Active Token Mixer

Guoqiang Wei[1]*, Zhizheng Zhang[2]*, Cuiling Lan[2], Yan Lu[2], Zhibo Chen[1]

[1]University of Science and Technology of China
[2]Microsoft Research Asia
wgq7441@mail.ustc.edu.cn,{zhizzhang, culan, yanlu}@microsoft.com, chenzhibo@ustc.edu.cn

Abstract

The three existing dominant network families, i.e., CNNs, Transformers, and MLPs, differ from each other mainly in the ways of fusing spatial contextual information, leaving designing more effective token-mixing mechanisms at the core of backbone architecture development. In this work, we propose an innovative token-mixer, dubbed Active Token Mixer (**ATM**), to actively incorporate flexible contextual information distributed across different channels from other tokens into the given query token. This fundamental operator actively predicts where to capture useful contexts and learns how to fuse the captured contexts with the query token at channel level. In this way, the spatial range of token-mixing can be expanded to a global scope with limited computational complexity, where the way of token-mixing is reformed. We take ATM as the primary operator and assemble ATMs into a cascade architecture, dubbed **ATMNet**. Extensive experiments demonstrate that ATMNet is generally applicable and comprehensively surpasses different families of SOTA vision backbones by a clear margin on a broad range of vision tasks, including visual recognition and dense prediction tasks. Code is available at https://github.com/microsoft/ActiveMLP.

Introduction

Convolutional neural networks (CNNs) (Krizhevsky, Sutskever, and Hinton 2012; Simonyan and Zisserman 2015; Szegedy et al. 2015, 2016; Huang et al. 2017) serve as the most prevalent vision backbones for a long time. Inspired by the successes in Natural Language Processing (NLP), DETR (Carion et al. 2020) and ViT (Kolesnikov et al. 2021) introduce self-attention based model, *i.e*., Transformer, into computer vision. Afterwards, Transformers spring up and make splendid breakthroughs on various vision tasks (Liu et al. 2021; He et al. 2021b; Wang et al. 2021a; Xie et al. 2021; Cheng, Schwing, and Kirillov 2021; Lin, Wang, and Liu 2021; He et al. 2021a). Most recently, the multi-layer perceptrons (MLPs) based architectures (Tolstikhin et al. 2021; Lian et al. 2022) have regained their light and been demonstrated capable of achieving stunning results on various vision tasks (Touvron et al. 2021a; Tolstikhin et al. 2021; Chen et al. 2022; Lian et al. 2022; Zhang et al. 2021; Tang et al. 2022).

Those three categories of architectures differ from each other mainly in their different ways of token mixing. For different architectures, we uniformly refer to each feature vector as one token. CNN-based architectures (Simonyan and Zisserman 2015; He et al. 2016; Huang et al. 2017) mix tokens locally within a sliding window of a fixed shape. Transformer-based architectures (Kolesnikov et al. 2021; Touvron et al. 2021b; Wang et al. 2021b) perform message passing from tokens in the global scope into the query token based on the pairwise attentions commonly modeled by the affinities between tokens in the embedding space. MLP-based architectures mostly enable spatial information interaction through the fully connected layers across all tokens (Tolstikhin et al. 2021; Touvron et al. 2021a; Hou et al. 2022; Tang et al. 2022) or certain tokens selected with hand-crafted rules in a deterministic manner (Chen et al. 2022; Zhang et al. 2021; Wang et al. 2022a; Yu et al. 2022; Lian et al. 2022; Tang et al. 2021). However, the fully connected layer across all tokens makes the model unable to cope with the inputs of variable resolutions. Adopting manually designed rules for token selection relaxes this constraint on fixed resolutions by restricting token mixing within a *deterministic* region, but sacrificing the adaptability to various visual contents of diverse feature patterns.

In this work, we first revisit the token mixing mechanisms in dominant types of architectures from a unified perspective, then propose a novel Active Token Mixer (ATM). As an innovative basic operator, ATM considers two properties of the learned features to actively select the tokens for mixing: 1) the semantics in different spatial positions may correspond to diverse scales and deformations; 2) different semantic attributes of a token would distribute in different channels (Bau et al. 2020; Wu, Lischinski, and Shechtman 2021). As illustrated in Fig. 1 (a), for a query, ATM actively predicts the locations offsets of tokens whose information should be incorporated for interaction. Particularly, ATM predicts the respective offset *channel-wisely* to select the context elements which are then recomposed to a new token. This empowers a more adaptive and flexible information interaction across tokens. We adopt this operation along the horizontal and vertical dimensions in parallel (Fig. 1 (b)), making such predictive context localization easier to be opti-

*Equal contribution. This work was done when Guoqiang Wei was an intern at MSRA.

mized. Then we learn to adaptively fuse the two recomposed tokens and the original query to be the output

The ATM can serve as a primary operator for constructing backbone architectures. To showcase this, we build a series of model variants with different model scales, named ATMNet-xT/T/S/B/L, respectively. ATMNet shows impressive effectiveness of ATM on a broad range of vision tasks as well as favorable scalability over different model scales. Besides, ATM can also serve as a plug-and-play enhanced replacement of the conventional convolution layers in FPN (Lin et al. 2017a) to enhance the pyramid feature learning for dense prediction tasks (object detection and segmentation).

Our contributions can be summarized below:

- We propose Active Token Mixer (**ATM**), a basic operator to efficiently enable content-adaptive and flexible global scope token mixing at channel level. It expands the range and reforms the way of message passing.
- We build an efficient vision backbone ATMNet with ATM as its primary ingredient for effective spatial information interaction. For the commonly used neck structure FPN, we build an enhanced FPN, *i.e.*, ATMFPN, powered by ATM, for dense prediction tasks.
- ATMNet achieves strong performance over different model scales and across various vision tasks. For image classification, only trained on ImageNet-1K, ATMNet achieves 82.0% top-1 accuracy with 27M parameters and reaches 84.8% when scaling up to 76M. Moreover, ATMNet outperforms recent prevalent backbones on dense prediction tasks by a significant margin with comparable or even less parameters and computation cost.

