This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

RMT: Retentive Networks Meet Vision Transformers

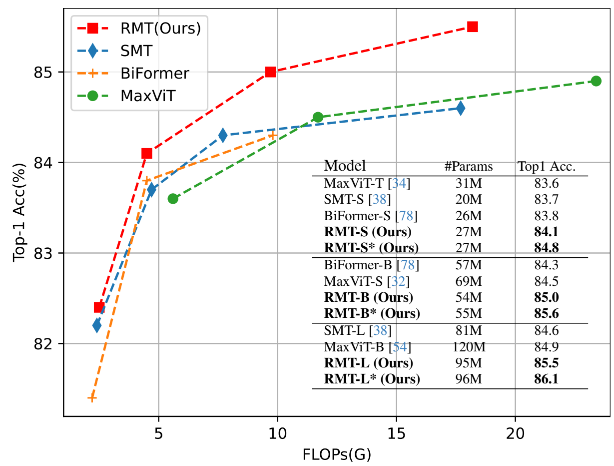
Qihang Fan [1,2], Huaibo Huang[1], Mingrui Chen[1,2], Hongmin Liu[3], Ran He[1,2]*
[1]MAIS & CRIPAC, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[2]School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[3]University of Science and Technology Beijing, Beijing, China
fanqihang.159@gmail.com, huaibo.huang@cripac.ia.ac.cn,
charmier@hust.edu.cn, hmliu_82@163.com, rhe@nlpr.ia.ac.cn

Abstract

*Vision Transformer (ViT) has gained increasing attention in the computer vision community in recent years. However, the core component of ViT, Self-Attention, lacks explicit spatial priors and bears a quadratic computational complexity, thereby constraining the applicability of ViT. To alleviate these issues, we draw inspiration from the recent Retentive Network (RetNet) in the field of NLP, and propose RMT, a strong vision backbone with explicit spatial prior for general purposes. Specifically, we extend the RetNet's temporal decay mechanism to the spatial domain, and propose a spatial decay matrix based on the Manhattan distance to introduce the explicit spatial prior to Self-Attention. Additionally, an attention decomposition form that adeptly adapts to explicit spatial prior is proposed, aiming to reduce the computational burden of modeling global information without disrupting the spatial decay matrix. Based on the spatial decay matrix and the attention decomposition form, we can flexibly integrate explicit spatial prior into the vision backbone with linear complexity. Extensive experiments demonstrate that RMT exhibits exceptional performance across various vision tasks. Specifically, without extra training data, RMT achieves **84.8%** and **86.1%** top-1 acc on ImageNet-1k with **27M/4.5GFLOPs** and **96M/18.2GFLOPs**. For downstream tasks, RMT achieves **54.5** box AP and **47.2** mask AP on the COCO detection task, and **52.8** mIoU on the ADE20K semantic segmentation task.*

1. Introduction

Vision Transformer (ViT) [13] is an excellent visual architecture highly favored by researchers. However, as the core module of ViT, Self-Attention's inherent structure lacking explicit spatial priors. Besides, the quadratic complexity



Model	#Params	Top1 Acc.
MaxViT-T [34]	31M	83.6
SMT-S [38]	20M	83.7
BiFormer-S [78]	26M	83.8
RMT-S (Ours)	27M	**84.1**
RMT-S* (Ours)	27M	**84.8**
BiFormer-B [78]	57M	84.3
MaxViT-S [32]	69M	84.5
RMT-B (Ours)	54M	**85.0**
RMT-B* (Ours)	55M	**85.6**
SMT-L [38]	81M	84.6
MaxViT-B [54]	120M	84.9
RMT-L (Ours)	95M	**85.5**
RMT-L* (Ours)	96M	**86.1**

Figure 1. FLOPs v.s. Top-1 accuracy on ImageNet-1K with 224×224 resolution. "*" indicates the model trained with token labeling [30].

of Self-Attention leads to significant computational costs when modeling global information. These issues limit the application of ViT.

Many works have previously attempted to alleviate these issues [14, 18, 33, 39, 53, 60, 64]. For example, in Swin Transformer [39], the authors partition the tokens used for self-attention by applying windowing operations. This operation not only reduces the computational cost of self-attention but also introduces spatial priors to the model through the use of windows and relative position encoding. In addition to it, NAT [21] changes the receptive field of Self-Attention to match the shape of convolution, reducing computational costs while also enabling the model to perceive spatial priors through the shape of its receptive field.

Different from previous methods, we draw inspiration from the recently successful Retentive Network (RetNet) [50] in the field of NLP. RetNet utilizes a distance-dependent temporal decay matrix to provide explicit temporal prior for one-dimensional and unidirectional text data. ALiBi [45], prior to RetNet, also applied a similar approach

*Ran He is the corresponding author.

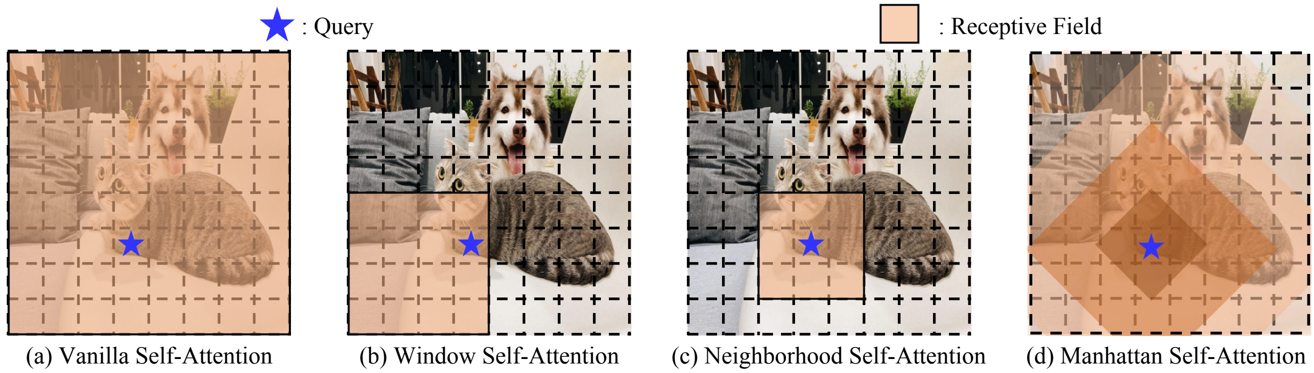


Figure 2. Comparison among different Self-Attention mechanisms. In MaSA, darker colors represent smaller spatial decay rates, while lighter colors represent larger ones. The spatial decay rates that change with distance provide the model with rich spatial priors.

and succeeded in NLP tasks. We extend this temporal decay matrix to the spatial domain, developing a two-dimensional bidirectional spatial decay matrix based on the Manhattan distance among tokens. In our space decay matrix, for a target token, the farther the surrounding tokens are, the greater the degree of decay in their attention scores. This property allows the target token to perceive global information while simultaneously assigning different levels of attention to tokens at varying distances. We introduce explicit spatial prior to the vision backbone using this spatial decay matrix. We name this Self-Attention mechanism, which is inspired by RetNet and incorporates the Manhattan distance as the explicit spatial prior, as **Ma**nhattan **S**elf-**A**ttention (MaSA).

