S^2-MLPv2: Improved Spatial-Shift MLP Architecture for Vision

Tan Yu, Xu Li, Yunfeng Cai, Mingming Sun, Ping Li

Cognitive Computing Lab
Baidu Research
10900 NE 8th St. Bellevue, Washington 98004, USA
No.10 Xibeiwang East Road, Beijing 100193, China
{tanyu01, lixu13, caiyunfeng, sunmingming01, liping11}@baidu.com

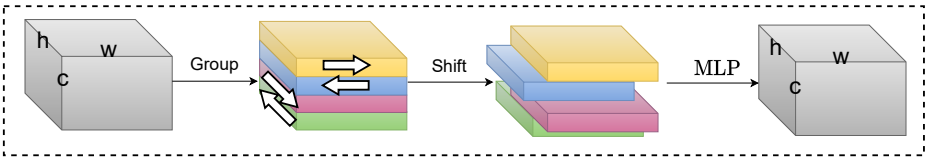
Abstract

Recently, MLP-based vision backbones emerge. MLP-based vision architectures with less inductive bias achieve competitive performance in image recognition compared with CNNs and vision Transformers. Among them, spatial-shift MLP (S^2-MLP), adopting the straightforward spatial-shift operation, achieves better performance than the pioneering works including MLP-mixer and ResMLP. More recently, using smaller patches with a pyramid structure, Vision Permutator (ViP) and Global Filter Network (GFNet) achieve better performance than S^2-MLP. In this paper, we improve the S^2-MLP vision backbone. We expand the feature map along the channel dimension and split the expanded feature map into several parts. We conduct different spatial-shift operations on split parts. Meanwhile, we exploit the split-attention operation to fuse these split parts. Moreover, like the counterparts, we adopt smaller-scale patches and use a pyramid structure for boosting the image recognition accuracy. We term the improved spatial-shift MLP vision backbone as S^2-MLPv2. Using 55M parameters, our medium-scale model, S^2-MLPv2-Medium achieves an 83.6% top-1 accuracy on the ImageNet-1K benchmark using 224×224 images without self-attention and external training data.

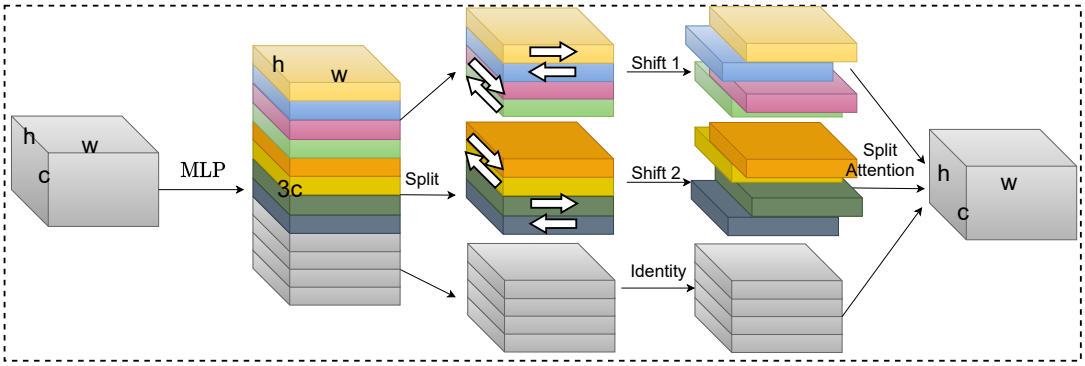
1 Introduction

Recently, extensive studies on computer vision are conducted to achieve high performance with less inductive bias. Two types of architectures emerge including vision Transformers (Dosovitskiy et al., 2021; Touvron et al., 2020) and MLP-based backbones (Tolstikhin et al., 2021; Touvron et al., 2021a). Compared with *de facto* vision backbone CNN (He et al., 2016) with delicately devised convolution kernels, both vision Transformers and MLP-based backbones have achieved competitive performance in image recognition without expensive hand-crafted design. Specifically, vision Transformer models stack a series of Transformer blocks, achieving the global reception field.

MLP-based methods such as MLP-Mixer (Tolstikhin et al., 2021) and ResMLP (Touvron et al., 2021a) achieve the communication between patches through projections along different patches implemented by MLP. Different from MLP-Mixer and ResMLP, spatial-shift MLP (S^2-MLP) (Yu et al., 2021b) adopts a very straightforward operation, spatial shifting, for communications between patches, achieving higher image recognition accuracy on ImageNet1K dataset without external training data. In parallel, Vision Permutator (ViP) (Hou et al., 2021) encodes the feature representation along the height and width dimensions and meanwhile exploits the finer patch size with a two-level pyramid structure, achieving better performance than S^2-MLP. CCS-MLP (Yu et al., 2021a) devises a circulant token-mixing MLP for achieving the translation-invariance property. Global Filter Networks (GFNet) (Rao et al., 2021b) exploits 2D Fourier Transform to map the spatial patch features into the frequency domain and conducts the cross-patch communications in the frequency domain. As pointed by Rao et al. (2021b), the token-mixing operation in the frequency domain is equivalent to depthwise convolution with circulant weights. To achieve a high recognition accuracy, GFNet also utilizes patches of smaller size with a pyramid structure. More recently, AS-MLP (Lian et al.,



(a) Spatial-shift operations in S^2-MLP



(b) Spatial-shift operations in S^2-MLPv2

Figure 1: Comparisons between the spatial-shift operations in S^2-MLP (Yu et al., 2021b) and the proposed S^2-MLPv2. In S^2-MLP, the channels are equally divided into four parts, and each part shifts along different directions. An MLP is conducted on the shifted channels. In contrast, in S^2-MLPv2, the c-channel feature map is expanded into the $3c$-channel feature map. Then the expanded map is equally split into three parts along the channel dimension. For each part, we conduct different spatial-shift operations. Then the shifted parts are merged through the split-attention operation (Zhang et al., 2020) to generate the c-channel feature map.