Related Work

CNN Based Models

Convolutional neural networks (CNNs) have been the mainstream architectures in computer vision for a long time. The CNN model is originally presented in (LeCun et al. 1998) for document recognition. Beginning with the significant success of AlexNet (Krizhevsky, Sutskever, and Hinton 2012) in ILSVRC 2012, various CNN-based architectures are designed or searched, *e.g.*, Inception (Szegedy et al. 2015, 2016, 2017), VGG (Simonyan and Zisserman 2015), ResNet (He et al. 2016), DenseNet (Huang et al. 2017), ResNeXt (Xie et al. 2017), EfficientNet (Tan and Le 2019), MNASNet, (Tan et al. 2019) and others (Wang et al. 2020; Ding et al. 2021; Liu et al. 2022b). In addition, there are a series of works dedicated to improving the convolution layers from different perspectives, *e.g.*, depthwise separable convolution (Chollet 2017; Howard et al. 2017; Sandler et al. 2018) for reduced computation costs and deformable convolution (Dai et al. 2017; Zhu et al. 2019) for objects of diverse shapes. It is noteworthy that the deformable convolution also allows learnable token selection for token mixing but ignores the semantic differences across channels (Bau et al. 2020; Wu, Lischinski, and Shechtman 2021) and usually suffers from optimization difficulties (Chan et al. 2021).

Self-Attention Based Models

(Kolesnikov et al. 2021) firstly introduces a pure self-attention based backbone to computer vision, *i.e.*, ViT, which achieves promising performance on image classification especially trained with extremely large-scale data. (Touvron et al. 2021b) improves the training strategy of ViT and proposes a knowledge distillation method, which helps ViT achieve higher performance trained only on ImageNet. Afterwards, various works endeavor to explore efficient vision Transformer architectures, *e.g.*, PVT (Wang et al. 2021b, 2022b), Swin (Liu et al. 2021, 2022a), Twins (Chu et al. 2021a), MViT (Yan et al. 2022; Li et al. 2021), and others (Chu et al. 2021b; Dong et al. 2021; Ali et al. 2021; Touvron et al. 2021c; Yang et al. 2021; Bertasius, Wang, and Torresani 2021; Li et al. 2022).

MLP-Like Models

Recently, MLP-like models have been reinvigorated. The pioneering works MLP-Mixer (Tolstikhin et al. 2021) and ResMLP (Touvron et al. 2021a) stack two types of MLP layers, *i.e.*, token-mixing MLP and channel-mixing MLP, alternately. The token-mixing MLP enables spatial information interaction over all tokens while the channel-mixing MLP mixes information across all channels within each token. ViP (Hou et al. 2022) and sMLP (Tang et al. 2022) encode the feature representations along two axial dimensions to improve MLPs' efficiency and capability. Shift (Wang et al. 2022a), ASMLP (Lian et al. 2022) and S^{2}MLP (Yu et al. 2022) perform spatial information mixing with spatial shift operations along different dimensions. CycleMLP (Chen et al. 2022), WaveMLP (Tang et al. 2021) and MorphMLP (Zhang et al. 2021) restrict the spatial information interaction within *hand-craft fixed local* windows in a *deterministic* way. As opposed to them, our ATM achieves *a learnable content-adaptive token-mixing*, which considers the diverse semantics attributed in different channels and spatial positions with global receptive fields, so that it can attain high flexibility and strong modeling capacity.

Method

A Unified Perspective of Token Mixing

For most prevailing model architectures, the input image is first patchified into a feature tensor $\mathbf{X} \in \mathbb{R}^{H\times W\times C}$ with the height H, the width W and the number of channels C. In vision tasks, token mixing is especially critical since the contextual information is inevitably required for understanding visual semantics. Before introducing our proposed method, we firstly review different token mixing mechanisms in the literature from a unified perspective. Mathematically, we formulate token mixing with a unified function:

$$f(\mathbf{X})|_{\mathbf{x}^q} = \sum_{k\in\mathcal{N}(\mathbf{x}^q)} \boldsymbol{\omega}^{k\rightarrow q} * g(\mathbf{x}^k), \qquad (1)$$

where $\mathbf{x}^q$ denotes the query token while $\mathcal{N}(\mathbf{x}^q)$ refers to a set of its contextual tokens. $\boldsymbol{\omega}^{k\rightarrow q}$ is the weight determining the degree of message passing from $\mathbf{x}^k$ to $\mathbf{x}^q$. $g(\cdot)$ is an embedding function. $*$ is a unified representation for element-wise or matrix multiplication.

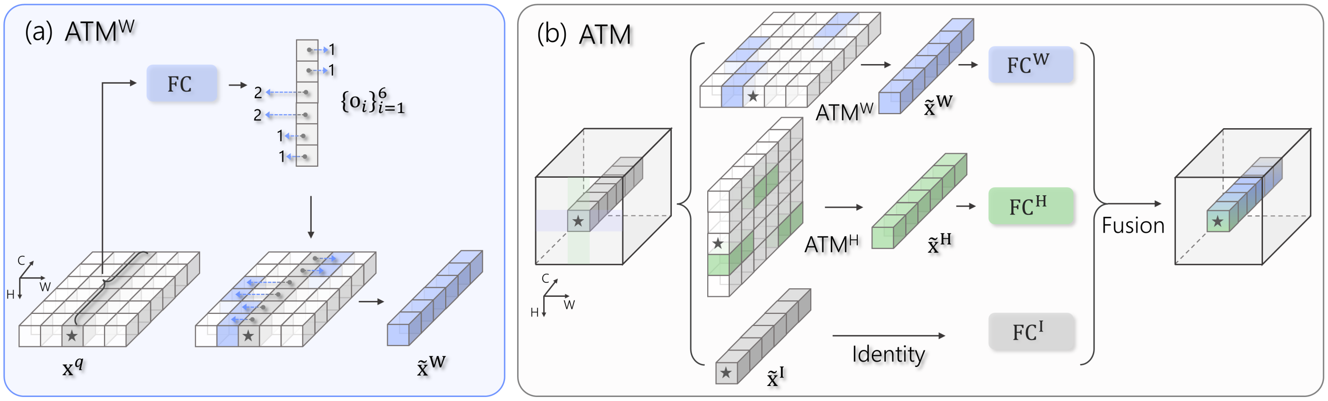

Figure 1: Illustration of our proposed Active Token Mixer (ATM). (a) ATM along the horizontal (width) dimension. For a query $\mathbf{x}^q$, ATM actively captures the useful contexts by recomposing the elements from selected tokens into $\tilde{\mathbf{x}}^W \in \mathbb{R}^C$ based on the learned channel-wise offsets. (b) ATM module consisting of ATM^W along horizontal dimension, ATM^H along vertical dimension, and the identity branch ATM^I. The two recomposed tokens ($\tilde{\mathbf{x}}^W$, $\tilde{\mathbf{x}}^H$) and the original $\tilde{\mathbf{x}}^I$ are then adaptively fused after being embedded by $FC^{\{W,H,I\}}$.