Besides explicit spatial priors, another issue caused by global modeling with Self-Attention is the enormous computational burden. Previous sparse attention mechanisms [12, 39, 56, 66, 78] and the way retention is decomposed in RetNet [50] mostly disrupt the spatial decay matrix, making them unsuitable for MaSA. In order to sparsely model global information without compromising the spatial decay matrix, we propose a method to decompose Self-Attention along both axes of the image. This decomposition method decomposes Self-Attention and the spatial decay matrix without any loss of prior information. The decomposed MaSA models global information with linear complexity and has the same receptive field shape as the original MaSA. We compare MaSA with other Self-Attention mechanisms in Fig. 2. It can be seen that our MaSA introduces richer spatial priors to the model than its counterparts.

Based on MaSA, we construct a powerful vision backbone called RMT. We demonstrate the effectiveness of the proposed method through extensive experiments. As shown in Fig. 1, our RMT outperforms the state-of-the-art (SOTA) models on image classification tasks. Additionally, our model exhibits more prominent advantages compared to other models in tasks such as object detection, instance segmentation, and semantic segmentation. Our contributions can be summarized as follows:

- We propose a spatial decay matrix based on Manhattan distance to augment Self-Attention, creating the Manhattan Self-Attention (MaSA) with an explicit spatial prior.
- We propose a decomposition form for MaSA, enabling linear complexity for global information modeling without disrupting the spatial decay matrix.
- Leveraging MaSA, we construct RMT, a powerful vision backbone for general purposes. RMT attains high top-1 accuracy on ImageNet-1k in image classification without extra training data, and excels in tasks like object detection, instance segmentation, and semantic segmentation.

2. Related Work

Transformer. Transformer architecture was firstly proposed in [55] to address the training limitation of recurrent model and then achieve massive success in many NLP tasks. By splitting the image into small, non-overlapped patches sequence, Vision Transformer (ViTs) [13] also have attracted great attention and become widely used on vision tasks [6, 16, 20, 44, 61, 68]. Unlike in the past, where RNNs and CNNs have respectively dominated the NLP and CV fields, the transformer architecture has shined through in various modalities and fields [29, 41, 46, 63]. In the computer vision community, many studies are attempting to introduce spatial priors into ViT to reduce the data requirements for training [7, 21, 52]. At the same time, various sparse attention mechanisms have been proposed to reduce the computational cost of Self-Attention [14, 56, 57, 60].

Prior Knowledge in Transformer. Numerous attempts have been made to incorporate prior knowledge into the Transformer model to enhance its performance. The original Transformers [13, 55] use trigonometric position encoding to provide positional information for each token. In vision tasks, [39] proposes the use of relative positional encoding as a replacement for the original absolute positional encoding. [7] points out that zero padding in convolutional layers could also provide positional awareness for the ViT, and this position encoding method is highly efficient. In many studies, Convolution in FFN [14, 18, 57] has been employed for vision models to further enrich the positional information in the ViT. For NLP tasks, in the recent Retentive Network [50], the temporal decay matrix has been introduced to provide the model with prior knowledge based on distance changes. Before RetNet, ALiBi [45] also uses a similar temporal decay matrix.

3. Methodology

3.1. Preliminary

Temporal decay in RetNet. Retentive Network (RetNet) is a powerful architecture for language models. This work proposes the retention mechanism for sequence modeling. Retention brings the temporal decay to the language model, which Transformers do not have. Retention firstly considers a sequence modeling problem in a recurrent manner. It can be written as Eq. 1:

$$o_n = \sum_{m=1}^{n} \gamma^{n-m}(Q_n e^{in\theta})(K_m e^{im\theta})^{\dagger} v_m \tag{1}$$

For a parallel training process, Eq. 1 is expressed as:

$$\begin{gathered} Q = (XW_Q) \odot \Theta, \quad K = (XW_K) \odot \overline{\Theta}, \quad V = XW_V \\ \Theta_n = e^{in\theta}, \quad D_{nm} = \begin{cases} \gamma^{n-m}, & n \geq m \\ 0, & n < m \end{cases} \\ \mathrm{Retention}(X) = (QK^{\intercal} \odot D)V \end{gathered} \tag{2}$$

where $\overline{\Theta}$ is the complex conjugate of Θ, and $D \in \mathbb{R}^{|x| \times |x|}$ contains both causal masking and exponential decay, which symbolizes the relative distance in one-dimensional sequence and brings the explicit temporal prior to text data.

3.2. Manhattan Self-Attention

Starting from the retention in RetNet, we evolve it into Manhattan Self-Attention (MaSA). Within MaSA, we transform the unidirectional and one-dimensional temporal decay observed in retention into bidirectional and two-dimensional spatial decay. This spatial decay introduces an explicit spatial prior linked to Manhattan distance into the vision backbone. Additionally, we devise a straightforward approach to concurrently decompose the Self-Attention and spatial decay matrix along the two axes of the image.

From Unidirectional to Bidirectional Decay: In RetNet, retention is unidirectional due to the causal nature of text data, allowing each token to attend only to preceding tokens and not those following it. This characteristic is ill-suited for tasks lacking causal properties, such as image recognition. Hence, we initially broaden the retention to a bidirectional form, expressed as Eq. 3:

$$\begin{gathered} \mathrm{BiRetention}(X) = (QK^{\intercal} \odot D^{Bi})V \\ D_{nm}^{Bi} = \gamma^{|n-m|} \end{gathered} \tag{3}$$

where $\mathrm{BiRetention}$ signifies bidirectional modeling.

From One-dimensional to Two-dimensional Decay: While retention now supports bi-directional modeling, this capability remains confined to a one-dimensional level and is inadequate for two-dimensional images. To address this limitation, we extend the one-dimensional retention to encompass two dimensions.

In the context of images, each token is uniquely positioned with a two-dimensional coordinate within the plane, denoted as (x_n, y_n) for the n-th token. To adapt to this, we adjust each element in the matrix D to represent the Manhattan distance between the respective token pairs based on their 2D coordinates. The matrix D is redefined as follows:

$$D_{nm}^{2d} = \gamma^{|x_n - x_m| + |y_n - y_m|} \tag{4}$$

In the retention, the $\mathrm{Softmax}$ is abandoned and replaced with a gating function. This variation gives RetNet multiple flexible computation forms, enabling it to adapt to parallel training and recurrent inference processes. Despite this flexibility, when exclusively utilizing RetNet's parallel computation form in our experiments, the necessity of retaining the gating function becomes debatable. Our findings indicate that this modification does not improve results for vision models; instead, it introduces extra parameters and computational complexity. Consequently, we continue to employ $\mathrm{Softmax}$ to introduce nonlinearity to our model. Combining the aforementioned steps, our Manhattan Self-Attention is expressed as

$$\begin{gathered} \mathrm{MaSA}(X) = (\mathrm{Softmax}(QK^{\intercal}) \odot D^{2d})V \\ D_{nm}^{2d} = \gamma^{|x_n - x_m| + |y_n - y_m|} \end{gathered} \tag{5}$$

