SpiralMLP: A Lightweight Vision MLP Architecture

Haojie Mu Burhan Ul Tayyab Nicholas Chua
Kookree
{mu, burhan, nicholas}@kookee.ai

Abstract

*We present **SpiralMLP**, a novel architecture introduces a **Spiral FC** layer as a replacement for the conventional Token Mixing approach. Differing from several existing MLP-based models that primarily emphasize axes, our Spiral FC layer is designed as a deformable convolution layer with spiral-like offsets. We further adapt Spiral FC into two variants: **Self-Spiral FC** and **Cross-Spiral FC**, enabling both local and global feature integration seamlessly, eliminating the need for additional processing steps. To thoroughly investigate the effectiveness of the spiral-like offsets and validate our design, we conduct ablation studies and explore optimal configurations. In empirical tests, SpiralMLP reaches state-of-the-art performance, similar to Transformers, CNNs, and other MLPs, benchmarking on ImageNet-1k, COCO and ADE20K. SpiralMLP still maintains linear computational complexity $O(HW)$ and is compatible with varying input image resolutions. Our study reveals that targeting the full receptive field is not essential for achieving high performance, instead, adopting a refined approach offers better results.*[1]

1. Introduction

1.1. Background

Earlier image classification systems mainly relied on CNN-based architectures [20, 52, 56, 84], which excel with controlled datasets but struggle with biased or uncontrolled conditions. Subsequently, Transformer-based architectures [2, 14, 27, 66] have emerged as alternatives, benefiting from self-attention mechanism that excel with large datasets [53] and are adaptable for various tasks [42]. However, they are often more expensive in pretraining and need specific datasets for better performance on downstream tasks.

MLP-based architectures [39, 59] have also shown promise in computer vision tasks, matching Transformer performance with a more data-efficient and lighter design. These systems use two main components: **Channel Mixing**, which projects features along the channel dimension, and **Token Mixing**, which captures spatial information by projecting feature along the spatial dimension. These mixing layers collectively enhance context aggregation, improving robustness and reducing training resource needs.

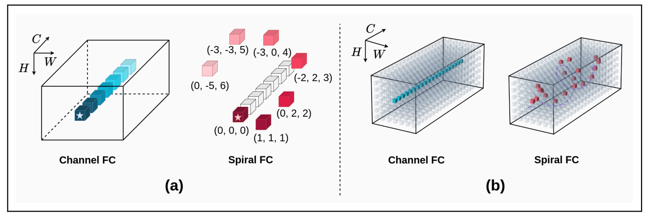

Figure 1. **(a)** While the Channel FC concentrates solely at the target point, marked with a ★, the Spiral FC captures richer spatial information. Spiral FC is in accordance with Eqs. (4) and (5), the input channel dimension $C_{\text{in}} = 14$, the maximum amplitude $A_{\text{max}} = 6$ and $T = 8$. The coordinate numbers are arranged as (H, W, C). This illustrative example only contains half of the C_{in}. **(b)** provides a complete visualization when the parameters are: $C_{\text{in}} = 20$, $A_{\text{max}} = 3$ and $T = 8$.

1.2. MLP-Based Architectures.

The pioneering MLP-Mixer [59] proposes a simple yet powerful architecture with both **Token Mixing** and **Channel Mixing**. Given a feature map $X \in \mathbb{R}^{H \times W \times C_{\text{in}}}$, where H, W are the height and weight, C_{in} is the input channel dimension, let $W^{\text{Tmix}} \in \mathbb{R}^{H \cdot W \times H \cdot W}$ denote the token mixing weight matrix, the operation applied to the reshaped input $X^T \in \mathbb{R}^{C_{\text{in}} \times H \cdot W}$ is described as follows:

$$\text{Tmix}(X) = (X^T W^{\text{Tmix}})^T \tag{1}$$

where, $\mathbb{R}^{H \cdot W}$ indicates the dimensions are flattened while $\mathbb{R}^{H \times W}$ denotes the dimensions are separated, and $\text{Tmix}(\cdot) \in \mathbb{R}^{H \cdot W \times C_{\text{in}}}$ is the output of token mixing. Eq. (1) is to simulate the attention operation to integrate spatial information, it is followed by the channel mixing that operates along the channel dimension. We define the channel mixing

[1]Our code is available at https://github.com/Kookree/SpiralMLP.