2021) axially shifts channels of the feature map and devises a four-level pyramid, achieving excellent performance. In parallel, Cycle-MLP (Chen et al., 2021a) devises several pseudo-kernels for spatial projection and also achieves outstanding performance. It is worth noting that, both AS-MLP (Lian et al., 2021) and Cycle-MLP (Chen et al., 2021a) are based on the well-devised four-level pyramid.

In this work, we rethink the design of spatial-shift MLP (S^2-MLP) (Yu et al., 2021b) and propose an improved spatial-shift MLP (S^2-MLPv2). Compared with the original S^2-MLP, the modifications are mainly conducted on two aspects:

- As visualized in Figure 1 (b), we expand the feature map along the channel dimension and split the expanded feature map into multiple parts. For different parts, we conduct different spatial-shift operations. We exploit the split-attention operation (Zhang et al., 2020) to fuse these split parts.
- We adopt smaller-scale patches and the hierarchical pyramid structure like existing MLP-based architectures such as ViP (Hou et al., 2021), GFNet (Rao et al., 2021b), AS-MLP (Lian et al., 2021) and Cycle-MLP (Chen et al., 2021a).

We term the improved spatial-shift MLP architecture as S^2-MLPv2. We visualize the difference between the original spatial-shift MLP (S^2-MLP) and the improved verision, S^2-MLPv2, in Figure 1. Our experiments conduct on the public benchmark, ImageNet-1K, demonstrates the state-of-the-art image recognition accuracy of the proposed S^2-MLPv2. Specifically, using 55M parameters, our medium-scale model, S^2-MLPv2-Medium achieves 83.6% top-1 accuracy using 224×224 images without self-attention and external training data.

2 Related Work

vision Transformer. vision Transformer (ViT) (Dosovitskiy et al., 2021) crops an image into 16×16 patches, and treat each patch as a token in the input of Transformer. These patches/tokens are processed by a stack of Transformer layers for communications with each other. It has achieved competitive image recognition accuracy as CNNs using huge-scale pre-training datasets. DeiT (Touvron et al., 2020) adopts more advanced optimizer as well as data augmentation methods, achieving promising results using medium-scale pre-training datasets. Pyramid vision transformer (PvT) (Wang et al., 2021b) and PiT (Heo et al., 2021) exploit a pyramid structure which gradually shrinks the spatial dimension and expands the hidden size, achieving better performance. Tokens-to-Token (T2T) (Yuan et al., 2021) and Transformer-in-Transformer (TNT) (Han et al., 2021) improve the effectiveness of modeling the local structure of each patch/token. To overcome the inefficiency of the global self-attention, Swin (Liu et al., 2021b) conducts the self-attention within local windows but achieves the global reception field through shifting the window settings. Shuffle Transformer (Huang et al., 2021) also exploits the local self-attention windows and achieves the cross-window communications through switching the spatial dimension and the feature dimension. Twins (Chu et al., 2021a) enhances the self-attention within local windows by the global sub-sampled attention. DynamicViT (Rao et al., 2021a) and SViTE (Chen et al., 2021b) exploit the sparsity for achieving high efficiency. CaiT (Touvron et al., 2021b) explores the extremely deep architecture by stacking tens of layers. Recently, PVTv2 (Wang et al., 2021a) improves PvT using overlapping patch embedding, convolutional feedforward networks, and linear-complexity attention layers. CSwin Transformer (Dong et al., 2021) improves Swin through cross-shaped windows computing self-attention in the horizontal and vertical stripes in parallel. Focal Transformer (Yang et al., 2021) also develops more advanced local windows which attend fine-grain tokens only locally, but the summarized ones globally.

MLP-based architectures. MLP-mixer (Tolstikhin et al., 2021) is the pioneering work for MLP-based vision backbone. It proposes a token-mixing MLP consisting of two fully-connected layers for communications between patches. Res-MLP (Touvron et al., 2021a) simplifies the token-mixing MLP to a single fully-connected layer and explores the deeper architecture with more layers. Spatial-shift MLP backbone (S^2-MLP) (Yu et al., 2021b) adopts the spatial-shift operation for cross-patch communications. Vision Permutator (ViP) (Hou et al., 2021) mixes tokens along the height dimension and the width dimension, separately. Meanwhile, ViP adopts a pyramid structure as PvT (Wang et al., 2021b) and achieves considerably better performance than MLP-mixer, Res-MLP and S^2-MLP. CCS-MLP (Yu et al., 2021a) rethinks the design of token-mixing MLP in MLP-mixer and Res-MLP, and propose a circulant channel-specific MLP. Specifically, they devise the weight matrix of token-mixing MLP as a circulant matrix, taking fewer parameters. Meanwhile, the multiplication between vector and circulant matrix can be efficiently computed through Fast Fourier Transform (FFT). Global Filter Network (GFNet) (Rao et al., 2021b) maps the patch features to the frequency domain through 2D FFT and mixes the tokens in the frequency domain. As proved by Rao et al. (2021b), the global filter in GFNet is equivalent to a depthwise global circular convolution with the filter size H $\times$ W. Meanwhile, GFNet also exploits pyramid structure for boosting the recognition accuracy. In this work, we rethink the design of S^2-MLP and considerably improves its performance in image recognition.