For conventional CNNs, $g(\cdot)$ is an identity function, and $\boldsymbol{\omega}^{k\to q} \in \mathbb{R}^{C\times C}$ corresponds to the convolutional kernels shared for different queries, and the message passing is restricted within a fixed-size sliding window $\mathcal{N}(\cdot)$. Transformers achieve a non-local $\mathcal{N}(\cdot)$ and adopt a computationally expensive $\boldsymbol{\omega}^{k\to q} \in \mathbb{R}^{C}$ through calculating the affinity between $\mathbf{x}^k$ and $\mathbf{x}^q$ in the embedding space. In recent MLP-like backbones (Chen et al. 2022; Touvron et al. 2021a; Tolstikhin et al. 2021; Zhang et al. 2021; Lian et al. 2022; Tang et al. 2022, 2021), $\mathcal{N}(\cdot)$ and $\boldsymbol{\omega}^{k\to q}$ are manually designed to perform token mixing in a *deterministic* way, leading to the *lack of content adaptivity*. In Transformers or MLPs, $g(\cdot)$ is commonly a learnable embedding function.

Active Token Mixer

Based on the token mixing methods detailed in Sec. , we have two key observations: 1) For the spatial dimension, visual objects/stuffs present diverse shapes and deformations. Therefore, information mixing within a fixed-range $\mathcal{N}(\cdot)$ (Touvron et al. 2021a; Chen et al. 2022; Tolstikhin et al. 2021; Lian et al. 2022) is inefficient and inadequate. The adaptive $\boldsymbol{\omega}^{k\to q}$ and $\mathcal{N}(\cdot)$ for message passing are desirable for extracting visual representations. 2) For the channel dimension, multiple semantic attributes carried in one token would distribute in its different channels (Bau et al. 2020; Wu, Lischinski, and Shechtman 2021). The token-level message passing with $\omega^{k\to q} \in \mathbb{R}$ shared over all channels can not treat different semantics adaptively and limits their full use, thus is less effective (Touvron et al. 2021a; Tolstikhin et al. 2021). In this work, *we pinpoint the importance of more fine-grained message passing for treating different semantics adaptively*.

To address the aforementioned limitations in existing token-mixing methods, we propose Active Token Mixer (ATM) as shown in Fig. 1. It first predicts the relative locations of useful contextual tokens along each direction at channel level, then learns to fuse the contextual tokens and query token. These two steps correspond to learn **where** the useful context tokens locate in and **how** to fuse them with the original information, respectively.

Drawing on the success of multi-branch design in (Hou et al. 2022; Chen et al. 2022; Lian et al. 2022), we propose a three-branch architecture for facilitating the context localization along different directions. Two branches are responsible for recomposing tokens into a new one along two axial directions separately as shown in Fig. 1 (b). In addition, we adopt an identity branch to preserve the original query information. The two recomposed tokens and query are further mixed as the final output.

ATM along the horizontal dimension We illustrate the ATM along the horizontal (width) dimension, denoted by ATM^W, in Fig. 1 (a). Given the query $\mathbf{x}^q \in \mathbb{R}^C$ (marked with $\star$), we first feed it into a FC layer to adaptively predict C offsets $\mathcal{O} = \{o_i\}_{i=1}^{C}$ for context localization. Note that we impose no constraint on the offset generation, thus $\mathcal{N}(\mathbf{x}^q)$ is allowed to be extended to all spatial positions along the horizontal direction. In this way, ATM can incorporate the information from the global scope, wherever needed, into $\mathbf{x}^q$ in a flexible and active manner. The predicted offsets determine the tokens in $\mathcal{N}(\cdot)$ per channel, which are used to recompose the selected tokens into a new token $\tilde{\mathbf{x}}^W \in \mathbb{R}^C$ as output of ATM^W:

$$\tilde{\mathbf{x}}^W = \left[\mathbf{X}_{[i,j+o_1,1]}, \mathbf{X}_{[i,j+o_2,2]}, \ldots, \mathbf{X}_{[i,j+o_C,C]}\right]^T, \quad (2)$$

where $\mathbf{X}_{[i,j+o,c]}$ denotes the c^{th} channel element of the token at spatial position $[i, j+o]$ where $[i, j]$ is the position of $\mathbf{x}^q$. ATM^W is capable of mixing information horizontally and globally into $\tilde{\mathbf{x}}^W$.

ATM along the vertical dimension Likewise, another ATM^H branch is adopted to recompose a token $\tilde{\mathbf{x}}^H$ along the vertical (height) dimension.

Fusion Here, we introduce how to fuse the recomposed $\tilde{\mathbf{x}}^W$, $\tilde{\mathbf{x}}^H$ and the original $\tilde{\mathbf{x}}^I$ into the final token-mixing re-

Model	Size	#P.(M)	FLOPs(G)	Top-1(%)
ResNet18	224^2	12	1.8	69.8
ResMLP-S12	224^2	15	3.0	76.6
CycleMLP-B1	224^2	15	2.1	78.9
ATMNet-xT	224^2	15	2.2	**79.7**
ResNet50	224^2	26	4.1	78.5
Deit-S	224^2	22	4.6	79.8
PVT-S	224^2	25	3.8	79.8
Swin-T	224^2	29	4.6	81.2
TwinsP-S	224^2	24	3.8	81.2
Twins-S	224^2	24	2.9	81.7
ResMLP-S24	224^2	30	6.0	79.4
ASMLP-T	224^2	28	4.4	81.3
ViP-S	224^2	25	6.9	81.5
MorphMLP-T	224^2	23	3.9	81.6
CycleMLP-B2	224^2	27	3.9	81.6
Shift-T	224^2	29	4.5	81.7
ATMNet-T	224^2	27	4.0	**82.0**
PVT-M	224^2	44	6.7	81.2
TwinsP-B	224^2	44	6.7	82.7
MorphMLP-S	224^2	38	7.0	82.6
CycleMLP-B3	224^2	38	6.9	82.6
ATMNet-S	224^2	39	6.9	**83.1**
PVT-L	224^2	61	9.8	81.7
Swin-S	224^2	50	8.7	83.2
Twins-B	224^2	56	8.6	83.2
ViP-M	224^2	55	16.3	82.7
Shift-S	224^2	50	8.8	82.8
CycleMLP-B4	224^2	52	10.1	83.0
ASMLP-S	224^2	50	8.5	83.1
MorphMLP-B	224^2	58	10.2	83.2
ATMNet-B	224^2	52	10.1	**83.5**
Deit-B	224^2	86	17.5	81.8
Swin-B	224^2	88	15.4	83.5
S^2MLP-W	224^2	71	14.0	80.0
CycleMLP-B5	224^2	76	15.3	83.1
ViP-L	224^2	88	24.4	83.2
Shift-B	224^2	89	15.6	83.3
ASMLP-B	224^2	88	15.2	83.3
MorphMLP-L	224^2	76	12.5	83.4
ATMNet-L	224^2	76	12.3	**83.8**
ViT-B/16↑	384^2	86	55.4	77.9
Deit-B↑	384^2	86	55.4	83.1
Swin-B↑	384^2	88	47.1	84.5
ATMNet-L↑	384^2	76	36.4	**84.8**