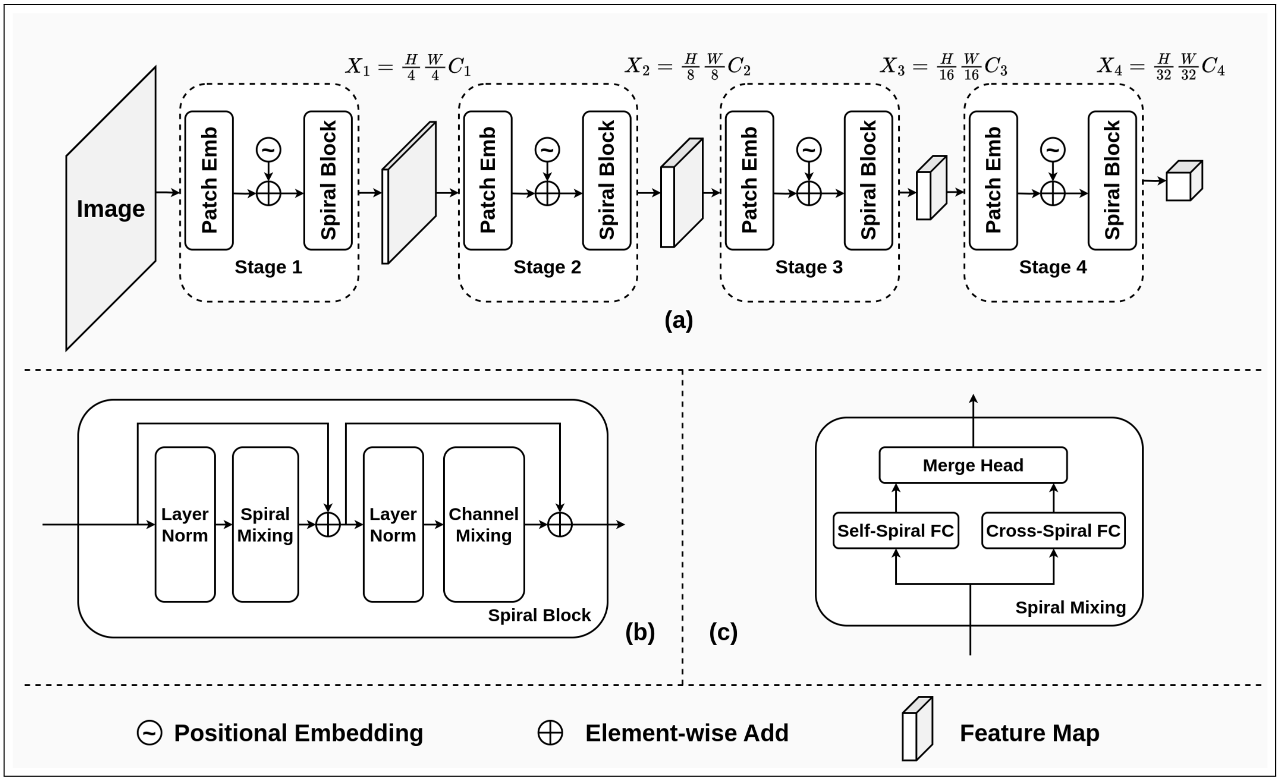

Figure 2. **(a)** displays the comprehensive architecture of SpiralMLP in PVT-style, featuring four distinct stages. Each stage is composed of multiple Patch Embedding layers and identically-configured Spiral Blocks. **(b)** explores the internal layout of a Spiral Block, where the proposed Spiral Mixing replaces the traditional Token Mixing. **(c)** outlines the components of Spiral Mixing, which incorporates the meticulously designed Spiral FC to effectively capture spatial information.

weight matrix as $W^{\text{Cmix}} \in \mathbb{R}^{C_{\text{in}} \times C_{\text{out}}}$, the channel mixing output $\text{Cmix}(\cdot) \in \mathbb{R}^{H \times W \times C_{\text{out}}}$ is expressed as follows:

$$\text{Cmix}(\text{Tmix}(X)) = (X^T W^{\text{Tmix}})^T W^{\text{Cmix}} \tag{2}$$

While MLP-Mixer shows strong performance, it is limited by its quadratic computational complexity $O(H^2W^2)$ (Eq. (1)) and requires fixed image sizes due to its fully-connected token mixing layer. Alternatives like gMLP [39] introduces a spatial gating unit for better integration, FNet [32] uses Fourier transforms for token mixing, and HireMLP [18] mimics self-attention by swapping elements across regions. Other developments include ResMLP [60], which replaces LayerNorm with trainable matrices, s^{2}MLP [76] utilizes shifting operations, WaveMLP [58] treats pixels as complex numbers, ViP [23] employs a permutator for spatial data, and MorphMLP [79] gradually expands its receptive field.

Despite advancements, the MLPs mentioned earlier haven't significantly cut computational complexity, leaving an opening for SparseMLP [57], ASMLP [36], CycleMLP [6], and ATM [69]. SparseMLP and ASMLP use dense token mixing along the channel dimension, while CycleMLP introduces Cycle FC for sparser, channel-wise mixing with fixed offsets S_H and S_W. ATM [69], on the other hand, uses trainable offsets for dynamic token mixing. However, these models restrict token mixing to horizontal and vertical axes, limiting their ability to integrate feature information across different spatial dimensions.

To address these challenges, we present **SpiralMLP** with its core component, **Spiral FC**, based on Channel FC as shown in Fig. 1(a). Spiral FC offers sufficient receptive field coverage while maintaining linear computational complexity. The paper is organized as follows:

- We introduce the SpiralMLP architecture and its foundational Spiral FC layer.
- We conduct experiments to demonstrate SpiralMLP's superiority over other state-of-the-art models.
- We perform ablation studies to explore optimal configurations, followed by conclusions and discussions on potential future improvements.

2. Methodology

2.1. Spiral FC

We aim to design a compact token mixing layer that captures spatial information efficiently. Our review indicates that traditional designs with criss-cross fully-connected layers fail to optimize the offset function, resulting in inadequate spatial coverage. To address these challenges, we draw inspiration from natural spiral patterns and AttentionViz [75], noted for its spiral patterns in transformer attention visualizations.

As a result, we introduce the **Spiral Fully-Connected Layer** (**Spiral FC**), intended to replace standard Token Mixing (Eq. (1)) in the MLP-Mixer architecture. Described in Fig. 1(b), Spiral FC leverages a **spiral trajectory** across the feature map $X \in \mathbb{R}^{H\times W\times C_{\text{in}}}$:

$$\text{Spiral FC}_{i,j,:}(X) = \sum_{c=0}^{C_{\text{in}}} X_{i+\phi_i(c),j+\phi_j(c),c} W^{\text{spiral}}_{c,:} + b^{\text{spiral}} \tag{3}$$

where, $W^{\text{spiral}} \in \mathbb{R}^{C_{\text{in}}\times C_{\text{out}}}$, $b^{\text{spiral}} \in \mathbb{R}^{C_{\text{out}}}$ are the trainable matrix and bias, Spiral $\text{FC}_{i,j,:}(\cdot)$ is the output at position $(i, j, :)$. Both $\phi_i(c)$ and $\phi_j(c)$ serve as the offset functions along H, W axes respectively within X. Furthermore, with the central axis of the spiral trajectory aligns along the channel dimension, the offset functions $\phi_i(c)$ and $\phi_j(c)$ are defined in a spiral manner:

$$\phi_i(c) = A(c)\cos\left(\frac{c\times 2\pi}{T}\right) \tag{4}$$

$$\phi_j(c) = A(c)\sin\left(\frac{c\times 2\pi}{T}\right) \tag{5}$$

where, T is the constant period, $A(\cdot)$ is the amplitude that controls the width of the spiral trajectory, for conciseness, we formulate the amplitude function $A(\cdot)$ with the basic pattern [2]:

$$A(c) = \begin{cases} \lfloor \frac{2A_{\text{max}}}{C_{\text{in}}} c \rfloor, & 0 \le c < \frac{C_{\text{in}}}{2} \\ \lfloor 2A_{\text{max}} - \frac{2A_{\text{max}}}{C_{\text{in}}} c \rfloor, & \frac{C_{\text{in}}}{2} \le c \le C_{\text{in}} \end{cases} \tag{6}$$

where, A_{max} is the maximum amplitude. When $A_{\text{max}} = 0$, the Spiral FC is identical to Channel FC, denoted as **Self-Spiral FC**. Conversely, when $A_{\text{max}} \neq 0$, it is termed as **Cross-Spiral FC**. Additionally, we employ a sliding window with a stepsize of 1. It not only makes the Spiral FC agonistic to the input size, but also enables the flexible feature extraction through meticulously modifying the offset functions (Eqs. (4) and (5)), thereby ensuring the Spiral FC operate with linear computational complexity.