3 Preliminary

3.1 Spatial-shift MLP (S^2-MLP)

In this section, we briefly review the structure of S^2-MLP (Yu et al., 2021b) architecture. It consists of the patch embedding layer, a stack of S^2-MLP blocks and the classification head.

Patch embedding layer. It first crops an image of $W \times H \times 3$ size into $w \times h$ patches. Each patch is of $p \times p \times 3$ size and $p = \frac{W}{w} = \frac{H}{h}$. It then maps each patch into a d-dimensional vector through a fully-connected layer.

Spatial-shift MLP block. As visualized in Figure 2, it consists of four MLP layers for mixing channels and a spatial shift layer for mixing patches. Below we only introduce the spatial-shift module. Given an input tensor $\mathcal{X} \in \mathbb{R}^{w \times h \times c}$, it first equally splits $\boldsymbol{\mathcal{X}}$ into four parts $\{\boldsymbol{\mathcal{X}}_i\}_{i=1}^4$ along

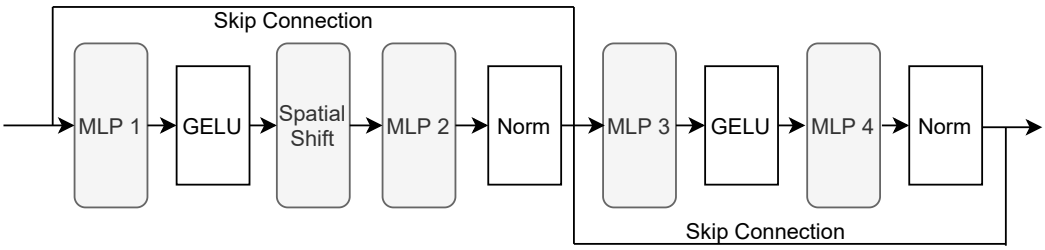


Figure 2: The structure of an S^2-MLP block.

the channel dimension and shifts them along four directions:

$$\begin{aligned}
&\boldsymbol{\mathcal{X}}[2:h,:,1:c/4] \leftarrow \boldsymbol{\mathcal{X}}[1:h-1,:,1:c/4],\\
&\boldsymbol{\mathcal{X}}[1:h-1,:,c/4+1:c/2] \leftarrow \boldsymbol{\mathcal{X}}[2:h,:,c/4+1:c/2],\\
&\boldsymbol{\mathcal{X}}[:,2:w,c/2:3c/4] \leftarrow \boldsymbol{\mathcal{X}}[:,1:w-1,c/2:3c/4],\\
&\boldsymbol{\mathcal{X}}[:,1:w-1,3c/4:c] \leftarrow \boldsymbol{\mathcal{X}}[:,2:w,3c/4:c].
\end{aligned} \tag{1}$$

It is worth noting that, S^2-MLP (Yu et al., 2021b) stacks N Spatial-shift MLP blocks with the same settings and does not exploit pyramid structure as its MLP-backone counterparts such as Vision Permutator (Hou et al., 2021) and Global Filter Network (GFNet) (Rao et al., 2021b).

3.2 Split Attention

Vision Permutator (Hou et al., 2021) adopts split attention proposed in ResNeSt (Zhang et al., 2020) for enhancing multiple feature maps from different operations. Specifically, we denote K features maps of the same size $n \times c$ by $[\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_K]$ where n is the number of patches and c is the number of channels, the split-attention operation first averages them and obtains

$$\mathbf{a} = \sum_{k=1}^{K} \mathbf{1}\mathbf{X}_k, \tag{2}$$

where $\mathbf{1} \in \mathbb{R}^n$ is the n-dimensional row vector with all 1s. Then $\mathbf{a} \in \mathbb{R}^c$ goes through a stack of MLPs and generates

$$\hat{\mathbf{a}} = \sigma(\mathbf{a}\mathbf{W}_1)\mathbf{W}_2, \tag{3}$$

where σ is the activation function implemented by GELU, $\mathbf{W}_1 \in \mathbb{R}^{c \times \bar{c}}$ and $\mathbf{W}_2 \in \mathbb{R}^{\bar{c} \times Kc}$ are weights of MLPs and the output $\hat{\mathbf{a}} \in \mathbb{R}^{Kc}$. Then $\hat{\mathbf{A}}$ is reshaped into a matrix $\hat{\mathbf{A}} \in \mathbb{R}^{K \times c}$, which is further processed by a softmax function along the first dimension and generates $\bar{\mathbf{A}} = \mathrm{softmax}(\hat{\mathbf{A}}) \in \mathbb{R}^{K \times c}$. Then it generates the attended feature map $\hat{\mathbf{X}}$ where each row of $\hat{\mathbf{X}}$, $\hat{\mathbf{X}}[i,:]$, is computed by

$$\hat{\mathbf{X}}[i,:] = \sum_{k=1}^{K} \mathbf{X}_k[i,:] \odot \bar{\mathbf{A}}[k,:], \tag{4}$$

where $\odot$ denotes the element-wise multiplication between two vectors.