Decomposed Manhattan Self-Attention. In the early stages of the vision backbone, an abundance of tokens leads to high computational costs for Self-Attention when attempting to model global information. Our MaSA encounters this challenge as well. Utilizing existing sparse attention mechanisms [12, 21, 39, 56, 66], or the original RetNet's recurrent/chunk-wise recurrent form directly, disrupts the spatial decay matrix based on Manhattan distance, resulting in the loss of explicit spatial prior. To address this, we introduce a simple decomposition method that not only decomposes Self-Attention but also decomposes the spatial decay matrix. The decomposed MaSA is represented in

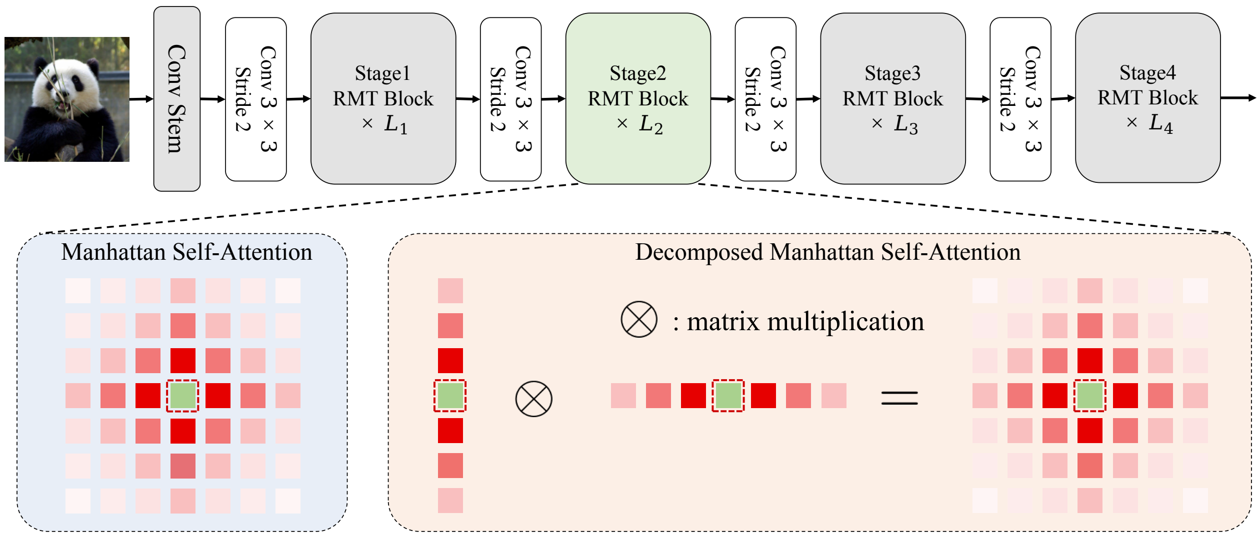


Figure 3. Overall architecture of RMT.

Eq. 6. Specifically, we calculate attention scores separately for the horizontal and vertical directions in the image. Subsequently, we apply the one-dimensional bidirectional decay matrix to these attention weights. The one-dimensional decay matrix signifies the horizontal and vertical distances between tokens ($D^H_{nm} = \gamma^{|y_n - y_m|}$, $D^W_{nm} = \gamma^{|x_n - x_m|}$):

$$\begin{aligned} Attn_H &= \mathrm{Softmax}(Q_H K_H^\intercal) \odot D^H, \\ Attn_W &= \mathrm{Softmax}(Q_W K_W^\intercal) \odot D^W, \\ \mathrm{MaSA}(X) &= Attn_H (Attn_W V)^\intercal \end{aligned} \tag{6}$$

Based on the decomposition of MaSA, the shape of the receptive field of each token is shown in Fig. 3, which is identical to the shape of the complete MaSA's receptive field. Fig. 3 indicates that our decomposition method fully preserves the explicit spatial prior.

To further enhance the local expression capability of MaSA, following [78], we introduce a Local Context Enhancement module using DWConv:

$$X_{out} = \mathrm{MaSA}(X) + \mathrm{LCE}(V); \tag{7}$$

3.3. Overall Architecture

We construct the RMT based on MaSA, and its architecture is illustrated in Fig. 3. Similar to previous general vision backbones [39, 56, 57, 73], RMT is divided into four stages. The first three stages utilize the decomposed MaSA, while the last uses the original MaSA. Like many previous backbones [18, 33, 74, 78], we incorporate CPE [7] into our model.

4. Experiments

We conducted extensive experiments on multiple vision tasks, such as image classification on ImageNet-1K [10], object detection and instance segmentation on COCO 2017 [37], and semantic segmentation on ADE20K [76]. We also make ablation studies to validate the importance of each component in RMT. More details can be found in Appendix.

4.1. Image Classification

Settings. We train our models on ImageNet-1K [10] from scratch. We follow the same training strategy in [52], with the only supervision being classification loss for a fair comparison. The maximum rates of increasing stochastic depth [26] are set to 0.1/0.15/0.4/0.5 for RMT-T/S/B/L [26], respectively. We use the AdamW optimizer with a cosine decay learning rate scheduler to train the models. We set the initial learning rate, weight decay, and batch size to 0.001, 0.05, and 1024, respectively. We adopt the strong data augmentation and regularization used in [39]. Our settings are RandAugment [9] (randm9-mstd0.5-inc1), Mixup [72] (prob=0.8), CutMix [71] (prob=1.0), Random Erasing [75] (prob=0.25). In addition to the conventional training methods, similar to LV-ViT [30] and VOLO [70], we train a model that utilizes token labeling to provide supplementary supervision.

Results. We compare RMT against many state-of-the-art models in Tab. 1. Results in the table demonstrate that RMT consistently outperforms previous models across all settings. Specifically, RMT-S achieves **84.1%** Top1-accuracy with only **4.5** GFLOPs. RMT-B also surpasses iFormer [49] by **0.4%** with similar FLOPs. Furthermore, our RMT-L model surpasses MaxViT-B [54] in top1-accuracy by **0.6%** while using fewer FLOPs. Our RMT-T has also outperformed many lightweight models. As for the model trained