Table 1: Comparisons with state-of-the-art models on ImageNet-1K without extra data. All models are trained with input size of 224×224, except ↑ with 384×384.

sult. First, we adopt three FC layers $FC^{\{W,H,I\}}$ to embed $\tilde{\mathbf{x}}^{\{W,H,I\}}$ to $\hat{\mathbf{x}}^{\{W,H,I\}}$, respectively, which are then mixed with learned weights, formulated as:

$$\hat{\mathbf{x}} = \boldsymbol{\alpha}^{W} \odot \hat{\mathbf{x}}^{W} + \boldsymbol{\alpha}^{H} \odot \hat{\mathbf{x}}^{H} + \boldsymbol{\alpha}^{I} \odot \hat{\mathbf{x}}^{I}, \tag{3}$$

where $\odot$ denotes element-wise multiplication. $\boldsymbol{\alpha}^{\{W,H,I\}} \in \mathbb{R}^{C}$ are learned from the summation $\hat{\mathbf{x}}^{\Sigma}$ of $\hat{\mathbf{x}}^{\{W,H,I\}}$ with $W^{\{W,H,I\}} \in \mathbb{R}^{C \times C}$:

$$[\boldsymbol{\alpha}^{W}, \boldsymbol{\alpha}^{H}, \boldsymbol{\alpha}^{I}] = \sigma([W^{W} \cdot \hat{\mathbf{x}}^{\Sigma}, W^{H} \cdot \hat{\mathbf{x}}^{\Sigma}, W^{I} \cdot \hat{\mathbf{x}}^{\Sigma}]), \tag{4}$$

where $\sigma(\cdot)$ is a *softmax* function for normalizing each channel separately.

Discussion Our ATM has three hallmarks: 1) *Content adaptivity.* The context selection/localization is adaptively learned for the query token in an active way, instead of being passively determined by manual designed rules (Chen et al. 2022; Lian et al. 2022; Yu et al. 2022; Zhang et al. 2021). 2) *Flexibility.* In general, different channels are characterized with different semantics. Our proposed ATM enables to dynamically select context tokens at the channel level from a global range $\mathcal{N}(\cdot)$, adaptive to visual contents with various scales and deformations. 3) *Efficiency.* By incorporating contexts from C tokens into the two recomposed tokens, the computation complexity of ATM is $\mathcal{O}\left(HWC^{2}\right)$, which is linear with the input resolution and is agnostic to the receptive fields, making it computation-friendly to larger-size images used in object detection and segmentation tasks.

Compared with the conventional convolutions, ATM is able to enlarge its receptive field to global-scope flexibly with constant computation cost. Compared with the multi-head self-attention in Transformers, ATM globally mixes token information per channel with the actively learned offsets, avoiding the computation-consuming attention calculation. ATM may be reminiscent of the deformable convolution (Dai et al. 2017; Zhu et al. 2019). In fact, there are two crucial differences: 1) The learned offsets in deformable convolutions are shared over all channels, without consideration on semantic differences across channels. Our ATM can incorporate contextual information in channel wise, achieving a more flexible and fine-grained context exploitation mechanism in token mixing. 2) We decouple the learning of context localization along different directions, making ATMNet easier to be optimized.

Model Architectures

ATM Block We build our ATMNet by stacking multiple ATM blocks in sequence. Here, we introduce the architecture of an ATM block. For the output $\mathbf{X}^{l\text{-}1}$ of the $(l-1)$-th block ATM$^{l\text{-}1}$, we feed it to the l-th block ATM^{l} for token mixing. Further, we use an MLP module to further modulate the feature along its channel dimension. Skip connections are adopted to facilitate the training. The entire process can be formulated as:

$$\hat{\mathbf{X}}^{l} = ATM^{l}(LN(\mathbf{X}^{l\text{-}1})) + \mathbf{X}^{l\text{-}1}, \tag{5}$$

$$\mathbf{X}^{l} = MLP^{l}(LN(\hat{\mathbf{X}}^{l})) + \hat{\mathbf{X}}^{l}, \tag{6}$$

where LN is LayerNorm (Ba, Kiros, and Hinton 2016).

UperNet			
Model	#P	FLOPs	mIoU/mIoU(ms)
Swin-T	60	945	44.5 / 45.8
Twins-S	54	931	46.2 / 47.1
ConvNeXt-T	60	939	- / 46.7
ASMLP-T	60	937	- / 46.5
CycleMLP-T	60	937	- / 47.1
ATMNet-T	57	927	**46.5 / 47.6**
Swin-B	121	1188	48.1 / 49.7
Twins-L	133	1236	48.8 / 50.2
ConvNeXt-T	122	1170	- / 49.9
ASMLP-B	121	1166	- / 49.5
CycleMLP-B	121	1166	- / 49.7
ATMNet-S	69	988	48.4 / 49.5
ATMNet-L	108	1106	**50.1 / 51.1**

Semantic FPN			
Model	#P	FLOPs	mIoU
Swin-T	31.9	48	41.5
Twins-S	28.3	37	43.2
TwinsP-S	28.4	40	44.3
MorphMLP-T	26.4	-	43.0
CycleMLP-B2	30.6	42	43.4
Wave-MLP-S	31.2	-	44.4
ATMNet-T	30.9	42.4	**45.8**
Swin-B	91.2	107	46.0
TwinsP-L	65.3	71	46.4
Twins-L	103.7	102	46.7
CycleMLP-B5	79.4	86	45.5
MorphMLP-B	59.3	-	45.9
ATMNet-L	79.8	86.6	**48.1**

Table 2: Semantic segmentation results on ADE20K `val` with UperNet and Semantic FPN. FLOPs are evaluated on 512×2048 for UperNet and 512×512 for Semantic FPN. All backbones are pretrained on ImageNet-1K. mIoU(ms): mIoU with multi-scale inference. The results of other variants are in the Supplementary.

