXiv preprint arXiv:1905.07799*, 2019.

Sun, C., Myers, A., Vondrick, C., Murphy, K., and Schmid, C. Videobert: A joint model for video and language representation learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 7464–7473, 2019.

Sundermeyer, M., Schlüter, R., and Ney, H. LSTM neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*, 2012.

Tan, M. and Le, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pp. 6105–6114. PMLR, 2019.

Tolstikhin, I., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., Yung, J., Keysers, D., Uszkoreit, J., Lucic, M., et al. Mlp-mixer: An all-mlp architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., and Jégou, H. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

Touvron, H., Bojanowski, P., Caron, M., Cord, M., El-Nouby, A., Grave, E., Joulin, A., Synnaeve, G., Verbeek, J., and Jégou, H. Resmlp: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021a.

Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., and Jégou, H. Going deeper with image transformers, 2021b.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Wang, X., Girshick, R., Gupta, A., and He, K. Non-local Neural Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, Salt Lake City, UT, USA, June 2018. IEEE. ISBN 978-1-5386-6420-9. doi: 10.1109/CVPR.2018.00813. URL https://ieeexplore.ieee.org/document/8578911/.

Wu, B., Xu, C., Dai, X., Wan, A., Zhang, P., Tomizuka, M., Keutzer, K., and Vajda, P. Visual Transformers: Token-based Image Representation and Processing for Computer Vision. *arXiv:2006.03677 [cs, eess]*, July 2020. URL http://arxiv.org/abs/2006.03677. arXiv: 2006.03677.

Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Tay, F. E., Feng, J., and Yan, S. Tokens-to-token vit: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021.

Zhai, X., Oliver, A., Kolesnikov, A., and Beyer, L. S4l: Self-supervised semi-supervised learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1476–1485, 2019.

Zhao, S., Zhou, L., Wang, W., Cai, D., Lam, T. L., and Xu, Y. Splitnet: Divide and co-training. *arXiv preprint arXiv:2011.14660*, 2020.