Cost	Model	Parmas (M)	FLOPs (G)	Top1-acc (%)
tiny model $\sim$ 2.5G	PVTv2-b1 [57]	13	2.1	78.7
	QuadTree-B-b1 [51]	14	2.3	80.0
	RegionViT-T [4]	14	2.4	80.4
	MPViT-XS [32]	11	2.9	80.9
	VAN-B1 [19]	14	2.5	81.1
	BiFormer-T [78]	13	2.2	81.4
	Conv2Former-N [25]	15	2.2	81.5
	CrossFormer-T [58]	28	2.9	81.5
	NAT-M [21]	20	2.7	81.8
	FAT-B2 [15]	14	2.0	81.9
	QnA-T [1]	16	2.5	82.0
	GC-ViT-XT [22]	20	2.6	82.0
	SMT-T [38]	12	2.4	82.2
	RMT-T	14	2.5	82.4
small model $\sim$ 4.5G	Swin-T [39]	29	4.5	81.3
	CrossViT-15 [3]	27	5.8	81.5
	RVT-S [42]	23	4.7	81.9
	ConvNeXt-T [40]	29	4.5	82.1
	Focal-T [66]	29	4.9	82.2
	CSWin-T [12]	23	4.3	82.7
	SG-Former-S [17]	23	4.8	83.2
	Ortho-S [27]	24	4.5	83.4
	InternImage-T [59]	30	5.0	83.5
	GC-ViT-T [22]	28	4.7	83.5
	CMT-S [18]	25	4.0	83.5
	STViT-S [28]	25	4.4	83.6
	FAT-B3 [15]	29	4.4	83.6
	SMT-S [38]	20	4.8	83.7
	BiFormer-S [78]	26	4.5	83.8
	RMT-S	27	4.5	84.1
	UniFormer-S* [33]	24	4.2	83.4
	WaveViT-S* [68]	23	4.7	83.9
	Dual-ViT-S* [69]	25	5.4	84.1
	VOLO-D1* [70]	27	6.8	84.2
	BiFormer-S* [78]	26	4.5	84.3
	RMT-S*	27	4.5	84.8

Cost	Model	Parmas (M)	FLOPs (G)	Top1-acc (%)
base model $\sim$ 9.0G	Swin-S [39]	50	8.7	83.0
	ConvNeXt-S [40]	50	8.7	83.1
	CrossFormer-B [58]	52	9.2	83.4
	NAT-S [21]	51	7.8	83.7
	Quadtree-B-b4 [51]	64	11.5	84.0
	Ortho-B [27]	50	8.6	84.0
	ScaleViT-B [67]	81	8.6	84.1
	MOAT-1 [65]	42	9.1	84.2
	DaViT-S [11]	50	8.8	84.2
	MViTv2-B [34]	52	10.2	84.4
	iFormer-B [49]	48	9.4	84.6
	SE-CoTNetD-152 [35]	56	26.5	84.6
	STViT-B [28]	52	9.9	84.8
	FAT-B4 [15]	52	9.3	84.8
	RMT-B	54	9.7	85.0
	WaveViT-B* [68]	34	7.2	84.8
	UniFormer-B* [33]	50	8.3	85.1
	Dual-ViT-B* [69]	43	9.3	85.2
	BiFormer-B* [78]	58	9.8	85.4
	RMT-B*	55	9.7	85.6
large model $\sim$ 18.0G	Swin-B [39]	88	15.4	83.3
	LITv2 [44]	87	13.2	83.6
	CrossFormer-L [58]	92	16.1	84.0
	Ortho-L [27]	88	15.4	84.2
	SMT-L [38]	81	17.7	84.6
	DaViT-B [11]	88	15.5	84.6
	iFormer-L [49]	87	14.0	84.8
	InterImage-B [59]	97	16.0	84.9
	MaxViT-B [54]	120	23.4	84.9
	FAT-B5 [15]	88	15.1	85.2
	STViT-L [28]	95	15.6	85.3
	RMT-L	95	18.2	85.5
	VOLO-D3* [70]	86	20.6	85.4
	WaveViT-L* [68]	58	14.8	85.5
	UniFormer-L* [33]	100	12.6	85.6
	RMT-L*	96	18.2	86.1

Table 1. Comparison with the state-of-the-art on ImageNet-1K classification. "*" indicates the model trained with token labeling [30].

using token labeling, our RMT-S outperforms the current state-of-the-art BiFormer-S by **0.5%**.

4.2. Object Detection and Instance Segmentation

Settings. We adopt MMDetection [5] to implement RetinaNet [36], Mask-RCNN [24] and Cascade Mask R-CNN [2]. We use the commonly used "1×" (12 training epochs) setting for the RetinaNet and Mask R-CNN. Besides, we use "3 × +MS" for Mask R-CNN and Cascade Mask R-CNN. Following [39], during training, images are resized to the shorter side of 800 pixels while the longer side is within 1333 pixels. We adopt the AdamW optimizer with a learning rate of 0.0001 and batch size of 16 to optimize the model. For the "1×" schedule, the learning rate declines with the decay rate of 0.1 at the epoch 8 and 11. While for the "3 × +MS" schedule, the learning rate declines with the decay rate of 0.1 at the epoch 27 and 33.

Results. Tab. 2, Tab. 3 and Tab. 4 show the results with different detection frameworks. The results demonstrate that our RMT performs best in all comparisons. For the RetinaNet framework, our RMT-T outperforms MPViT-XS by **+1.3** AP, while S/B/L also perform better than other methods. As for the Mask R-CNN with "1×" schedule, RMT-L outperforms the recent InternImage-B by **+2.8** box AP and **+1.9** mask AP. For "3 × +MS" schedule, RMT-S outperforms InternImage-T for **+1.6** box AP and **+1.2** mask AP. Besides, regarding the Cascade Mask R-CNN, our RMT still performs much better than other backbones. All the above results tell that RMT outperforms its counterparts by evident margins.

4.3. Semantic Segmentation

Settings. We adopt the Semantic FPN [31] and UperNet [62] based on MMSegmentation [8], apply RMTs

Backbone	Params	FLOPs	Mask R-CNN 1×						Params	FLOPs	RetinaNet 1×					
	(M)	(G)	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}	(M)	(G)	AP^b	AP^b_{50}	AP^b_{75}	AP^b_S	AP^b_M	AP^b_L
PVT-T [56]	33	240	39.8	62.2	43.0	37.4	59.3	39.9	23	221	39.4	59.8	42.0	25.5	42.0	52.1
PVTv2-B1 [57]	33	243	41.8	54.3	45.9	38.8	61.2	41.6	23	225	41.2	61.9	43.9	25.4	44.5	54.3
MPViT-XS [32]	30	231	44.2	66.7	48.4	40.4	63.4	43.4	20	211	43.8	65.0	47.1	28.1	47.6	56.5
RMT-T	33	218	**47.1**	**68.8**	**51.7**	**42.6**	**65.8**	**45.9**	23	199	**45.1**	**66.2**	**48.1**	**28.8**	**48.9**	**61.1**
Swin-T [39]	48	267	43.7	66.6	47.7	39.8	63.3	42.7	38	248	41.7	63.1	44.3	27.0	45.3	54.7
CMT-S [18]	45	249	44.6	66.8	48.9	40.7	63.9	43.4	44	231	44.3	65.5	47.5	27.1	48.3	59.1
ScalableViT-S [67]	46	256	45.8	67.6	50.0	41.7	64.7	44.8	36	238	45.2	66.5	48.4	29.2	49.1	60.3
MPViT-S [32]	43	268	46.4	68.6	51.2	42.4	65.6	45.7	32	248	45.7	57.3	48.8	28.7	49.7	59.2
InternImage-T [59]	49	270	47.2	69.0	52.1	42.5	66.1	45.8	–	–	–	–	–	–	–	–
SMT-S [38]	40	265	47.8	69.5	52.1	43.0	66.6	46.1	–	–	–	–	–	–	–	–
RMT-S	46	262	**49.0**	**70.8**	**53.9**	**43.9**	**67.8**	**47.4**	36	244	**47.8**	**69.1**	**51.8**	**32.1**	**51.8**	**63.5**
ResNet-101 [23]	63	336	40.4	61.1	44.2	36.4	57.7	38.8	58	315	38.5	57.8	41.2	21.4	42.6	51.1
Swin-S [39]	69	359	45.7	67.9	50.4	41.1	64.9	44.2	60	339	44.5	66.1	47.4	29.8	48.5	59.1
ScalableViT-B [67]	95	349	46.8	68.7	51.5	42.5	65.8	45.9	85	330	45.8	67.3	49.2	29.9	49.5	61.0
InternImage-S [59]	69	340	47.8	69.8	52.8	43.3	67.1	46.7	–	–	–	–	–	–	–	–
CSWin-S [12]	54	342	47.9	70.1	52.6	43.2	67.1	46.2	–	–	–	–	–	–	–	–
STViT-B [28]	70	359	49.7	71.7	54.7	44.8	68.9	48.7	–	–	–	–	–	–	–	–
RMT-B	73	373	**51.1**	**72.5**	**56.1**	**45.5**	**69.7**	**49.3**	63	355	**49.1**	**70.3**	**53.0**	**32.9**	**53.2**	**64.2**
Swin-B [39]	107	496	46.9	69.2	51.6	42.3	66.0	45.5	98	477	45.0	66.4	48.3	28.4	49.1	60.6
Focal-B [66]	110	533	47.8	70.2	52.5	43.2	67.3	46.5	101	514	46.3	68.0	49.8	31.7	50.4	60.8
MPViT-B [32]	95	503	48.2	70.0	52.9	43.5	67.1	46.8	85	482	47.0	68.4	50.8	29.4	51.3	61.5
CSwin-B [12]	97	526	48.7	70.4	53.9	43.9	67.8	47.3	–	–	–	–	–	–	–	–
InternImage-B [59]	115	501	48.8	70.9	54.0	44.0	67.8	47.4	–	–	–	–	–	–	–	–
RMT-L	114	557	**51.6**	**73.1**	**56.5**	**45.9**	**70.3**	**49.8**	104	537	**49.4**	**70.6**	**53.1**	**34.2**	**53.9**	**65.2**