[2] Sec. 4.1 provides additional cases with universal offset functions.

2.2. Spiral Mixing

At a specific position $(i, j, :)$, Self-Spiral FC captures the local information from itself, yielding an output denoted as $X^{\text{self}}_{i,j,:}$. Conversely, Cross-Spiral FC selectively incorporates spatial information from within the receptive field which is determined by A_{max}, and the output is represented as $X^{\text{cross}}_{i,j,:}$. Across the whole feature map, both the Self-Spiral FC and Cross-Spiral FC operate in parallel, and their outputs, $X^{\text{self}} \in \mathbb{R}^{H\times W\times C_{\text{out}}}$ and $X^{\text{cross}} \in \mathbb{R}^{H\times W\times C_{\text{out}}}$, merge together in the subsequent **Merge Head**[3]:

$$a = \sigma(W^{\text{merge}} \times [\frac{1}{HW}\sum_{i=0}^{HW} \mathcal{F}(X^{\text{self}} + X^{\text{cross}})_{i,:}]) \tag{7}$$

where, the reshaping function $\mathcal{F} : \mathbb{R}^{H\times W\times C_{\text{out}}} \to \mathbb{R}^{H\cdot W\times C_{\text{out}}}$ flattens the first two dimensions of the input, creating a new projection along the HW dimension. Then, the newly generated projection is averaged into $\mathbb{R}^{1\times C_{\text{out}}}$. Subsequently, $W^{\text{merge}} \in \mathbb{R}^{2,1}$ maps this average from $\mathbb{R}^{1\times C_{\text{out}}}$ to $\mathbb{R}^{2\times C_{\text{out}}}$. Finally, the SoftMax function $\sigma(\cdot)$ determines the weights $a \in \mathbb{R}^{2\times C_{\text{out}}}$. Then at position $(i, j, :)$, the Merge Head generates the output:

$$X^{\text{spiral}}_{i,j,:} = a_{1,:} \odot X^{\text{self}}_{i,j,:} + a_{2,:} \odot X^{\text{cross}}_{i,j,:} \tag{8}$$

where, $\odot$ represents the element-wise multiplication. The weights a is to modulate the contribution of the inputs. Furthermore, across the entire X^{spiral}, the weights a is broadcast to influence all elements in both X^{self} and X^{cross}.

Collectively, Self-Spiral FC, Cross-Spiral FC and Merge Head together constitute the **Spiral Mixing**, as depicted in Fig. 2 (c). Spiral Mixing transforms the input feature map $X \in \mathbb{R}^{H\times W\times C_{\text{in}}}$ to $X^{\text{spiral}} \in \mathbb{R}^{H\times W\times C_{\text{out}}}$, functioning similarly to vanilla Token Mixing.

2.3. Spiral Block

The output X^{spiral} of Spiral Mixing subsequently proceeds to the **Channel Mixing** structured as a MLP with a GeLU [21] activation function $\zeta(\cdot)$:

$$X^{\text{chn}} = \zeta(X^{\text{spiral}} \times W^{\text{mlp1}}) \times W^{\text{mlp2}} \tag{9}$$

where, $W^{\text{mlp1}} \in \mathbb{R}^{C_{\text{out}}\times C_{\text{mlp}}}$ and $W^{\text{mlp2}} \in \mathbb{R}^{C_{\text{mlp}}\times C_{\text{out}}}$ are the linear layer weight matrices. X^{chn} is the output of Channel Mixing.

Spiral Mixing and Channel Mixing collectively compose the **Spiral Block**, as depicted in Fig. 2 (b). To summarize, Spiral Block accepts the feature map $X \in \mathbb{R}^{H\times W\times C_{\text{in}}}$ as the input, and initially processes it through a LayerNorm [1] before the Spiral Mixing. Then it produces X' integrated with a residual connection. Following this, X' is processed

[3] Detailed explanation is provided in Appendix.

Model	CIFAR-10(%)	CIFAR-100(%)	Params(M)
Spiral-B1 (ours)	**95.6**	**78.6**	**14**
CaiT [62]	94.9	76.9	9
MONet-T [7]	94.8	77.2	10.3
Cycle-B1 [6]	94.5	77.3	15
PiT [22]	94.2	75.0	7
Swin [42]	94.0	77.3	7
VGG19-bn [52]	94.0	72.2	39
ResNet50 [20]	93.7	77.4	24
ViT [14]	93.6	73.8	3
Swin-v2-T [41]	89.7	70.2	28

Table 1. Top-1 accuracy achieved through training from scratch on both CIFAR-10 and CIFAR-100.

through another LayerNorm and then Channel Mixing, coupled with another residual connection, resulting in the output Y:

$$X' = \text{Spiral Mixing}(\text{LN}(X)) + X \tag{10}$$
$$Y = \text{Channel Mixing}(\text{LN}(X')) + X' \tag{11}$$

2.4. Overall Architecture and Model Zoo

We firstly construct our **SpiralMLP** based on the PVT [68] framework, the models are scaled from **SpiralMLP-B1** to **SpiralMLP-B5** by adjusting the hyper-parameters. In each model, 4 stages are integrated, and the spatial resolution is reduced while the channel dimension is increased along with the process. Thereby it facilitates effective down-sampling of spatial resolution and optimizes computational efficiency. A depiction of the PVT-style SpiralMLP architecture can be found in Fig. 2 (a).

Furthermore, we have also developed variants modeled after the Swin architecture. The models are categorized into three types: **SpiralMLP-T (Tiny)**, **SpiralMLP-S (Small)**, and **SpiralMLP-B (Base)**. The structural details of both PVT-style and Swin-style will be further provided in the appendix.