ATMNet Following the typical hierarchical architecture designs (He et al. 2016; Liu et al. 2021), we provide five four-stage backbone architecture variants with different channel dimensions and numbers of the ATM blocks, which are ATMNet-xT/T/S/B/L, respectively. Note that the offset generation layer is shared across the tokens within each ATM branch. Here, the awareness of the position of query token can facilitate offsets prediction. We thus introduce one positional encoding generator (PEG) (Chu et al. 2021b) for each stage before ATM, which helps a little for dense prediction tasks. More details are placed in the supplementary.

ATMFPN In addition to the strong capability of constructing vision backbones, ATM is also an enhanced alternative for conventional convolutions in convolution-based decoders for downstream tasks. We replace the convolutions in the prevailing FPN (Lin et al. 2017b), which is widely applied as the neck for object detection and segmentation, with our ATM and name this new neck as **ATMFPN**. We demonstrate the effectiveness of our ATMFPN in Table 4.

Experiments

ImageNet-1K Classification

Settings We train our models on the ImageNet-1K dataset (Deng et al. 2009) from scratch. All models are trained with input size of 224×224 for 300 epochs with the batch size of 1024. The ATMNet-L is finetuned with input size of 384×384 for 30 epochs. More details are shown in the Supplementary.

Results We report the top-1 accuracy comparison between our ATMNet with recent CNN-, Transformer- and MLP-based backbones in Table 1, where all methods are categorized into different groups w.r.t. the model size (#Parameters) and computation complexity (FLOPs). All our different variants achieve higher accuracy compared with the scale-comparable methods. 1) Our ATMNet-T, -B, and -L variants outperform the prominent Transformer Swin-T, -S, and -B by +0.8%, +0.3% and +0.3% with comparable parameters and FLOPs. For larger models, ATMNet-L↑ surpasses Swin-B↑ with **-23%** computation cost. 2) Our ATMNet also surpasses all recent MLP-like backbones (ASMLP, CycleMLP, ViP, and *etc*). Compared with the recent CycleMLP mixing tokens in a *deterministic and local* manner, our five variants outperforms the corresponding CycleMLP variants by +0.8%, 0.5%, +0.4%, +0.5% and +0.7% respectively, with comparable computation cost.

Note that some MLP-like backbones (*e.g.*, MLP-Mixer, ResMLP, gMLP, ResMLP, ViP, sMLP and *etc*) in Table 1 are not validated in downstream dense prediction tasks, where the most architectures are not compatible with various input resolutions. In contrast, our ATMNet is capable of dealing with different input scales, and shows pronounced performance on dense prediction tasks, which will be shown in the following sections.

Semantic Segmentation

Settings Following the common practice (Chu et al. 2021a; Lian et al. 2022), we evaluate the potential of ATMNet on the challenging semantic segmentation task on ADE20K (Zhou et al. 2019). We adopt two widely used frameworks, Semantic FPN (Kirillov et al. 2019) and UperNet (Xiao et al. 2018). More experimental details can be found in the Supplementary.

Results The results on top of UperNet and Semantic FPN are shown in Table 2. For different model scales, ATMNet outperforms all previous methods with comparable computation costs. The largest ATMNet-L with Sematic FPN outperforms previous state-of-the-art Twins-L by **+1.4**

Backbone	#Params.	FLOPs	Mask R-CNN 1×						Mask R-CNN 3× MS					
	(M)	(G)	AP^b	AP$^b_{50}$	AP$^b_{75}$	AP^m	AP$^m_{50}$	AP$^m_{75}$	AP^b	AP$^b_{50}$	AP$^b_{75}$	AP^m	AP$^m_{50}$	AP$^m_{75}$
ResNet-50	44	260	38.0	58.6	41.4	34.4	55.1	36.7	41.0	61.7	44.9	37.1	58.4	40.1
Swin-T	48	264	42.2	64.6	46.2	39.1	61.6	42.0	46.0	68.2	50.2	41.6	65.1	44.8
ConvNext-T	48	262	-	-	-	-	-	-	46.2	67.9	50.8	41.7	65.0	44.9
ASMLP-T	48	260	-	-	-	-	-	-	46.0	67.5	50.7	41.5	64.6	44.5
CycleMLP-B2	47	250	42.1	64.0	45.7	38.9	61.2	41.8	-	-	-	-	-	-
WaveMLP-S	47	250	44.0	65.8	48.2	40.0	63.1	42.9	-	-	-	-	-	-
ATMNet-xT	35	215	42.8	64.9	46.9	39.5	62.1	42.5	45.0	67.4	49.5	41.1	64.4	44.2
ATMNet-T	47	251	**44.8**	**66.9**	**49.0**	**41.0**	**64.2**	**44.3**	**47.1**	**69.0**	**51.7**	**42.7**	**66.5**	**46.0**
X101-64	102	493	42.8	63.8	47.3	38.4	60.6	41.3	44.4	64.9	48.8	39.7	61.9	42.6
Twins-L	120	474	45.9	-	-	41.6	-	-	-	-	-	-	-	-
Swin-B	107	496	45.5	-	-	41.3	-	-	48.5	69.8	53.2	43.4	66.8	46.9
MViT-B	73	438	-	-	-	-	-	-	48.8	71.2	53.5	44.2	68.4	47.6
CycleMLP-B5	95	421	44.1	65.5	48.4	40.1	62.8	43.0	-	-	-	-	-	-
WaveMLP-B	75	353	45.7	67.5	50.1	27.8	49.2	59.7	-	-	-	-	-	-
ATMNet-B	72	377	46.5	68.6	51.0	42.5	66.1	45.8	49.0	70.7	54.0	43.9	67.7	47.5
ATMNet-L	96	424	**47.4**	**69.9**	**52.0**	**43.2**	**67.3**	**46.5**	**49.5**	**71.5**	**54.3**	**44.5**	**68.7**	**48.1**

Table 3: Object detection results on COCO `val2017` with Mask R-CNN 1× and RetinaNet 1×. FLOPS are evaluated with resolution 800×1280. The complete comparison table and results of 3× can be found in the Supplementary.

mIoU with **-23%** parameters and **-16%** FLOPs. ATMNet-L also achieves the new state-of-the-art (**51.1** ms mIoU) with UperNet, which surpasses the representative network Swin-B by **+1.4** mIoU with **-10%** parameters. Note that ATMNet-S achieves comparable performance with Swin-B, but only requires about **-50%** parameters.