4 S^2-MLPv2

In this section, we introduce the proposed S^2-MLPv2 architecture. Similar to S^2-MLP backbone, S^2-MLPv2 backbone consists of the patch embedding layer, a stack of S^2-MLPv2 blocks and the classification head. Since we have introduced the patch embedding layer in the previous section, we only introduce the proposed S^2-MLPv2 block below.

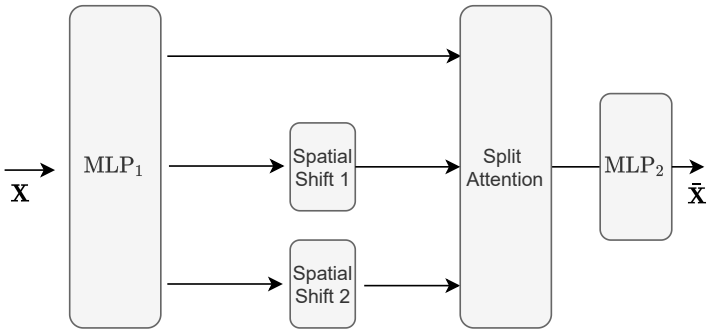


Figure 3: The structure of an S^2-MLPv2 block.

4.1 S^{2}V2 BLOCK

The S^2-MLPv2 block consists of two parts, the S^2-MLPv2 component and the channel-mixing MLP (CM-MLP) component. Given an input feature map $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{w \times h \times c}$, it conducts

$$\begin{aligned} \boldsymbol{\mathcal{Y}} &= \text{S}^2\text{-MLPv2}(\text{LN}(\boldsymbol{\mathcal{X}})) + \boldsymbol{\mathcal{X}}, \\ \boldsymbol{\mathcal{Z}} &= \text{CM-MLP}(\text{LN}(\boldsymbol{\mathcal{Y}})) + \boldsymbol{\mathcal{Y}}. \end{aligned} \tag{5}$$

The channel-mixing MLP (CM-MLP) adopts the same structure as MLP-mixer (Tolstikhin et al., 2021) and ResMLP (Touvron et al., 2021a) and thus we skip their details here. Below we only introduce the proposed S^2-MLPv2 component in detail.

Given an input feature map $\boldsymbol{\mathcal{X}} \in \mathbb{R}^{w \times h \times c}$, the proposed S^2-MLPv2 component first expands the channels of $\boldsymbol{\mathcal{X}}$ from c to $3c$ by an MLP:

$$\hat{\boldsymbol{\mathcal{X}}} = \text{MLP}_1(\boldsymbol{\mathcal{X}}) \in \mathbb{R}^{w \times h \times 3c}. \tag{6}$$

Then it equally splits the expanded feature map $\hat{\boldsymbol{\mathcal{X}}}$ along the channel dimension into three parts:

$$\boldsymbol{\mathcal{X}}_1 = \hat{\boldsymbol{\mathcal{X}}}[:, :, 1:c], \boldsymbol{\mathcal{X}}_2 = \hat{\boldsymbol{\mathcal{X}}}[:, :, c+1:2c], \boldsymbol{\mathcal{X}}_3 = \hat{\boldsymbol{\mathcal{X}}}[:, :, 2c+1:3c]. \tag{7}$$

It shifts $\boldsymbol{\mathcal{X}}_1$ and $\boldsymbol{\mathcal{X}}_2$ through two two spatial-shift layers $\text{SS}_1(\cdot)$ and $\text{SS}_2(\cdot)$. $\text{SS}_1(\cdot)$ conducts the same spatial-shift operation as equation 1. In contrast, $\text{SS}_2(\cdot)$ conducts an asymmetric spatial-shift operation with respect to $\text{SS}_1(\cdot)$. To be specific, given the feature map $\boldsymbol{\mathcal{X}}_2$, $\text{SS}_2(\boldsymbol{\mathcal{X}}_2)$ conducts:

$$\begin{aligned} &\boldsymbol{\mathcal{X}}_2[:, 2:w, 1:c/4] \leftarrow \boldsymbol{\mathcal{X}}_2[:, 1:w-1, 1:c/4], \\ &\boldsymbol{\mathcal{X}}_2[:, 1:w-1, c/4+1:c/2] \leftarrow \boldsymbol{\mathcal{X}}_2[:, 2:w, c/4+1:c/2], \\ &\boldsymbol{\mathcal{X}}_2[2:h, :, c/2:3c/4] \leftarrow \boldsymbol{\mathcal{X}}_2[1:h-1, :, c/2:3c/4], \\ &\boldsymbol{\mathcal{X}}_2[1:h-1, :, 3c/4:c] \leftarrow \boldsymbol{\mathcal{X}}_2[2:h, :, 3c/4:c]. \end{aligned} \tag{8}$$

Note that we intentionally devise $\text{SS}_1(\cdot)$ and $\text{SS}_2(\cdot)$ in an asymmetric structure so that they are complementary to each other. Meanwhile, we do not shift $\boldsymbol{\mathcal{X}}_3$ and just keep it as well.

After that, $\{\boldsymbol{\mathcal{X}}_k\}_{k=1}^3$ are reshape into matrices $\{\mathbf{X}_k\}_{k=1}^3$ where $\mathbf{X}_k \in \mathbb{R}^{wh \times c}$, which are fed into a split-attention (SA) module as equation 2, equation 3, and equation 4, to generate

$$\hat{\mathbf{X}} = \text{SA}(\{\mathbf{X}_k\}_{k=1}^3). \tag{9}$$

Then the attended feature map $\mathbf{A}$ is further fed into another MLP layer for generating the output

$$\bar{\mathbf{X}} = \text{MLP}_2(\hat{\mathbf{X}}). \tag{10}$$

The structure of the proposed S^2-MLP module is visualized in Figure 3 and the details are listed in Algorithm 1.