3. Experiments

We initially perform experiments with SpiralMLP-B1 on CIFAR-10 [29] and CIFAR-100 [29], comparing it against architectures of similar scale, including MLPs, CNNs, and Transformers. The outcomes are presented in Sec. 3, all of the models are trained from scratch.

We extend our experimentation to include image classification on ImageNet-1k [50], as well as object detection and instance segmentation on the COCO [38]. Furthermore, we assess its semantic segmentation capabilities on ADE20K [82].

3.1. Image Classification on ImageNet-1k

3.1.1 Settings

Our implementation primarily draws from DeiT [61]. The training is 4 NVIDIA A100 GPUs for a total of 300 epochs. The overall batch size is 512 and we employ the Top-1 accuracy for image classification.

3.1.2 Comparison with MLPs

As shown in Sec. 3 , SpiralMLP-B achieves a Top-1 accuracy of 84.0% on the ImageNet-1k, with the input resolution of 224×224. This performance notably exceeds that of the best-performing models of ATMNet-L [69], HireMLP-Large [18], WaveMLP-B [58], MorphMLP-L [79] and CycleMLP-B [6], by +0.2%, +0.2%, +0.4%, +0.6% and +0.6%, respectively. Furthermore, compared to the S^{2}MLP-wide [76], which has a similar model size with 71M parameters, SpiralMLP surpasses it by +4.0% with only 68M parameters. In addition to the advantage on the model size, SpiralMLP also demonstrates potential balance between computational efficiency and accuracy. It is evident that among a cohort of models with accuracy exceeding 83% (including ATMNet-L [69], HireMLP-Large [18], WaveMLP-B [58], MorphMLP-B [79], MorphMLP-L [79], CycleMLP-B [6], CycleMLP-B5 [6], sMLP-B [57] and ASMLP-B [36], SpiralMLP-B5 stands out due to a lower FLOPs of 11.0G and the highest accuracy.

3.1.3 Comparison with other SOTAs

SpiralMLP remains competitive over Transformers, CNNs and State-Space Models, particularly in significantly reducing the number of parameters and the FLOPs as referenced in Sec. 3 , , . For instance, when comparing SpiralMLP-B5 to CNNs , it outperforms VanillaNet-13-1.5 [4] by +1.5% and has the same performance to DeepMAD-89M [51]. When comparing between State-Space Models and SpiralMLP-B4 as well as SpiralMLP-S, SpiralMLP demonstrates a notable performance improvement of approximate +4.0%. Furthermore, when comparing with the Transformers , SpiralMLP-B5 has nearly 20M fewer parameters than Swin-B [42] while achieving +0.5% higher in accuracy. Particularly the vision transformers continue struggling with quadratic complexity. And in order to better demonstrate, we visualize the heatmaps in Fig. 3 in comparison with the performance of ASMLP [36] and Swin [42].

3.2. Object Detection and Instance Segmentation on COCO

3.2.1 Settings

We conduct object detection and instance segmentation experiments on COCO [38], wherein we demonstrate Spi-

Model	Top-1 Acc (%)	Params (M)	FLOPs (G)	Model	Top-1 Acc (%)	Params (M)	FLOPs (G)
SpiralMLP-B5 (ours)	**84.0**	**68**	**11.0**	Swin-B [42]	83.5	88	15.4
SpiralMLP-B4 (ours)	83.8	46	8.2	gSwin-S [17]	83.0	19	4.2
SpiralMLP-B (ours)	83.6	67	11.0	SimA-XCiT-S12/16 [28]	82.1	26	4.8
SpiraMLP-S (ours)	83.3	56	9.1	SimA-CvT-13 [28]	81.4	20	4.5
ATMNet-L [69]	83.8	76	12.3	SimA-DeiT-S [28]	79.8	22	4.6
HireMLP-Large [18]	83.8	96	13.4	NOAH [33]	77.3	26	-
WaveMLP-B [58]	83.6	63	10.2	CRATE-L [77]	71.3	78	-
MorphMLP-L [79]	83.4	76	12.5	CRATE-B [77]	70.8	23	-
MorphMLP-B [79]	83.2	58	10.2	DeepMAD-89M [51]	84.0	89	15.4
CycleMLP-B [6]	83.4	88	15.2	DeepMAD-50M [51]	83.9	50	8.7
CycleMLP-B5 [6]	83.1	76	12.3	EfficientNet-B4 [56]	82.6	19	4.2
sMLP-B [57]	83.4	66	14.0	VanillaNet-13-1.5 [4]	82.5	128	26.5
ASMLP-B [36]	83.3	88	15.2	VanillaNet-13 [4]	82.1	59	11.9
gMLP [39]	81.6	45	31.6	HGRN-DeiT-Small [47]	80.1	24	-
ConvMixer-1536/20 [65]	81.4	52	-	HGRN-DeiT-Tiny [47]	74.4	6	-
ConvMixer-1536/20 [65]	80.4	49	-	ResNet-50 [20]	75.3	25	3.8
MONet-S [7]	81.3	33	6.8	Vim-S [83]	80.5	26	-
MONet-T [7]	77.0	10	2.8	Vim-Ti [83]	78.3	7	-
ResMLP-B24 [60]	81.0	116	23.0	M2-ViT-b [15]	79.5	45	-
S^2MLP-deep [76]	80.7	51	10.5	ViT-b-Monarch [15]	78.9	33	-
S^2MLP-wide [76]	80.0	71	14.0	HyenaViT-b [46]	78.5	88	-
ConvMLP-L [34]	80.2	43	9.9	RepMLP-Res50-g8/8 [13]	76.4	59	12.7
ConvMLP-M [34]	79.0	17	4.0	MLPMixer-B/16 [59]	76.4	59	12.7
RepMLP-Res50-g4/8 [13]	80.1	87	8.2	AFFNet [25]	79.8	6	1.5

Table 2. Top-1 accuracy on ImageNet-1k, with 224 × 224 as the input resolution. In terms of background colors, , , , denote MLPs, Transformers, CNNs and State-Space Models, respectively.