It also shows that most previous MLP-like backbones (*e.g.*, CycleMLP, ASMLP, MorphMLP) perform better than Transformer-based Swin/Twins for smaller models, but lag behind them for larger models. These *manually designed* token mixing methods within them leads to remarkable limitations in exploring rich feature patterns, while the global-scope attention in Transformers allows extracting better features as model scaling up. In contrast, ATMNet shows its strong capability and scalability on segmentation over different model scales, especially for the large-scale models. The superiority of ATMNet lies in the flexibility of ATM, which provides great capability to exploit sufficient features from visual signals with various scales and deformations, especially for the pixel-level tasks heavily relying on spatial information interaction.

Object Detection

Settings We further evaluate the performance of our ATMNet on object detection task on the COCO (Lin et al. 2014) dataset. We adopt three detection frameworks (Mask R-CNN, RetinaNet and Cascade Mask R-CNN) and report the 1×/3× (MS) schedule results on COCO 2017 `val`. Detailed configurations can be found in the Supplementary.

Results The object detection results for Mask R-CNN 1× and 3×(MS) are shown in Table 3. Thanks to ATM's flexibility and effectiveness for token mixing, our ATMNet obtains promising results on the challenging object detection. ATMNet achieves the state-of-the-art for the most model scales with different detectors. For the Mask R-CNN 1× setting, our different model variants outperform the corresponding parameter-comparable Swin variants by **+2.6/+1.9** and **+1.9/+1.9** mAP^b/mAP^m respectively, which demonstrates the ATMNet's superiority on dense prediction task, where the input is usually with larger resolution. For the largest models, ATMNet-L surpasses the state-of-the-art Twins-L by **+1.5** mAP^b with **-20%** parameters. The comparisons with RetinaNet 1×/3× and Cascade Mask R-CNN 1×/3× can be found in the Supplementary.

		Semantic FPN		Mask R-CNN	
Backbone	Neck	FLOPS	mIoU	FLOPS	AP^b
ResNet-50	FPN	45.9	37.3	259.8	38.0
	ATMFPN	48.9	40.3$_{\uparrow 3.0}$	298.9	39.9$_{\uparrow 1.9}$
Swin-Tiny	FPN	47.5	41.5	267.0	42.2
	ATMFPN	47.5	43.7$_{\uparrow 2.2}$	267.1	43.5$_{\uparrow 1.3}$
ATMNet-T	FPN	42.4	45.8	251.1	44.8
	ATMFPN	41.4	46.5$_{\uparrow 0.7}$	247.0	45.6$_{\uparrow 0.8}$
ATMNet-L	FPN	86.6	48.1	423.7	47.4
	ATMFPN	86.6	48.3$_{\uparrow 0.2}$	423.8	48.4$_{\uparrow 1.0}$

Table 4: FPN/ATMFPN for semantic segmentation with Semantic FPN and object detection with Mask R-CNN 1×.

ATMFPN

Our proposed ATM can be adopted not only for constructing vision backbones, but also as an enhanced alternative for convolution-based decoders. Based on FPN (Lin et al. 2017b), we build an ATMFPN neck with ATM, and report

ID	Model	FLOPs	INT	COCO	ADE20K
①	Baseline	3.890	79.3	36.0	37.9
②	ATMNet w/o PEG	3.924	$82.0_{\uparrow 2.7}$	$43.4_{\uparrow 7.4}$	$45.6_{\uparrow 7.7}$
③	ATMNet	3.972	$82.0_{\uparrow 2.7}$	$43.6_{\uparrow 7.6}$	$45.8_{\uparrow 7.9}$

Table 5: Ablation study. FLOPs are obtained on ImageNet-1K. COCO: AP^b for RetinaNet 1×. ADE20K: mIoU for Semantic FPN.

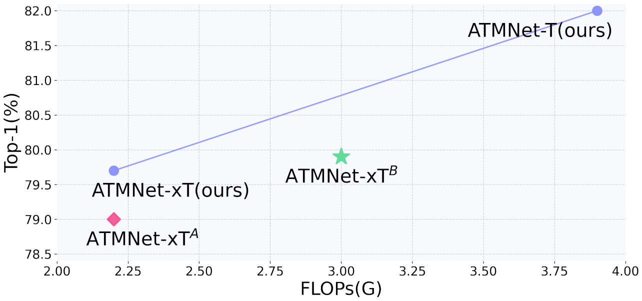

Figure 2: Comparisons of different offset configurations on ImageNet-1K. ATMNet-xT^A(♦): offset learning is not decoupled along different directions, *i.e.*, the selected contextual tokens are directly recomposed as $\tilde{\mathbf{x}} = [\mathbf{X}_{[i+oh_1, j+ow_1, 1]}, \mathbf{X}_{[i+oh_2, j+ow_2, 2]}, \dots]^T$ with the predicted offsets $\{oh_c, ow_c\}$. ATMNet-xT^B(★): the number of selected contextual tokens per channel is extended from 1 to 3 for each direction.

the results on different backbones for object detection and semantic segmentation in Table 4. With comparable computation cost, ResNet-50 with ATMFPN outperforms the naïve FPN by **+3.0** mIoU/**+1.9** AP^b for segmentation and object detection respectively. ATMFPN also helps improve the performance for the backbone of Swin and ATMNet. Thanks to the flexibility, our proposed ATM is basically applicable for extracting better visual feature representations.

Ablation Study and Analysis

Effectiveness of ATM Table 5 shows our ablation results. In the baseline ① of ATMNet, all offsets are fixed to 0, which means there is no spatial information interaction between different tokens in ①. This baseline achieves 79.3% accuracy on ImageNet-1K while its performance on dense prediction tasks is severely bounded due to the lack of adequate spatial interaction. This also validates that token mixing is sorely vital for dense prediction tasks. With our proposed ATMNet w/o PEG (②), the classification accuracy is improved by **+2.7%**, and the performance on the dense prediction task is *significantly improved by a large margin* (**+7.4** mAP^b on COCO and **+7.7** mIoU on ADE20K). Our proposed ATM brings sufficient information mixing to help extract more powerful features with negligible additional computation overhead. The PEG module is introduced for providing position information for offset generation, which helps a little for dense tasks.