Algorithm 1 Pseudocode of our S^2-MLPv2 module.

```
def spatial_shift1(x):
    b,w,h,c = x.size()
    x[:,1:,:,:c/4] = x[:,:w-1,:,:c/4]
    x[:,:w-1,:,c/4:c/2] = x[:,1:,:,c/4:c/2]
    x[:,:,1:,c/2:c*3/4] = x[:,:,:h-1,c/2:c*3/4]
    x[:,:,:h-1,3*c/4:] = x[:,:,1:,3*c/4:]
    return x

def spatial_shift2(x):
    b,w,h,c = x.size()
    x[:,:,1:,:c/4] = x[:,:,:h-1,:c/4]
    x[:,:,:h-1,c/4:c/2] = x[:,:,1:,c/4:c/2]
    x[:,1:,:,c/2:c*3/4] = x[:,:w-1,:,c/2:c*3/4]
    x[:,:w-1,:,3*c/4:] = x[:,1:,:,3*c/4:]
    return x

class S2-MLPv2(nn.Module):
   def __init__(self, channels):
      super().__init__()
      self.mlp1 = nn.Linear(channels,channels*3)
      self.mlp2 = nn.Linear(channels,channels)
      self.split_attention = SplitAttention()
   def forward(self, x):
      b,w,h,c = x.size()
      x = self.mlp1(x)
      x1 = spatial_shift1(x[:,:,:,:c/3])
      x2 = spatial_shift2(x[:,:,:,c/3:c/3*2])
      x3 = x[:,:,:,c/3*2:]
      a = self.split_attention(x1,x2,x3)
      x = self.mlp2(a)
      return x
```

4.2 Pyramid structure

Following vision Permutator (Hou et al., 2021), we also exploit the two-level pyramid structure to enhance the performance. To make a fair comparison with Vision Permutator (Hou et al., 2021), we adopt the exact same pyramid structure. The details are in Table 1. We notice that counterpart works such as PVTv2 (Wang et al., 2021a), AS-MLP (Lian et al., 2021), and Cycle-MLP (Chen et al., 2021a) adopt more advanced pyramid structure with smaller patches in the early blocks. The smaller patches might be better at capturing the fine-grained visual details and lead to higher recognition accuracy. Nevertheless, due to the limited computing resources, it is unfeasible for us to re-implement all these pyramid structures. Moreover, we also notice that Vision Permutator devises a large model with considerably more parameters and FLOPs. Nevertheless, due to the limited computing resources, the large model is not feasible for us, either.

Settings	Patch Size	# of Tokens	Hidden Size	# of Blocks	Patch Size	# of Tokens	Hidden Size	# of Blocks	Expa. Ratio
Small/7	7×7	32^2	192	4	2×2	16^2	384	14	3
Medium/7	7×7	32^2	256	7	2×2	16^2	512	17	3

Table 1: The configurations of the two-level pyramid structure used in our S^2-MLPv2. We exploit both small and medium settings, which are the exactly same as Vision Permutator (Hou et al., 2021) fo a fair comparison. AS-MLP (Lian et al., 2021) and Cycle-MLP (Chen et al., 2021a) adopt more advanced four-level pyramid structure with patches of the smaller scale.

5 EXPERIMENTS

We testify the proposed S^2-MLPv2 on ImageNet-1K dataset (Deng et al., 2009). We do not use external data for training. The implementation is based on the PaddlePaddle deep learning platform.

Implementation details. Following DeiT (Touvron et al., 2020), we adopt AdamW (Loshchilov & Hutter, 2019) as optimizer. We train both the small model and the medium model using four NVIDIA A100 GPU cards. For the small model, we set the batch size as 1024. In contrast, for the medium model, we only set the batch size as 744 due to the GPU memory limitation of four NVIDIA A100 GPU cards. We set the initial learning rate as 2e-3 and decay it to 1e-5 in 300 epochs using a cosine function. The weight decay rate is set to be 5e-2 following previous works (Touvron et al., 2020; Hou et al., 2021). We also conduct warming up in the first 10 epochs following Hou et al. (2021). Moreover, as adopted by Touvron et al. (2020); Hou et al. (2021), we conduct multiple data augmentation methods including Rand-Augment (Cubuk et al., 2020), Mixup (Zhang et al., 2018) and CutMix (Yun et al., 2019) and CutOut (Zhong et al., 2020). Like DeiT (Touvron et al., 2020) and Vision Permutator (Hou et al., 2021), we adopt exponential moving average (EMA) model (Laine & Aila, 2016). Meanwhile, we also use label smoothing (Szegedy et al., 2016) with a smooth ratio of 0.1 and DropPath (Huang et al., 2016) with a drop ratio of 0.1 for both small and medium settings.