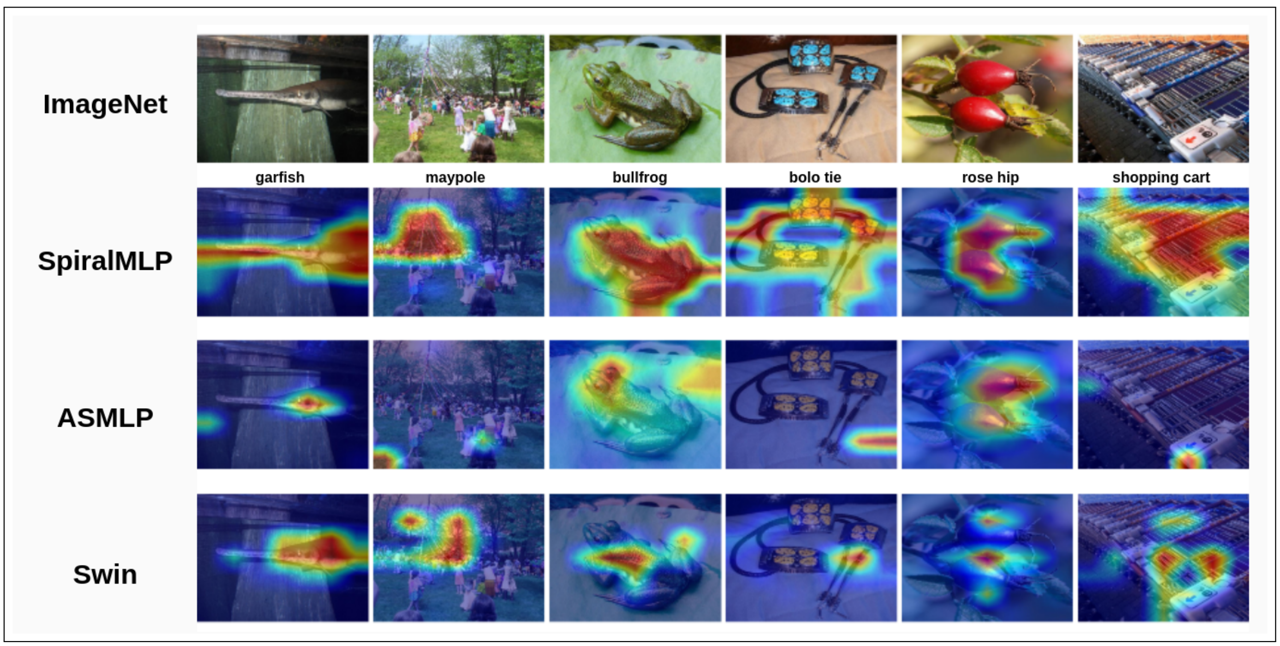

Figure 3. In contrast to ASMLP [36] and Swin [42], our SpiralMLP demonstrates superior object-focused attention. SpiralMLP exhibits enhanced sensitivity, especially for elongated or curved objects. The backbones employed for heatmaps generation are SpiralMLP-B5, ASMLP-B and Swin-B. The images are sourced from the ImageNet-1k [50] validation dataset, with corresponding labels.

BackBone	RetinaNet $1\times$							
	Params (M)	FLOPs (G)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet101 [20]	56.7	492.2	38.5	57.8	41.2	21.4	42.6	51.1
ConvMLP-L [34]	52.9	-	40.2	59.3	43.3	23.5	43.8	53.3
ResNeXt101-64x4d [71]	95.5	-	41.0	60.9	44.0	23.9	45.2	54.0
CycleMLP-B5 [6]	85.9	360.3	42.7	63.3	45.3	24.1	46.3	57.4
ATMNet-L [69]	86.0	405.0	46.1	67.4	49.4	29.9	50.1	61.0
PVTv2-B5 [68]	91.7	-	46.2	67.1	49.5	28.5	50.0	62.5
SpiralMLP-B5 (ours)	**79.8**	**325.0**	**46.5**	**67.7**	**49.8**	**30.3**	**50.8**	**62.8**
BackBone	Mask R-CNN $1\times$							
	Params (M)	FLOPs (G)	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet101 [20]	63.2	-	40.4	61.1	44.2	36.4	57.7	38.8
ConvMLP-L [34]	62.2	-	41.7	62.8	45.5	38.2	59.9	41.1
Swin-T [42]	47.8	267.0	42.7	65.2	46.8	39.3	62.2	42.2
ResNeXt101-64x4d [71]	101.9	-	42.8	63.8	47.3	38.4	60.6	41.3
VanillaNet-13 [4]	76.3	421.0	42.9	65.5	46.9	39.6	62.5	42.2
CycleMLP-B5 [6]	95.3	-	44.1	65.5	48.4	40.1	62.8	43.0
HireMLP-Large (1x) [18]	155.2	443.5	45.9	67.2	50.4	41.7	64.7	45.3
PVTv2-B5 [68]	101.6	334.5	47.4	68.6	51.9	42.5	65.7	46.0
ATMNet-L [69]	96.0	424.0	47.4	69.9	52.0	43.2	67.3	46.5
SpiralMLP-B (ours)	**89.1**	**342.0**	**47.8**	**71.6**	**53.2**	**43.6**	**69.3**	**47.2**

Table 3. Object detection performance with RetinaNet $1\times$ and MASK R-CNN $1\times$ on the COCO validation dataset, all of the backbones are pretrained on the ImageNet-1k. The FLOPs are evaluated at a resolution of 1280×800. The entries are sorted in ascending order based on AP performance.

Figure 4. Several examples of the object detection and instance segmentation from COCO [38] test dataset.

Figure 5. Several examples of the semantic segmentation from ADE20K [82] validation dataset.

ralMLP with PVT and Swin architectures, adopting two distinct configurations. We leverage SpiralMLP-B5 and Spiral-B with the pretrained weights on ImageNet-1k [50] as the backbones, together with Xavier initialization [16] applied to the newly added layers.

3.2.2 Results

Comparative results are detailed in Sec. 3.1.2, where we employ either RetinaNet [37] or Mask R-CNN [19] as the detection framework. When comparing under the RetinaNet $1\times$, SpiralMLP-B5 stands out in terms of the highest AP. In particular, it achieves +0.3% higher than PVTv2-B5, with -11.9M fewer parameters. In the context of Mask R-CNN $1\times$, SpiralMLP-B outperforms ATMNet-L by +0.4% in AP, alongside a reduction of 6.9M in model parameters. Visual representations of the object detection and instance segmentation are presented in Fig. 4.