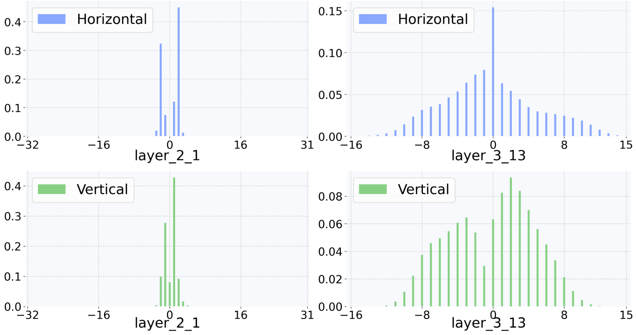

Figure 3: Histograms of learned offsets for center token of different layers, counted on all samples of ADE20K `val`. `layer_i_j`: the jth layer of the ith stage.

Comparison with other offset configurations 1) *Effectiveness of directional decomposition.* As show in Fig. 2, our ATMNet-xT is clearly superior to ATMNet-xT^A(♦) with very close FLOPs, demonstrating the effectiveness of directional decomposition during predicting offsets. 2) *The number of selected contextual tokens.* The ATMNet-xT^B(★) with more contextual tokens for each query outperforms ATMNet-xT by 0.2% but with **+50%** additional computation cost. This shows our ATM is a better trade-off between the computation cost and the final performance as an efficient and effective token mixer.

Analyses of learned offsets We investigate the distributions of the learned offsets via the histograms of offsets w.r.t. the center token in Fig. 3. We observe: 1) As the depth increases, the learned offsets expand to a larger range. This is in line with the conclusion in (Raghu et al. 2021; Zhang et al. 2021) that local receptive fields in shallower layers are conductive to training vision models, while the long-range information is required for deeper layers. 2) For a query token, the learned offsets differ for different channels and such flexibility enables efficient semantic-adaptive information interaction. 3) Besides the network depth, the learned offsets of ATM are also adaptive to different datasets or tasks (shown in the Supplementary), endowing ATMNet with higher flexibility and better adaptivity. This observation indicates that mixing tokens with *hand-crafted and deterministic* rules is in fact insufficient to model the various distributions of different datasets. More results are in the Supplementary.

Conclusion

In this work, we propose an innovative token mixing mechanism, ATM, which actively and meticulously learns to fuse content-adaptive contextual information in the global scope. With the proposed basic operator, we build a general vision backbone ATMNet for various vision tasks and an enhanced FPN, *i.e.*, ATMFPN for dense prediction tasks. ATMNet is capable of flexibly and effciently capturing diverse visual patterns. Comprehensive experiments demonstrate our ATMNet is generally applicable and effective for various vision tasks including image classification, object detection and semantic segmentation.

References

Ali, A.; Touvron, H.; Caron, M.; Bojanowski, P.; Douze, M.; Joulin, A.; Laptev, I.; Neverova, N.; Synnaeve, G.; Verbeek, J.; et al. 2021. Xcit: Cross-covariance image transformers. *NeurIPS*, 34.

Ba, J. L.; Kiros, J. R.; and Hinton, G. E. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Bau, D.; Zhu, J.-Y.; Strobelt, H.; Lapedriza, A.; Zhou, B.; and Torralba, A. 2020. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48): 30071–30078.

Bertasius, G.; Wang, H.; and Torresani, L. 2021. Is space-time attention all you need for video understanding. *ICML*, 2(3): 4.

Carion, N.; Massa, F.; Synnaeve, G.; Usunier, N.; Kirillov, A.; and Zagoruyko, S. 2020. End-to-end object detection with transformers. In *ECCV*, 213–229. Springer.

Chan, K. C.; Wang, X.; Yu, K.; Dong, C.; and Loy, C. C. 2021. Understanding deformable alignment in video super-resolution. In *AAAI*, volume 35, 973–981.

Chen, S.; Xie, E.; GE, C.; Chen, R.; Liang, D.; and Luo, P. 2022. CycleMLP: A MLP-like Architecture for Dense Prediction. In *ICLR*.

Cheng, B.; Schwing, A.; and Kirillov, A. 2021. Per-pixel classification is not all you need for semantic segmentation. In *NeurIPS*, volume 34.

Chollet, F. 2017. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, 1251–1258.

Chu, X.; Tian, Z.; Wang, Y.; Zhang, B.; Ren, H.; Wei, X.; Xia, H.; and Shen, C. 2021a. Twins: Revisiting the design of spatial attention in vision transformers. *NeurIPS*, 34.

Chu, X.; Tian, Z.; Zhang, B.; Wang, X.; Wei, X.; Xia, H.; and Shen, C. 2021b. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*.

Dai, J.; Qi, H.; Xiong, Y.; Li, Y.; Zhang, G.; Hu, H.; and Wei, Y. 2017. Deformable convolutional networks. In *ICCV*, 764–773.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR*, 248–255. Ieee.

Ding, X.; Zhang, X.; Ma, N.; Han, J.; Ding, G.; and Sun, J. 2021. Repvgg: Making vgg-style convnets great again. In *CVPR*, 13733–13742.

Dong, X.; Bao, J.; Chen, D.; Zhang, W.; Yu, N.; Yuan, L.; Chen, D.; and Guo, B. 2021. Cswin transformer: A general vision transformer backbone with cross-shaped windows. *arXiv preprint arXiv:2107.00652*.

He, K.; Chen, X.; Xie, S.; Li, Y.; Dollár, P.; and Girshick, R. 2021a. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *CVPR*, 770–778.

He, S.; Luo, H.; Wang, P.; Wang, F.; Li, H.; and Jiang, W. 2021b. Transreid: Transformer-based object re-identification. In *ICCV*, 15013–15022.

Hou, Q.; Jiang, Z.; Yuan, L.; Cheng, M.-M.; Yan, S.; and Feng, J. 2022. Vision permutator: A permutable mlp-like architecture for visual recognition. *TPAMI*, 45(1): 1328–1334.

Howard, A. G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; and Adam, H. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *CVPR*, 4700–4708.

Kirillov, A.; Girshick, R.; He, K.; and Dollár, P. 2019. Panoptic feature pyramid networks. In *CVPR*, 6399–6408.