5.1 COMPARISONS WITH STATE-OF-THE-ART METHODS

Model	Pyramid	Para.	FLOPs	Train Size	Test Size	Top-1 Acc. (%)
Small models						
EAMLP-14 (Guo et al., 2021)		30M	–	224	224	78.9
ResMLP-S24 (Touvron et al., 2021a)		30M	6.0B	224	224	79.4
gMLP-S (Liu et al., 2021a)		20M	4.5B	224	224	79.6
GFNet-S (Rao et al., 2021b)		25M	4.5B	224	224	80.0
GFNet-H-S (Rao et al., 2021b)	✓	32M	4.5B	224	224	81.5
AS-MLP-T (Lian et al., 2021)	✓	28M	4.4B	224	224	81.3
CycleMLP-B2 (Chen et al., 2021a)	✓	27M	3.9B	224	224	81.6
ViP-Small/7 (Hou et al., 2021)	✓	25M	6.9B	224	224	81.5
S^2-MLPv2-Small/7 (ours)	✓	25M	6.9B	224	224	**82.0**
Medium models						
MLP-mixer (Tolstikhin et al., 2021)		59M	11.6B	224	224	76.4
EAMLP-19 (Guo et al., 2021)		55M	–	224	224	79.4
S^2-MLP-deep (Yu et al., 2021b)		51M	10.5B	224	224	80.7
CCS-MLP-36 (Yu et al., 2021a)		43M	8.9B	224	224	80.6
GFNet-B (Rao et al., 2021b)		43M	7.9G	224	224	80.7
GFNet-H-B (Rao et al., 2021b)	✓	54M	8.4B	224	224	82.9
AS-MLP-S (Lian et al., 2021)	✓	50M	8.5B	224	224	83.1
CycleMLP-B4 (Chen et al., 2021a)	✓	52M	10.1B	224	224	83.0
ViP-Medium/7 (Hou et al., 2021)	✓	55M	16.3B	224	224	82.7
S^2-MLPv2-Medium/7 (ours)	✓	55M	16.3B	224	224	**83.6**
Large models						
CycleMLP-B5 (Chen et al., 2021a)	✓	76M	12.3B	224	224	83.2
AS-MLP-B (Lian et al., 2021)	✓	88M	15.2B	224	224	83.3
ViP-Large/7 (Hou et al., 2021)	✓	88M	24.3B	224	224	83.2

Table 2: Comparisons with MLP-like backbones on ImageNet-1K benchmark without extra data. Our S^2-MLPv2-Medium/7 achieves the state-of-the-art performance on the benchmark among medium-scale MLP models and even outperforms the existing large-scale MLP models. M denotes million and B denotes billion.

Comparisons with existing MLP-like methods. In Table 2, we compare our S^2-MLPv2 with existing MLP-like backbones including MLP-Mixer (Tolstikhin et al., 2021), EAMLP (Guo et al., 2021), ResMLP (Touvron et al., 2021a), gMLP (Liu et al., 2021a), S2-MLP-deep (Yu et al., 2021b), CCS-MLP (Yu et al., 2021a), GFNet (Rao et al., 2021b), AS-MLP (Lian et al., 2021), CycleMLP (Chen

et al., 2021a) and ViP (Hou et al., 2021) on both small and base settings. Among them, MLP-Mixer, ResMLP, gMLP, S2-MLP, CCS-MLP do not exploit the pyramid structure, and thus they cannot achieve competitive recognition accuracy compared with GFNet, AS-MLP, CycleMLP, and ViP, which are with the pyramid structure as shown in Table 2. Meanwhile, as shown in the table, our S^2-MLPv2 consistently outperforms its counterparts in both small and medium settings using a comparable number of parameters. Meanwhile, our medium model performs even better than the large models of AS-MLP (Lian et al., 2021), CycleMLP (Chen et al., 2021a) and ViP (Hou et al., 2021) with considerably more parameters. On the other hand, we notice that both S^2-MLPv2 and ViP take more FLOPs compared with GFNet, AS-MLP, CycleMLP. This is due to that ours and ViP use a very coarse pyramid structure, whereas GFNet, AS-MLP, CycleMLP use a more advanced pyramid. Both S^2-MLPv2 and ViP might potentially reduce FLOPs by using a well-devised pyramid structure like GFNet, AS-MLP, CycleMLP.

Comparisons with CNNs and vision Transformers. We compare the proposed S^2-MLPv2 with CNN models including ResNet50 (He et al., 2016), RegNet (Radosavovic et al., 2020) and EfficientNet (Tan & Le, 2019). Meanwhile, we also compare with vision Transformer models including ViT (Dosovitskiy et al., 2021), DeiT (Touvron et al., 2020), PVT (Wang et al., 2021b), T2T (Yuan et al., 2021), TNT (Han et al., 2021), PiT (Heo et al., 2021), MViT (Fan et al., 2021) CaiT (Touvron et al., 2021b), Swin (Liu et al., 2021b), Shuffle Transformer (Huang et al., 2021), Nest Transformer (Zhang et al., 2021), Focal Transformer (Yang et al., 2021), and CSWin (Dong et al., 2021).

As shown in Table 3, our S^2-MLPv2-Medium achieves comparable accuracy as its vision Transformer counterparts using fewer parameters but more FLOPs. Using a more advanced pyramid as PvTv2, the FLOPs of our S^2-MLPv2-Medium might be reduced. Noting that, compared with vision Transformer requiring complex self-attention operations, ours is much simpler in formulation and takes consideraly fewer parameters, making it a competitive choice in practical deployment.