3.3. Semantic Segmentation on ADE20K

3.3.1 Settings

We perform semantic segmentation on the ADE20K dataset using UperNet [70] and Semantic FPN [26] as the frameworks. For the backbones, we employ SpiralMLP-B5 and SpiralMLP-B, with the weights pretrained on ImageNet-1k. Additionally, the newly add layers are initialized with Xavier [16].

3.3.2 Results

As depicted in Sec. 3.2.2, SpiralMLP still exhibits comparable performance when integrated with Semantic FPN and UperNet for semantic segmentation tasks. In the Semantic FPN evaluations, SpiralMLP-B5 surpasses its closest competitor, PVTv2-B5, by +0.2%, and exceeds the second-best model, ATMNet-L, by +0.6%. When integrated with

Model	Semantic FPN			Model	UperNet		
	Params	FLOPs	mIoU		Params	FLOPs	mIoU
ResNet101 [20]	47.5	10.1	38.8	DeepMAD-29M* [51]	27	56	46.9
ConvMLP-L [34]	46.3	-	40.0	HireMLP-Large [18]	127	1125	48.8
ResNeXt101-64x4d [71]	86.4	103.9	40.2	Focal-B [73]	126	-	49.0
CycleMLP-B5 [6]	79.4	86.0	45.5	ConvNeXt-T [43]	82	-	48.7
MorphMLP-B [79]	59.3	76.8	45.9	ConvNeXt-B [43]	122	-	49.1
Swin-B [42]	91.2	107.0	46.0	AS-MLP-B [36]	121	1166	49.5
Twins-L [8]	103.7	102.0	46.7	Swin-B [42]	121	1188	49.7
ConvNeXt-T [43]	27.8	93.2	46.7	CycleMLP-B [6]	121	1166	49.7
ATMNet-L [69]	79.8	86.6	48.1	ATMNet-L [69]	108	1106	50.1
PVTv2-B5 [68]	85.7	91.1	48.7	FocalNet-B(LRF) [72]	126	-	50.5
SpiralMLP-B5 (ours)	**73.2**	**75.5**	**48.9**	**SpiralMLP-B (ours)**	**100**	**1061**	**50.7**

Table 4. Semantic segmentation performance on ADE20K validation dataset with Semantic FPN as well as UperNet. When evaluated with Semantic FPN, the FLOPs are measured at a resolution of 512×512. When evaluated with UperNet, the FLOPs are measured at a resolution of 2048×512. The entries are sorted in ascending order based on mIOU performance.

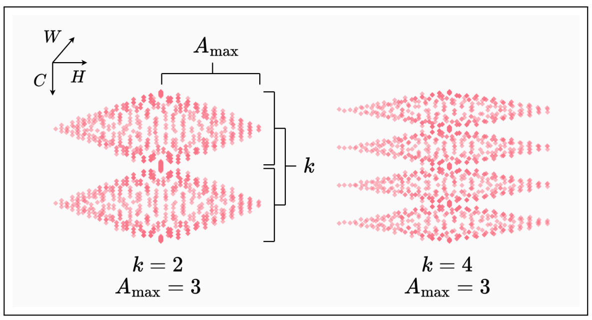

Figure 6. Visualization of varying k on spiral trajectory as described by Eqs. (14) and (15), while maintaining a constant $A_{\max} = 3$.

Case 1	k				
$A_{\max} = 3$	1	2	3	4	5
Acc(%)	83.9	84.0	83.8	83.6	83.3
Case 2	$A_{\max}$				
$k = 2$	2	3	4	5	6
Acc(%)	83.8	84.0	83.7	83.4	82.9

Table 5. Experiments on k and $A_{\max}$. After reaching their respective peaks, both trends show a rapid decline.

UperNet, SpiralMLP-B still emerges as the top-performing model, outperforming FocalNet-B(LRF) [72] by +0.2% and ATMNet-L [69] by +0.6%. Visual representations of the semantic segmentation are presented in Fig. 5.

4. Ablation

4.1. Update the Offset Functions

The offset functions $\phi_i(\cdot)$ and $\phi_j(\cdot)$ (Eqs. (4) and (5)) are originally designed into a two-partition pattern, and we further expand them to a more generic multi-partition pattern. To incorporate this update, we introduce k as the number of partitions along the channel dimension. The partitions can be defined as follows:

$$P = \left\{0, \frac{C_{\text{in}}}{k}, \frac{2 * C_{\text{in}}}{k}, \dots, C_{\text{in}}\right\} \tag{12}$$

By introducing k and considering individual partition, we can create multiple spiral structures that capture the characteristics of each partition along the channel dimension. Furthermore, we define the length of the partition as $C_w = \frac{C_{\text{in}}}{k}$, which is the distance between two adjacent endpoints, then the amplitude function Eq. (6) is updated to:

$$A^*(c) = \begin{cases} \left\lfloor \frac{2A_{\max}}{C_w}(c - iC_w) \right\rfloor, & 0 \le c < \frac{iC_w}{2} \\ \left\lfloor (2A_{\max} - \frac{2A_{\max}}{C_w})(c - iC_w) \right\rfloor, & \frac{iC_w}{2} \le c \le iC_w \end{cases} \tag{13}$$

where, $i \in [0, 1 \dots, k-1]$ represents the i^{th} partition in partitions P (Eq. (12)), and c in Eq. (6) is replaced by z within the i^{th} partition. Accordingly, Eqs. (4) and (5) are updated as:

$$\phi_i^*(c) = A^*(c)\cos(\frac{c * 2\pi}{T}) \tag{14}$$

$$\phi_j^*(c) = A^*(c)\sin(\frac{c * 2\pi}{T}) \tag{15}$$

We also provide the visualizations of Eqs. (14) and (15), as depicted in Fig. 6, showcasing variations with different numbers of partitions k.