Kolesnikov, A.; Dosovitskiy, A.; Weissenborn, D.; Heigold, G.; Uszkoreit, J.; Beyer, L.; Minderer, M.; Dehghani, M.; Houlsby, N.; Gelly, S.; Unterthiner, T.; and Zhai, X. 2021. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 25.

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

Li, K.; Wang, Y.; Gao, P.; Song, G.; Liu, Y.; Li, H.; and Qiao, Y. 2022. Uniformer: Unified Transformer for Efficient Spatiotemporal Representation Learning. *ICLR*.

Li, Y.; Wu, C.-Y.; Fan, H.; Mangalam, K.; Xiong, B.; Malik, J.; and Feichtenhofer, C. 2021. Improved multiscale vision transformers for classification and detection. *arXiv preprint arXiv:2112.01526*.

Lian, D.; Yu, Z.; Sun, X.; and Gao, S. 2022. As-mlp: An axial shifted mlp architecture for vision. *ICLR*.

Lin, K.; Wang, L.; and Liu, Z. 2021. End-to-end human pose and mesh reconstruction with transformers. In *CVPR*, 1954–1963.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017a. Feature pyramid networks for object detection. In *CVPR*, 2117–2125.

Lin, T.-Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; and Belongie, S. 2017b. Feature pyramid networks for object detection. In *CVPR*, 2117–2125.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *ECCV*, 740–755. Springer.

Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; Wei, F.; and Guo, B. 2022a. Swin Transformer V2: Scaling Up Capacity and Resolution. In *CVPR*.

Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; and Guo, B. 2021. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 10012–10022.

Liu, Z.; Mao, H.; Wu, C.-Y.; Feichtenhofer, C.; Darrell, T.; and Xie, S. 2022b. A ConvNet for the 2020s. *CVPR*.

Raghu, M.; Unterthiner, T.; Kornblith, S.; Zhang, C.; and Dosovitskiy, A. 2021. Do vision transformers see like convolutional neural networks? *NeurIPS*, 34.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 4510–4520.

Simonyan, K.; and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. *ICLR*.

Szegedy, C.; Ioffe, S.; Vanhoucke, V.; and Alemi, A. A. 2017. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *CVPR*, 1–9.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *CVPR*, 2818–2826.

Tan, M.; Chen, B.; Pang, R.; Vasudevan, V.; Sandler, M.; Howard, A.; and Le, Q. V. 2019. Mnasnet: Platform-aware neural architecture search for mobile. In *CVPR*, 2820–2828.

Tan, M.; and Le, Q. 2019. Efficientnet: Rethinking model scaling for convolutional neural networks. In *ICML*, 6105–6114. PMLR.

Tang, C.; Zhao, Y.; Wang, G.; Luo, C.; Xie, W.; and Zeng, W. 2022. Sparse MLP for Image Recognition: Is Self-Attention Really Necessary? *AAAI*.

Tang, Y.; Han, K.; Guo, J.; Xu, C.; Li, Y.; Xu, C.; and Wang, Y. 2021. An image patch is a wave: Phase-aware vision mlp. *arXiv preprint arXiv:2111.12294*.

Tolstikhin, I. O.; Houlsby, N.; Kolesnikov, A.; Beyer, L.; Zhai, X.; Unterthiner, T.; Yung, J.; Steiner, A.; Keysers, D.; Uszkoreit, J.; et al. 2021. Mlp-mixer: An all-mlp architecture for vision. *NeurIPS*, 34.

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

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jegou, H. 2021b. Training data-efficient image transformer distillation through attention. In *ICML*, volume 139, 10347–10357.

Touvron, H.; Cord, M.; Sablayrolles, A.; Synnaeve, G.; and Jégou, H. 2021c. Going deeper with image transformers. In *ICCV*, 32–42.

Wang, G.; Zhao, Y.; Tang, C.; Luo, C.; and Zeng, W. 2022a. When Shift Operation Meets Vision Transformer: An Extremely Simple Alternative to Attention Mechanism. *AAAI*.

Wang, J.; Sun, K.; Cheng, T.; Jiang, B.; Deng, C.; Zhao, Y.; Liu, D.; Mu, Y.; Tan, M.; Wang, X.; et al. 2020. Deep high-resolution representation learning for visual recognition. *TPAMI*, 43(10): 3349–3364.

Wang, N.; Zhou, W.; Wang, J.; and Li, H. 2021a. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *CVPR*, 1571–1580.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2021b. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 568–578.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2022b. Pvtv2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3): 1–10.

Wu, Z.; Lischinski, D.; and Shechtman, E. 2021. Stylespace analysis: Disentangled controls for stylegan image generation. In *CVPR*, 12863–12872.

Xiao, T.; Liu, Y.; Zhou, B.; Jiang, Y.; and Sun, J. 2018. Unified perceptual parsing for scene understanding. In *ECCV*, 418–434.

Xie, E.; Wang, W.; Yu, Z.; Anandkumar, A.; Alvarez, J. M.; and Luo, P. 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. In *NeurIPS*, volume 34.

Xie, S.; Girshick, R.; Dollár, P.; Tu, Z.; and He, K. 2017. Aggregated residual transformations for deep neural networks. In *CVPR*, 1492–1500.

Yan, S.; Xiong, X.; Arnab, A.; Lu, Z.; Zhang, M.; Sun, C.; and Schmid, C. 2022. Multiview Transformers for Video Recognition. *arXiv preprint arXiv:2201.04288*.

Yang, J.; Li, C.; Zhang, P.; Dai, X.; Xiao, B.; Yuan, L.; and Gao, J. 2021. Focal self-attention for local-global interactions in vision transformers. *NeurIPS*.

Yu, T.; Li, X.; Cai, Y.; Sun, M.; and Li, P. 2022. S2-mlp: Spatial-shift mlp architecture for vision. In *WACV*, 297–306.

Zhang, D. J.; Li, K.; Chen, Y.; Wang, Y.; Chandra, S.; Qiao, Y.; Liu, L.; and Shou, M. Z. 2021. MorphMLP: A Self-Attention Free, MLP-Like Backbone for Image and Video. *arXiv preprint arXiv:2111.12527*.

Zhou, B.; Zhao, H.; Puig, X.; Xiao, T.; Fidler, S.; Barriuso, A.; and Torralba, A. 2019. Semantic understanding of scenes through the ade20k dataset. *IJCV*, 127(3): 302–321.

Zhu, X.; Hu, H.; Lin, S.; and Dai, J. 2019. Deformable convnets v2: More deformable, better results. In *CVPR*, 9308–9316.