Table 2. Comparison to other backbones using RetinaNet and Mask R-CNN on COCO val2017 object detection and instance segmentation.

Backbone	Params	FLOPs	Mask R-CNN 3×+MS					
	(M)	(G)	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
ConvNeXt-T [40]	48	262	46.2	67.9	50.8	41.7	65.0	45.0
Focal-T [66]	49	291	47.2	69.4	51.9	42.7	66.5	45.9
NAT-T [21]	48	258	47.8	69.0	52.6	42.6	66.0	45.9
GC-ViT-T [22]	48	291	47.9	70.1	52.8	43.2	67.0	46.7
MPViT-S [32]	43	268	48.4	70.5	52.6	43.9	67.6	47.5
SMT-S [38]	40	265	49.0	70.1	53.4	43.4	67.3	46.7
CSWin-T [12]	42	279	49.0	70.7	53.7	43.6	67.9	46.6
RMT-S	46	262	**50.7**	**71.9**	**55.6**	**44.9**	**69.1**	**48.4**
ConvNeXt-S [40]	70	348	47.9	70.0	52.7	42.9	66.9	46.2
NAT-S [21]	70	330	48.4	69.8	53.2	43.2	66.9	46.4
Swin-S [39]	69	359	48.5	70.2	53.5	43.3	67.3	46.6
InternImage-S [59]	69	340	49.7	71.1	54.5	44.5	68.5	47.8
SMT-B [38]	52	328	49.8	71.0	54.4	44.0	68.0	47.3
RMT-B	73	373	**52.2**	**72.9**	**57.0**	**46.1**	**70.4**	**49.9**

Table 3. Comparison to other backbones using Mask R-CNN with "$3\times$+MS" schedule.

Backbone	Params	FLOPs	Cascade Mask R-CNN 3×+MS					
	(M)	(G)	AP^b	AP^b_{50}	AP^b_{75}	AP^m	AP^m_{50}	AP^m_{75}
NAT-T [21]	85	737	51.4	70.0	55.9	44.5	67.6	47.9
GC-ViT-T [22]	85	770	51.6	70.4	56.1	44.6	67.8	48.3
SMT-S [38]	78	744	51.9	70.5	56.3	44.7	67.8	48.6
UniFormer-S [33]	79	747	52.1	71.1	56.6	45.2	68.3	48.9
Ortho-S [27]	81	755	52.3	71.3	56.8	45.3	68.6	49.2
HorNet-T [47]	80	728	52.4	71.6	56.8	45.6	69.1	49.6
CSWin-T [12]	80	757	52.5	71.5	57.1	45.3	68.8	48.9
RMT-S	83	741	**53.2**	**72.0**	**57.8**	**46.1**	**69.8**	**49.8**
Swin-S [39]	107	838	51.9	70.7	56.3	45.0	68.2	48.8
NAT-S [21]	108	809	51.9	70.4	56.2	44.9	68.2	48.6
GC-ViT-S [22]	108	866	52.4	71.0	57.1	45.4	68.5	49.3
DAT-S [61]	107	857	52.7	71.7	57.2	45.5	69.1	49.3
CSWin-S [12]	92	820	53.7	72.2	58.4	46.4	69.6	50.6
RMT-B	111	852	**54.5**	**72.8**	**59.0**	**47.2**	**70.5**	**51.4**

Table 4. Comparison to other backbones using Cascade Mask R-CNN with "$3\times$+MS" schedule.

which are pretrained on ImageNet-1K as backbone. We use the same setting of PVT [56] to train the Semantic FPN, and we train the model for 80k iterations. All models are trained with the input resolution of 512×512. When testing the model, we resize the shorter side of the image to 512 pixels. As for UperNet, we follow the default settings in Swin [39]. We take AdamW with a weight decay of 0.01 as the optimizer to train the models for 160K iterations. The learning rate is set to 6×10^{-5} with 1500 iterations warmup.

Results. The results of semantic segmentation can be found in Tab. 5. All the FLOPs are measured with the resolution of 512×2048, except the group of RMT-T, which are measured with the resolution of 512×512. All our models achieve the best performance in all comparisons. Specifically, our RMT-S exceeds Shunted-S for **+1.2** mIoU with Semantic FPN. Moreover, our RMT-B outperforms the recent InternImage-S for **+1.8** mIoU. All the above results demonstrate our model's superiority in dense prediction.