Model	Scale	Top-1 (%)	Params (M)	FLOPs (B)
CNN-based models				
ResNet50 (He et al., 2016)	224	76.2	25.6	4.1
RegNetY-16GF (Radosavovic et al., 2020)	224	80.4	83.6	15.9
EfficientNet-B3 (Tan & Le, 2019)	300	81.6	12	1.8
EfficientNet-B5 (Tan & Le, 2019)	456	84.0	30	9.9
Transformer-based models				
ViT-B/16* (Dosovitskiy et al., 2021)	224	79.7	86.4	17.6
DeiT-B/16 (Touvron et al., 2020)	224	81.8	86.4	17.6
PVT-L (Wang et al., 2021b)	224	82.3	61.4	9.8
TNT-B (Han et al., 2021)	224	82.8	65.6	14.1
T2T-24 (Yuan et al., 2021)	224	82.6	65.1	15.0
CPVT-B (Chu et al., 2021b)	224	82.3	88	17.6
PiT-B/16 (Heo et al., 2021)	224	82.0	73.8	12.5
MViT-B-24 (Fan et al., 2021)	224	83.1	53.5	10.9
CaiT-S32 (Touvron et al., 2021b)	224	83.3	68	13.9
Swin-B (Liu et al., 2021b)	224	83.3	88	15.4
Shuffle-B (Huang et al., 2021)	224	84.0	88	15.6
Nest-B (Zhang et al., 2021)	224	83.8	68	17.9
PvTv2-B4 (Wang et al., 2021a)	224	83.6	62.6	10.1
Focal-Base (Yang et al., 2021)	224	83.8	89.8	16.0
CSWin-B (Dong et al., 2021)	224	84.2	78	15.0
Our models				
S^2-MLPv2-Small/7	224	82.0	25	6.9
S^2-MLPv2-Medium/7	224	83.6	55	16.3

Table 3: Comparisons with CNN and Transformer models on ImageNet-1K without extra data. ViT-B/16* denotes the result of ViT-B/16 reported by Tolstikhin et al. (2021) with extra regularization. Compared with Transformer-based models, our S^2-MLPv2-Medium/7 model achieves comparable recognition accuracy on the benchmark without self-attention and considerably fewer parameters.

5.2 ABLATION STUDIES

Influence of the pyramid structure. To evaluate the influence of the pyramid structure on the proposed S^2-MLPv2, we compare the Small/7 settings and the Small/14 settings. The details of Small/7 settings and the Small/14 settings are in Table 4. Both of them are the same as that in Vision Permutator (Hou et al., 2021). Specifically, the initial patch size in Small/7 is 7×7, which is smaller than the 14×14 patches in the Small/14 settings. Intuitively, the smaller patches are beneficial to modeling fine-grained details in the images and tend to achieve higher recognition accuracy. Table 5 compares the performance of these two settings. As shown in the table, by utilizing the pyramid, S^2-MLPv2-Small/7 achieves a considerably better performance than S^2-MLPv2-Small/14.

Settings	Patch Size	# of Tokens	Hidden Size	# of Blocks	Patch Size	# of Tokens	Hidden Size	# of Blocks	Expa. Ratio
Small/7	7×7	32^2	192	4	2×2	16^2	384	14	3
Small/14	14×14	16^2	384	4	2×2	16^2	384	14	3

Table 4: The configurations of the Small/7 settings with the pyramid structure and Small/14 without the pyramid structure.

Settings	Pyramid	Top-1 (%)	# of parameters	FLOPs
S^2-MLPv2-Small/7	✓	82.0	25M	6.9B
S^2-MLPv2-Small/14		80.9	30M	5.7B

Table 5: Comparisons between Small/7 and Small/14 settings.

Influence of the split attention. Recall from equation 9 that, we use the split-attention (SA) for fusing the feature maps $\mathbf{A} = \mathrm{SA}(\{\mathbf{X}_k\}_{k=1}^3)$. An alternating fusing manner is sum-pooling them implemented by $\mathbf{A} = \sum_{k=1}^3 \mathbf{X}_k/3$. We compare these two manners in Table 6. The experiments are conducted in Small/7 settings. As shown in Table 6, the split attention significantly outperforms sum pooling with a slight increase in the number of parameters and FLOPs.

Settings	Top-1 (%)	# of parameters	FLOPs
Sum-pooling	79.8	22M	6.9B
Split-attention	82.0	25M	6.9B

Table 6: Performance comparisons between split attention and sum pooling. The experiments are conducted in Small/7 settings.

Influence of each split. As equation 9, we fuse three splits $\{\mathbf{X}_k\}_{k=1}^3$ through the split attention. In this section, we evaluate the influence of removing one of them. The experiments are conducted in Small/7 settings. As shown in Table 7, when using only $\mathbf{X}_1$ and $\mathbf{X}_3$, the top-1 accuracy drops from 82.0% to 81.6%. Meanwhile, when removing $\mathbf{X}_3$, the top-1 accuracy decreases to 81.6%.

$\mathbf{X}_1$	$\mathbf{X}_2$	$\mathbf{X}_3$	Top-1 (%)	# of parameters	FLOPs
✓	✓	✓	82.0	25M	6.9B
✓		✓	81.6	22M	6.2B
✓	✓		81.7	22M	6.2B

Table 7: The influence of each split. The experiments are conducted in Small/7 settings.

6 CONCLUSION

In this paper, we improve the spatial-shift MLP (S^2-MLP) and propose an S^2-MLPv2 model. It expands the feature map and splits the expanded feature map into three splits. It shifts each split individually and then fuses the split feature maps through split-attention. Meanwhile, we exploit the hierarchical pyramid to improve its capability of modeling fine-grained details for higher recognition accuracy. Using 55M parameters, our S^2-MLPv2-Medium model achieves 83.6% top-1 accuracy on ImageNet1K dataset using 224×224 images without external training datasets, which is the state-of-the-art performance among MLP-based methods. Meanwhile, compared with Transformer-based methods, our S^2-MLPv2 model has achieved comparable accuracy without self-attention and fewer parameters.