4.2. Ablation Study on k

We updated the offset functions $\phi_i^*(\cdot)$ and $\phi_j^*(\cdot)$ (Eqs. (14) and (15)) to analyze how varying the number of

	SpiralFC **(ours)**	PATM	ATMLayer	CycleFC	RandomFC
Acc (%)	**95.6**	95.3	95.2	94.7	94.5
Params (M)	**14**	17	15	15	14

Table 6. The accuracy on CIFAR-10, each Fully-Connected Layer is configured into the SpiralMLP-B1 architecture and is trained from scratch.

partitions k affects Top-1 Accuracy on ImageNet-1k. Results, shown in Tab. 5 with a constant maximum amplitude A_{max} of 3, indicate that accuracy initially rises, peaks at $k = 2$, then decreases.

This trend suggests that different k values alter the focus on the peripheral regions of the receptive field, where $k = 2$ results in denser clustering of feature points along the edges compared to $k = 4$, as seen in Fig. 6. Lower k values cause a dense, narrow concentration of features, while higher values disperse them too widely, potentially reducing model effectiveness.

4.3. Ablation Study on A_{max}

We investigate several cases when the maximum amplitude A_{max} takes various values. From the results shown in Tab. 5, we observe an initial improvement in the Top-1 Accuracy on ImageNet-1k. However, a decline becomes evident once the A_{max} exceeds 3.

Similarly, the underlying reason is that, as A_{max} increases, the receptive field's extent expands. However, due to the characteristics of Spiral FC, the number of selected feature points remains constant at C_{in}. Consequently, a larger A_{max} results in a more sparse distribution of feature points. If A_{max} is too small, the Spiral FC may fail to encompass a sufficient number of neighboring features. On the other hand, if A_{max} is excessively large, the Spiral FC might not effectively capture detailed information within the receptive field.

Although the discrete experimental design does not guarantee the discovery of optimal hyperparameters, it indeed facilitates the insight of underlying trends and tendencies.

4.4. Ablation Study on Fully-Connected Layers

To illustrate the effectiveness of Spiral FC, we perform experiments on the CIFAR-10 [29] using SpiralMLP-B1[4] as the base architecture. In these experiments, the Spiral FC is substituted with various alternatives, including `PATM` from WaveMLP [58], `ATMLayer` from ATM [69], `CycleFC` from CycleMLP [6] and a `RandomFC`. The Random FC is architecturally identical to Spiral FC, except that the offset function is generated randomly.

[4]The configuration of SpiralMLP-B1 is demonstrated in Appendix.

Model	Params(M)	Latency(ms)
SpiralMLP-B4	46	47.00
SpiralMLP-B5	68	39.22
CycleMLP-B4 [6]	52	57.94
CycleMLP-B5	76	48.38
ATM-B [69]	52	64.77
ATM-L	76	54.09
PVTv2-B4 [68]	63	43.96
PVTv2-B5	82	55.71

Table 7. Inference latency measured in *milliseconds* on one A100. SpiralMLP outperforms other models of similar size in terms of speed. A single image of with 224^2 resolution serves as the input.

4.5. Latency Analysis

To highlight the speed efficiency of Spiral FC, we assess its performance across various input resolutions compared to other proposed architectures. We adopt the format from EfficientFormer [35] and detail the latency analysis in Sec. 4.5. We present SpiralMLP-B4 and SpiralMLP-B5 with several other architectures closely related to our study and specifically at the 224^2 resolution. For a comprehensive latency comparison across different scenarios, please refer to the Appendix.

5. Conclusion and Future Work

In this paper, we present Spiral FC, part of Spiral Mixing designed to replace traditional Token Mixing. We introduce SpiralMLP, a new computer vision framework compatible with PVT-style and Swin-style architectures. SpiralMLP performs comparably to leading models while using fewer parameters and less computational power.

We believe we are the first to use carefully designed offset functions to capture comprehensive feature information, setting us apart from models like CyelcMLP [6], ASMLP [36] and ATM [69], which focus on optimizing cross-like layers. Given its strong performance, further research into optimizing SpiralMLP's hyperparameters could lead to even more efficient information capture.

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. 3

[2] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers, 2022. 1, 12

[3] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020. 12

[4] Hanting Chen, Yunhe Wang, Jianyuan Guo, and Dacheng Tao. Vanillanet: the power of minimalism in deep learning. *Advances in Neural Information Processing Systems*, 36, 2024. 4, 5, 6

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 14

[6] Shoufa Chen, Enze Xie, Chongjian Ge, Runjian Chen, Ding Liang, and Ping Luo. Cyclemlp: A mlp-like architecture for dense prediction, 2022. 2, 4, 5, 6, 7, 8, 16, 17

[7] Yixin Cheng, Grigorios G Chrysos, Markos Georgopoulos, and Volkan Cevher. Multilinear operator networks. *arXiv preprint arXiv:2401.17992*, 2024. 4, 5

[8] 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. *Advances in Neural Information Processing Systems*, 34:9355–9366, 2021. 7

[9] Jean-Baptiste Cordonnier, Andreas Loukas, and Martin Jaggi. On the relationship between self-attention and convolutional layers, 2020. 12, 13

[10] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019. 14

[11] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 15

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019. 12

[13] Xiaohan Ding, Chunlong Xia, Xiangyu Zhang, Xiaojie Chu, Jungong Han, and Guiguang Ding. Repmlp: Reparameterizing convolutions into fully-connected layers for image recognition. *arXiv preprint arXiv:2105.01883*, 2021. 5

[14] 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, 2021. 1, 4, 12, 16, 17

[15] Dan Fu, Simran Arora, Jessica Grogan, Isys Johnson, Evan Sabri Eyuboglu, Armin Thomas, Benjamin Spector, Michael Poli, Atri Rudra, and Christopher Ré. Monarch mixer: A simple sub-quadratic gemm-based architecture. *Advances in Neural Information Processing Systems*, 36, 2024. 5

[16] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. 6, 14

[17] Mocho Go and Hideyuki Tachibana. gswin: Gated mlp vision model with hierarchical structure of shifted window. In *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5. IEEE, 2023. 5

[18] Jianyuan Guo, Yehui Tang, Kai Han, Xinghao Chen, Han Wu, Chao Xu, Chang Xu, and Yunhe Wang. Hire-mlp: Vision mlp via hierarchical rearrangement, 2021. 2, 4, 5, 6, 7