Backbone	Method	Params(M)	FLOPs(G)	mIoU(%)
PVTv2-B1 [57]	FPN	17.8	34.2	42.5
VAN-B1 [19]	FPN	18.1	34.9	42.9
FAT-B2 [15]	FPN	17.2	32.2	45.4
EdgeViT-S [43]	FPN	16.9	32.1	45.9
RMT-T	FPN	17.0	33.7	**46.4**
DAT-T [61]	FPN	32	198	42.6
CrossFormer-S [58]	FPN	34	221	46.0
Shuted-S [48]	FPN	26	183	48.2
FAT-B3 [15]	FPN	33	179	48.9
RMT-S	FPN	30	180	**49.4**
DAT-S [61]	FPN	53	320	46.1
RegionViT-B+ [4]	FPN	77	459	47.5
UniFormer-B [33]	FPN	54	350	47.7
CrossFormer-B [58]	FPN	56	331	47.7
RMT-B	FPN	57	294	**50.4**
DAT-B [61]	FPN	92	481	47.0
CrossFormer-L [58]	FPN	95	497	48.7
CSWin-B [12]	FPN	81	464	49.9
RMT-L	FPN	98	482	**51.4**
NAT-T [21]	UperNet	58	934	47.1
InternImage-T [59]	UperNet	59	944	47.9
MPViT-S [32]	UperNet	52	943	48.3
SMT-S [38]	UperNet	50	935	49.2
RMT-S	UperNet	56	937	**49.8**
DAT-S [61]	UperNet	81	1079	48.3
InterImage-S [59]	UperNet	80	1017	50.2
MPViT-B [32]	UperNet	105	1186	50.3
CSWin-S [12]	UperNet	65	1027	50.4
RMT-B	UperNet	83	1051	**52.0**
Swin-B [39]	UperNet	121	1188	48.1
InternImage-B [59]	UperNet	128	1185	50.8
CSWin-B [12]	UperNet	109	1222	51.1
RMT-L	UperNet	125	1241	**52.8**

Table 5. Comparison with the state-of-the-art on ADE20K.

Model	Params(M)	FLOPs(G)	IN(%)	IN-A(%)	IN-R(%)
Swin-T [39]	29	4.5	81.3	21.6	41.3
FAN-S-Hybrid [77]	26	6.7	83.6	33.9	50.7
RMT-S	27	4.5	**84.1**	**38.9**	**51.1**
FAN-B-Hybrid [77]	50	11.3	83.9	39.6	52.9
RMT-B	54	9.7	**85.0**	**47.6**	**55.0**
Swin-B [39]	88	15.4	83.4	35.8	46.6
FAN-L-Hybrid [77]	77	16.9	84.3	41.8	53.2
RMT-L	95	18.2	**85.5**	**51.3**	**56.7**

Table 6. Robustness evaluation on OOD data.

4.4. Ablation Study

Robustness of RMT. We evaluate RMT (trained on ImageNet-1k) on ImageNet-A(IN-A) and ImageNet-R(IN-R). RMT performs better than Swin and robustness-oriented FAN.(In Tab. 6).

Results on ImageNet-V2. RMT's performance on ImageNet-V2 is shown in Tab. 7. "IN-V2" denotes ImageNet-V2. Our RMT exhibit very impressive performance on ImageNet-V2.

Model	Params(M)	FLOPs(G)	Top1(%)	Top5(%)	IN-V2(%)
tiny-MOAT-2 [65]	10	2.3	81.0	–	70.1
RMT-T	14	2.5	**82.4**	**96.2**	**72.1**
MOAT-0 [65]	28	5.7	83.3	–	72.8
BiFormer-S [78]	26	4.5	83.8	96.6	73.6
RMT-S	27	4.5	**84.1**	**96.8**	**74.1**
VOLO-D1* [70]	27	6.8	84.2	96.8	74.0
RMT-S*	27	4.5	**84.8**	**97.1**	**74.8**
MOAT-1 [65]	42	9.1	84.2	–	74.2
BiFormer-B [78]	57	9.8	84.3	96.9	74.0
RMT-B	54	9.7	**85.0**	**97.3**	**75.6**
VOLO-D2* [70]	59	14.1	85.2	97.2	75.2
RMT-B*	55	9.7	**85.6**	**97.3**	**76.1**
MOAT-2 [65]	73	17.2	84.7	–	74.3
RMT-L	95	18.2	**85.5**	**97.4**	**76.3**
VOLO-D3* [70]	86	20.6	85.4	97.3	75.6
RMT-L*	96	18.2	**86.1**	**97.6**	**76.6**

Table 7. Classification Results. "*" indicates using token label.

Strict comparison with previous works. In order to make a strict comparison with previous methods, we align RMT's hyperparameters (such as whether to use hierarchical structure, the number of channels in the four stages of the hierarchical model, whether to use positional encoding and convolution stem, etc.) of the overall architecture with DeiT [52] and Swin [39], and only replace the Self-Attention/Window Self-Attention with our MaSA. The comparison results are shown in Tab. 8, where RMT significantly outperforms DeiT-S, Swin-T, and Swin-S.

MaSA. We verify the impact of Manhattan Self-Attention on the model, as shown in the Tab. 8. MaSA improves the model's performance in image classification and downstream tasks by a large margin. Specifically, the classification accuracy of MaSA is **0.8%** higher than that of vanilla attention.

LCE. Local Context Enhancement also plays a role in the excellent performance of our model. LCE improves the classification accuracy of RMT by 0.3% and enhances the model's performance in downstream tasks.

CPE. Just like previous methods, CPE provides our model with flexible position encoding and more positional information, contributing to the improvement in the model's performance in image classification and downstream tasks.

Convolutional Stem. The initial convolutional stem of the model provides better local information, thereby further enhancing the model's performance on various tasks.

Decomposed MaSA. In RMT-S, we substitute the decomposed MaSA (MaSA-d) in the third stage with the original MaSA to validate the effectiveness of our decomposition method, as illustrated in Tab. 9. In terms of image classification, MaSA-d and MaSA achieve comparable accuracy. However, for semantic segmentation, employing MaSA-d

Model	Params(M)	FLOPs(G)	Top1-acc(%)	AP^b	AP^m	mIoU(%)
DeiT-S [52]	22	4.6	79.8	–	–	–
RMT-DeiT-S	22	4.6	81.7(+1.9)	–	–	–
Swin-T [39]	29	4.5	81.3	43.7	39.8	44.5
RMT-Swin-T	29	4.7	83.6(+2.3)	47.8(+4.1)	43.1(+3.3)	49.1(+4.6)
Swin-S [39]	50	8.8	83.0	45.7	41.1	47.6
RMT-Swin-S	50	9.1	84.5(+1.5)	49.5(+3.8)	44.2(+3.1)	51.0 (+3.4)
RMT-T	14.3	2.5	82.4	47.1	42.6	46.4
MaSA→Attention	14.3	2.5	81.6(-0.8)	44.6(-2.5)	40.7(-1.9)	43.9(-2.5)
w/o LCE	14.2	2.4	82.1	46.7	42.3	46.0
w/o CPE	14.3	2.5	82.2	47.0	42.4	46.4
w/o Stem	14.3	2.2	82.2	46.8	42.3	46.2

Table 8. Ablation study. We make a strict comparison among RMT, DeiT, and Swin-Transformer.

3rd stage	FLOPs(G)	Top1(%)	FLOPs(G)	mIoU(%)
MaSA-d	4.5	84.1	180	49.4
MaSA	4.8	84.1	246	49.7

Table 9. Comparison between decomposed MaSA (MaSA-d) and original MaSA.