Compared with the pioneering MLP-based works such as MLP-mixer, ResMLP as well as recent MLP-like models including Vision Permutator and GFNet, another important advantage of the spatial-shift MLP is that the shapes of spatial-shift MLPs are invariant to the input scale of images. Thus, the spatial-shift MLP model pre-trained by images of a specific scale can be well adopted for down-stream tasks with various-sized input images.

The future work will be devoted to continuously improving the image recognition accuracy of the spatial-shift MLP architecture. A promising and straightforward direction is to attempt smaller-size patches and the more advanced four-level pyramid as CycleMLP and AS-MLP for further reducing the FLOPs and shortening the recognition gap between the Transformer-based models.

REFERENCES

Shoufa Chen, Enze Xie, Chongjian Ge, Ding Liang, and Ping Luo. Cyclemlp: A mlp-like architecture for dense prediction. *arXiv preprint arXiv:2107.10224*, 2021a.

Tianlong Chen, Yu Cheng, Zhe Gan, Lu Yuan, Lei Zhang, and Zhangyang Wang. Chasing sparsity in vision transformers: An end-to-end exploration. *arXiv preprint arXiv:2106.04533*, 2021b.

Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *arXiv preprint arXiv:2104.13840*, 2021a.

Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021b.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 702–703, 2020.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *Proceedings of the 2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 248–255, Miami, FL, 2009.

Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. *arXiv preprint arXiv:2107.00652*, 2021.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *Proceedings of the 9th International Conference on Learning Representations (ICLR)*, Virtual Event, 2021.

Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. *arXiv preprint arXiv:2104.11227*, 2021.

Meng-Hao Guo, Zheng-Ning Liu, Tai-Jiang Mu, and Shi-Min Hu. Beyond self-attention: External attention using two linear layers for visual tasks. *arXiv preprint arXiv:2105.02358*, 2021.

Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *arXiv preprint arXiv:2103.00112*, 2021.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, Las Vegas, NV, 2016.

Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. *arXiv: 2103.16302*, 2021.

Qibin Hou, Zihang Jiang, Li Yuan, Ming-Ming Cheng, Shuicheng Yan, and Jiashi Feng. Vision permutator: A permutable mlp-like architecture for visual recognition. *arXiv preprint arXiv:2106.12368*, 2021.

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

Zilong Huang, Youcheng Ben, Guozhong Luo, Pei Cheng, Gang Yu, and Bin Fu. Shuffle transformer: Rethinking spatial shuffle for vision transformer. *arXiv preprint arXiv:2106.03650*, 2021.

Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

Dongze Lian, Zehao Yu, Xing Sun, and Shenghua Gao. As-mlp: An axial shifted mlp architecture for vision. *arXiv preprint arXiv:2107.08391*, 2021.

Hanxiao Liu, Zihang Dai, David R So, and Quoc V Le. Pay attention to mlps. *arXiv preprint arXiv:2105.08050*, 2021a.

Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021b.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *Proceedings of the 7th International Conference on Learning Representations (ICLR)*, New Orleans, LA, 2019.

Ilija Radosavovic, Raj Prateek Kosaraju, Ross B. Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10425–10433, Seattle, WA, 2020.

Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. Dynamicvit: Efficient vision transformers with dynamic token sparsification. *arXiv preprint arXiv:2106.02034*, 2021a.

Yongming Rao, Wenliang Zhao, Zheng Zhu, Jiwen Lu, and Jie Zhou. Global filter networks for image classification. *arXiv preprint arXiv:2107.00645*, 2021b.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, Las Vegas, NV, 2016.

Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, pp. 6105–6114, Long Beach, CA, 2019.

Ilya Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, Mario Lucic, and Alexey Dosovitskiy. MLP-Mixer: An all-MLP architecture for vision. *arXiv preprint arXiv:2105.01601*, 2021.

Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020.

Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. ResMLP: Feedforward networks for image classification with data-efficient training. *arXiv preprint arXiv:2105.03404*, 2021a.

Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. *arXiv preprint arXiv:2103.17239*, 2021b.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *arXiv preprint arXiv:2106.13797*, 2021a.

Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021b.

Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal self-attention for local-global interactions in vision transformers. *arXiv preprint arXiv:2107.00641*, 2021.

Tan Yu, Xu Li, Yunfeng Cai, Mingming Sun, and Ping Li. Rethinking token-mixing mlp for mlp-based vision backbone. *arXiv preprint arXiv:2106.14882*, 2021a.

Tan Yu, Xu Li, Yunfeng Cai, Mingming Sun, and Ping Li. S2-MLP: Spatial-shift mlp architecture for vision. *arXiv preprint arXiv:2106.07477*, 2021b.

Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zihang Jiang, Francis EH Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token ViT: Training vision transformers from scratch on imagenet. *arXiv preprint arXiv:2101.11986*, 2021.

Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Seong Joon Oh, Youngjoon Yoo, and Junsuk Choe. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6022–6031, Seoul, Korea, 2019.

Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Muller, R. Manmatha, Mu Li, and Alexander Smola. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *Proceedings of the 6th International Conference on Learning Representations (ICLR)*, Vancouver, Canada, 2018.

Zizhao Zhang, Han Zhang, Long Zhao, Ting Chen, and Tomas Pfister. Aggregating nested transformers. *arXiv preprint arXiv:2105.12723*, 2021.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, pp. 13001–13008, New York, NY, 2020.