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018. 6, 14

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 1, 4, 5, 6, 7, 12, 16, 17

[21] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus), 2023. 3

[22] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11936–11945, 2021. 4

[23] Qibin Hou, Zihang Jiang, Li Yuan, Ming-Ming Cheng, Shuicheng Yan, and Jiashi Feng. Vision permutator: A permutable mlp-like architecture for visual recognition, 2021. 2

[24] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth, 2016. 14

[25] Zhipeng Huang, Zhizheng Zhang, Cuiling Lan, Zheng-Jun Zha, Yan Lu, and Baining Guo. Adaptive frequency filters as efficient global token mixers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6049–6059, 2023. 5

[26] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks, 2019. 6, 14

[27] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning, 2020. 1

[28] Soroush Abbasi Koohpayegani and Hamed Pirsiavash. Sima: Simple softmax-free attention for vision transformers. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2607–2617, 2024. 5

[29] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 4, 8

[30] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. 12

[31] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 12

[32] James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms, 2022. 2

[33] Chao Li, Aojun Zhou, and Anbang Yao. Noah: Learning pairwise object category attentions for image classification. *arXiv preprint arXiv:2402.02377*, 2024. 5

[34] Jiachen Li, Ali Hassani, Steven Walton, and Humphrey Shi. Convmlp: Hierarchical convolutional mlps for vision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6306–6315, 2023. 5, 6, 7

[35] Yanyu Li, Geng Yuan, Yang Wen, Ju Hu, Georgios Evangelidis, Sergey Tulyakov, Yanzhi Wang, and Jian Ren. Efficientformer: Vision transformers at mobilenet speed. *Advances in Neural Information Processing Systems*, 35:12934–12949, 2022. 8

[36] Dongze Lian, Zehao Yu, Xing Sun, and Shenghua Gao. As-mlp: An axial shifted mlp architecture for vision, 2022. 2, 4, 5, 7, 8, 16

[37] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018. 6, 14

[38] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. 4, 6, 14

[39] Hanxiao Liu, Zihang Dai, David R. So, and Quoc V. Le. Pay attention to mlps, 2021. 1, 2, 5

[40] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019. 12

[41] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12009–12019, 2022. 4

[42] 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, 2021. 1, 4, 5, 6, 7, 12

[43] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022. 7

[44] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019. 14

[45] OpenAI. Gpt-4 technical report, 2023. 12

[46] Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. *arXiv preprint arXiv:2302.10866*, 2023. 5

[47] Zhen Qin, Songlin Yang, and Yiran Zhong. Hierarchically gated recurrent neural network for sequence modeling. *Advances in Neural Information Processing Systems*, 36, 2024. 5

[48] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners, 2019. 12

[49] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer, 2020. 12

[50] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge, 2015. 4, 5, 6, 14

[51] Xuan Shen, Yaohua Wang, Ming Lin, Yilun Huang, Hao Tang, Xiuyu Sun, and Yanzhi Wang. Deepmad: Mathematical architecture design for deep convolutional neural network. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6163–6173, 2023. 4, 5, 7

[52] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, 2015. 1, 4, 12

[53] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era, 2017. 1

[54] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. Ernie: Enhanced representation through knowledge integration, 2019. 12

[55] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions, 2014. 12

[56] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020. 1, 5

[57] Chuanxin Tang, Yucheng Zhao, Guangting Wang, Chong Luo, Wenxuan Xie, and Wenjun Zeng. Sparse mlp for image recognition: Is self-attention really necessary?, 2022. 2, 4, 5

[58] Yehui Tang, Kai Han, Jianyuan Guo, Chang Xu, Yanxi Li, Chao Xu, and Yunhe Wang. An image patch is a wave: Phase-aware vision mlp, 2022. 2, 4, 5, 8, 17

[59] 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, 2021. 1, 5

[60] Hugo Touvron, Piotr Bojanowski, Mathilde Caron, Matthieu Cord, Alaaeldin El-Nouby, Edouard Grave, Gautier Izacard, Armand Joulin, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. Resmlp: Feedforward networks for image classification with data-efficient training, 2021. 2, 5

[61] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention, 2021. 4, 16, 17

[62] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers, 2021. 4

[63] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. 12

[64] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. 12

[65] Asher Trockman and J Zico Kolter. Patches are all you need? *arXiv preprint arXiv:2201.09792*, 2022. 5

[66] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. 1, 12

[67] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 568–578, 2021. 16, 17

[68] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 4, 6, 7, 8, 17

[69] Guoqiang Wei, Zhizheng Zhang, Cuiling Lan, Yan Lu, and Zhibo Chen. Active token mixer, 2022. 2, 4, 5, 6, 7, 8, 17

[70] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding, 2018. 6, 14

[71] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 6, 7

[72] Jianwei Yang, Chunyuan Li, Xiyang Dai, and Jianfeng Gao. Focal modulation networks. *Advances in Neural Information Processing Systems*, 35:4203–4217, 2022. 7

[73] 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. 7

[74] Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2020. 12

[75] Catherine Yeh, Yida Chen, Aoyu Wu, Cynthia Chen, Fernanda Viégas, and Martin Wattenberg. Attentionviz: A global view of transformer attention, 2023. 3, 17

[76] Tan Yu, Xu Li, Yunfeng Cai, Mingming Sun, and Ping Li. S^2-mlp: Spatial-shift mlp architecture for vision, 2021. 2, 4, 5

[77] Yaodong Yu, Sam Buchanan, Druv Pai, Tianzhe Chu, Ziyang Wu, Shengbang Tong, Benjamin Haeffele, and Yi Ma. White-box transformers via sparse rate reduction. *Advances in Neural Information Processing Systems*, 36, 2024. 5

[78] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features, 2019. 14

[79] David Junhao Zhang, Kunchang Li, Yali Wang, Yunpeng Chen, Shashwat Chandra, Yu Qiao, Luoqi Liu, and Mike Zheng Shou. Morphmlp: An efficient mlp-like backbone for spatial-temporal representation learning, 2022. 2, 4, 5, 7

[80] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization, 2018. 14

[81] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation, 2017. 14

[82] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset, 2018. 4, 6, 14

[83] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024. 5

[84] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition, 2018. 1