Model	Params(M)	FLOPs(G)	acc(%)	AP^b	AP^m
Swin-T	29	4.5	81.3	43.7	39.8
WSA→decomposed SA	29	4.7	82.4	45.5	41.4
$+\gamma = 0.998$ (for all heads)	29	4.7	82.4	45.7	41.7
$+\gamma = 0.35$ (for all heads)	29	4.7	82.6	46.1	41.9
$+\gamma = 0.75$ (for all heads)	29	4.7	82.9	46.5	42.2
$+\gamma \in$[0.93,0.998] (vary across heads)	29	4.7	83.2	46.7	42.3
$+\gamma \in$[0.5,0.992] (vary across heads)	29	4.7	83.3	47.4	42.8
$+\gamma \in$[0.75,0.996] (vary across heads)	29	4.7	83.6	47.8	43.1

Table 10. Ablation of MaSA.

Model	Params (M)	FLOPs↓ (G)	Throughput↑ (imgs/s)	Top1 (%)
BiFormer-T [78]	13	2.2	1602	81.4
SMT-T [38]	12	2.4	636	82.2
RMT-T	14	2.5	1650	82.4
CMT-S [18]	25	4.0	848	83.5
SMT-S [38]	20	4.8	356	83.7
BiFormer-S [78]	26	4.5	766	83.8
RMT-Swin-T	29	4.7	1192	83.6
RMT-S	27	4.5	876	84.1
SMT-B [38]	32	7.7	237	84.3
BiFormer-B [78]	57	9.8	498	84.3
CMT-B [18]	46	9.3	447	84.5
RMT-Swin-S	50	9.1	722	84.5
RMT-B	54	9.7	457	85.0
SMT-L [38]	80	17.7	158	84.6
RMT-L	95	18.2	326	85.5

Table 11. Comparison of inference speed among SOTA models.

significantly reduces computational burden while yielding similar result.

Why MaSA works? As shown in Tab. 10, gradually transforming Swin-T to RMT-Swin-T revealed key insights: 1)Decomposed form. Substituting Swin's WSA with decomposed Self-Attention improves performance, likely due to the global perception of the decomposed form. 2)Decay matrix introduces local bias. UniFormer highlights the importance of local bias in early model stages. Introducing local bias through an appropriate decay rate is beneficial. 3)Multiple decay rates. Assigning distinct decay rates to each MaSA head introduces multi-scale local bias, further enhancing performance.

Inference Speed. We compare the RMT's inference speed with the recent best performing vision backbones in Tab. 11. Our RMT demonstrates the optimal trade-off between speed and accuracy.

5. Conclusion

In this work, we propose RMT, a vision backbone with explicit spatial prior. RMT extends the temporal decay used for causal modeling in NLP to the spatial level and introduces a spatial decay matrix based on the Manhattan distance. The matrix incorporates explicit spatial prior into the Self-Attention. Additionally, RMT utilizes a Self-Attention decomposition form that can sparsely model global information without disrupting the spatial decay matrix. The combination of spatial decay matrix and attention decomposition form enables RMT to possess explicit spatial prior and linear complexity. Extensive experiments in image classification, object detection, instance segmentation, and semantic segmentation validate the superiority of RMT.

6. Acknowledgements

This work is partially funded by National Natural Science Foundation of China (Grant No. U21B2045), Beijing Nova Program (20230484276), Youth Innovation Promotion Association CAS (Grant No. 2022132), and CAAI-Huawei MindSpore Open Fund.

References

[1] Moab Arar, Ariel Shamir, and Amit H. Bermano. Learned queries for efficient local attention. In *CVPR*, 2022. 5

[2] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *CVPR*, 2018. 5

[3] Chun-Fu (Richard) Chen, Quanfu Fan, and Rameswar Panda. CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification. In *ICCV*, 2021. 5

[4] Chun-Fu (Richard) Chen, Rameswar Panda, and Quanfu Fan. RegionViT: Regional-to-Local Attention for Vision Transformers. In *ICLR*, 2022. 5, 7

[5] Kai Chen, Jiaqi Wang, Jiangmiao Pang, et al. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 5

[6] 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. In *NeurIPS*, 2021. 2

[7] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, and Chunhua Shen. Conditional positional encodings for vision transformers. In *ICLR*, 2023. 2, 4

[8] MMSegmentation Contributors. Mmsegmentation, an open source semantic segmentation toolbox, 2020. 5

[9] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, et al. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPRW*, 2020. 4

[10] Jia Deng, Wei Dong, Richard Socher, et al. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 4

[11] Mingyu Ding, Bin Xiao, Noel Codella, et al. Davit: Dual attention vision transformers. In *ECCV*, 2022. 5

[12] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, et al. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *CVPR*, 2022. 2, 3, 5, 6, 7

[13] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 2

[14] Qihang Fan, Huaibo Huang, Jiyang Guan, and Ran He. Rethinking local perception in lightweight vision transformer, 2023. 1, 2, 3

[15] Qihang Fan, Huaibo Huang, Xiaoqiang Zhou, and Ran He. Lightweight vision transformer with bidirectional interaction. In *NeurIPS*, 2023. 5, 7

[16] Li Gao, Dong Nie, Bo Li, and Xiaofeng Ren. Doubly-fused vit: Fuse information from vision transformer doubly with local representation. In *ECCV*, 2022. 2

[17] SG-Former: Self guided Transformer with Evolving Token Reallocation. Sucheng ren, xingyi yang, songhua liu, xinchao wang. In *ICCV*, 2023. 5

[18] Jianyuan Guo, Kai Han, Han Wu, Chang Xu, Yehui Tang, Chunjing Xu, and Yunhe Wang. Cmt: Convolutional neural networks meet vision transformers. In *CVPR*, 2022. 1, 3, 4, 5, 6, 8

[19] Meng-Hao Guo, Cheng-Ze Lu, Zheng-Ning Liu, Ming-Ming Cheng, and Shi-Min Hu. Visual attention network. *arXiv preprint arXiv:2202.09741*, 2022. 5, 7

[20] Kai Han, An Xiao, Enhua Wu, et al. Transformer in transformer. In *NeurIPS*, 2021. 2

[21] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *CVPR*, 2023. 1, 2, 3, 5, 6, 7

[22] Ali Hatamizadeh, Hongxu Yin, Greg Heinrich, Jan Kautz, and Pavlo Molchanov. Global context vision transformers. In *ICML*, 2023. 5, 6

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Sun Jian. Deep residual learning for image recognition. In *CVPR*, 2016. 6

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross B. Girshick. Mask r-cnn. In *ICCV*, 2017. 5

[25] Qibin Hou, Cheng-Ze Lu, Ming-Ming Cheng, and Jiashi Feng. Conv2former: A simple transformer-style convnet for visual recognition. *arXiv preprint arXiv:2211.11943*, 2022. 5

[26] Gao Huang, Yu Sun, and Zhuang Liu. Deep networks with stochastic depth. In *ECCV*, 2016. 4

[27] Huaibo Huang, Xiaoqiang Zhou, and Ran He. Orthogonal transformer: An efficient vision transformer backbone with token orthogonalization. In *NeurIPS*, 2022. 5, 6

[28] Huaibo Huang, Xiaoqiang Zhou, Jie Cao, Ran He, and Tieniu Tan. Vision transformer with super token sampling. In *CVPR*, 2023. 5, 6

[29] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, et al. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 2

[30] Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. All tokens matter: Token labeling for training better vision transformers. In *NeurIPS*, 2021. 1, 4, 5

[31] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *CVPR*, 2019. 5

[32] Youngwan Lee, Jonghee Kim, Jeffrey Willette, and Sung Ju Hwang. Mpvit: Multi-path vision transformer for dense prediction. In *CVPR*, 2022. 1, 5, 6, 7

[33] Kunchang Li, Yali Wang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatiotemporal representation learning, 2022. 1, 4, 5, 6, 7

[34] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *CVPR*, 2022. 1, 5

[35] Yehao Li, Ting Yao, Yingwei Pan, and Tao Mei. Contextual transformer networks for visual recognition. *TPAMI*, 2022. 5

[36] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, and Kaiming He andPiotr Dollár. Focal loss for dense object detection. In *ICCV*, 2017. 5

[37] Tsung-Yi Lin, Michael Maire, Serge Belongie, et al. Microsoft coco: Common objects in context. In *ECCV*, 2014. 4

[38] Weifeng Lin, Ziheng Wu, Jiayu Chen, Jun Huang, and Lianwen Jin. Scale-aware modulation meet transformer. In *ICCV*, 2023. 1, 5, 6, 7, 8

[39] 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. In *ICCV*, 2021. 1, 2, 3, 4, 5, 6, 7, 8

[40] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, et al. A convnet for the 2020s. In *CVPR*, 2022. 5, 6

[41] Jiasen Lu, Christopher Clark, Rowan Zellers, Roozbeh Mottaghi, and Aniruddha Kembhavi. Unified-io: A unified model for vision, language, and multi-modal tasks. In *ICLR*, 2023. 2

[42] Xiaofeng Mao, Gege Qi, Yuefeng Chen, Xiaodan Li, Ranjie Duan, Shaokai Ye, Yuan He, and Hui Xue. Towards robust vision transformer. In *CVPR*, 2022. 5

[43] Junting Pan, Adrian Bulat, Fuwen Tan, et al. Edgevits: Competing light-weight cnns on mobile devices with vision transformers. In *ECCV*, 2022. 7

[44] Zizheng Pan, Jianfei Cai, and Bohan Zhuang. Fast vision transformers with hilo attention. In *NeurIPS*, 2022. 2, 5

[45] Ofir Press, Noah Smith, and Mike Lewis. Train short, test long: Attention with linear biases enables input length extrapolation. In *ICLR*, 2022. 1, 3

[46] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2

[47] Yongming Rao, Wenliang Zhao, Yansong Tang, Jie Zhou, Ser-Lam Lim, and Jiwen Lu. Hornet: Efficient high-order spatial interactions with recursive gated convolutions. In *NeurIPS*, 2022. 6

[48] Sucheng Ren, Daquan Zhou, Shengfeng He, Jiashi Feng, and Xinchao Wang. Shunted self-attention via multi-scale token aggregation. In *CVPR*, 2022. 7

[49] Chenyang Si, Weihao Yu, Pan Zhou, Yichen Zhou, Xinchao Wang, and Shuicheng YAN. Inception transformer. In *NeurIPS*, 2022. 4, 5

[50] Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to Transformer for large language models. *ArXiv*, abs/2307.08621, 2023. 1, 2, 3

[51] Shitao Tang, Jiahui Zhang, Siyu Zhu, et al. Quadtree attention for vision transformers. In *ICLR*, 2022. 5

[52] Hugo Touvron, Matthieu Cord, Matthijs Douze, et al. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 2, 4, 7, 8

[53] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *ICCV*, 2021. 1

[54] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *ECCV*, 2022. 1, 4, 5

[55] Ashish Vaswani, Noam Shazeer, Niki Parmar, et al. Attention is all you need. In *NeurIPS*, 2017. 2

[56] 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 *ICCV*, 2021. 2, 3, 4, 6

[57] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):1–10, 2022. 2, 3, 4, 5, 6, 7

[58] Wenxiao Wang, Lu Yao, Long Chen, Binbin Lin, Deng Cai, Xiaofei He, and Wei Liu. Crossformer: A versatile vision transformer hinging on cross-scale attention. In *ICLR*, 2022. 5, 7

[59] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *CVPR*, 2023. 5, 6, 7

[60] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021. 1, 2

[61] Zhuofan Xia, Xuran Pan, Shiji Song, Li Erran Li, and Gao Huang. Vision transformer with deformable attention. In *CVPR*, 2022. 2, 6, 7

[62] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 5

[63] Haiyang Xu, Qinghao Ye, Ming Yan, Yaya Shi, Jiabo Ye, Yuanhong Xu, Chenliang Li, Bin Bi, Qi Qian, Wei Wang, Guohai Xu, Ji Zhang, Songfang Huang, Fei Huang, and Jingren Zhou. mplug-2: A modularized multi-modal foundation model across text, image and video. In *ICML*, 2023. 2

[64] Chenglin Yang, Yilin Wang, Jianming Zhang, et al. Lite vision transformer with enhanced self-attention. In *CVPR*, 2022. 1

[65] Chenglin Yang, Siyuan Qiao, Qihang Yu, et al. Moat: Alternating mobile convolution and attention brings strong vision models. In *ICLR*, 2023. 5, 7

[66] 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. In *NeurIPS*, 2021. 2, 3, 5, 6

[67] Rui Yang, Hailong Ma, Jie Wu, Yansong Tang, Xuefeng Xiao, Min Zheng, and Xiu Li. Scalablevit: Rethinking the context-oriented generalization of vision transformer. In *ECCV*, 2022. 5, 6

[68] Ting Yao, Yingwei Pan, Yehao Li, Chong-Wah Ngo, and Tao Mei. Wave-vit: Unifying wavelet and transformers for visual representation learning. In *Proceedings of the European conference on computer vision (ECCV)*, 2022. 2, 5

[69] Ting Yao, Yehao Li, Yingwei Pan, Yu Wang, Xiao-Ping Zhang, and Tao Mei. Dual vision transformer. *TPAMI*, 2023. 5

[70] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *TPAMI*, 2022. 4, 5, 7

[71] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, et al. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 4

[72] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, et al. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 4

[73] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *ICCV*, 2021. 4

[74] Qiming Zhang, Yufei Xu, Jing Zhang, and Dacheng Tao. Vsa: Learning varied-size window attention in vision transformers. In *ECCV*, 2022. 4

[75] Zhun Zhong, Liang Zheng, Guoliang Kang, et al. Random erasing data augmentation. In *AAAI*, 2020. 4

[76] Bolei Zhou, Hang Zhao, Xavier Puig, et al. Scene parsing through ade20k dataset. In *CVPR*, 2017. 4

[77] Daquan Zhou, Zhiding Yu, Enze Xie, Chaowei Xiao, Anima Anandkumar, Jiashi Feng, and Jose M. Alvarez. Understanding the robustness in vision transformers. In *ICML*, 2022. 7

[78] Lei Zhu, Xinjiang Wang, Zhanghan Ke, Wayne Zhang, and Rynson Lau. Biformer: Vision transformer with bi-level routing attention. In *CVPR*, 2023. 1, 2, 4, 5, 7, 8