Vision-RWKV: Efficient and Scalable Visual Perception with RWKV-Like Architectures

Yuchen Duan[*2,1], Weiyun Wang[*3,1], Zhe Chen[*4,1], Xizhou Zhu[5,1,6], Lewei Lu[6], Tong Lu[4], Yu Qiao[1], Hongsheng Li[2], Jifeng Dai[5,1], and Wenhai Wang[2,1✉]

[1]OpenGVLab, Shanghai AI Laboratory
[2]The Chinese University of Hong Kong [3]Fudan University
[4]Nanjing University [5]Tsinghua University [6]SenseTime Research

Abstract. Transformers have revolutionized computer vision and natural language processing, but their high computational complexity limits their application in high-resolution image processing and long-context analysis. This paper introduces Vision-RWKV (VRWKV), a model adapted from the RWKV model used in the NLP field with necessary modifications for vision tasks. Similar to the Vision Transformer (ViT), our model is designed to efficiently handle sparse inputs and demonstrate robust global processing capabilities, while also scaling up effectively, accommodating both large-scale parameters and extensive datasets. Its distinctive advantage lies in its reduced spatial aggregation complexity, which renders it exceptionally adept at processing high-resolution images seamlessly, eliminating the necessity for windowing operations. Our evaluations in image classification demonstrate that VRWKV matches ViT's classification performance with significantly faster speeds and lower memory usage. In dense prediction tasks, it outperforms window-based models, maintaining comparable speeds. These results highlight VRWKV's potential as a more efficient alternative for visual perception tasks. Code is released at https://github.com/OpenGVLab/Vision-RWKV.

Keywords: RWKV · Visual Perception · Linear Attention

1 Introduction

Vision Transformers (ViTs) [12, 19, 44, 48, 51], renowned for their flexibility and global information processing capabilities, have established new benchmarks in a variety of vision tasks in the past few years. However, the quadratic computational complexity associated with ViTs limits their ability to efficiently process high-resolution images and lengthy sequences, posing a significant barrier to their broader application. As a result, the exploration of a vision architecture that integrates the versatility and comprehensive processing strengths of ViTs, while reducing computational demands, has emerged as a crucial area of research.

In recent developments within natural language processing (NLP), models like RWKV [37] and Mamba [16] have emerged as popular solutions for achieving heightened efficiency and processing lengthy texts. These innovative models

* Equal contribution; ✉ Corresponding author (wangwenhai362@gmail.com)

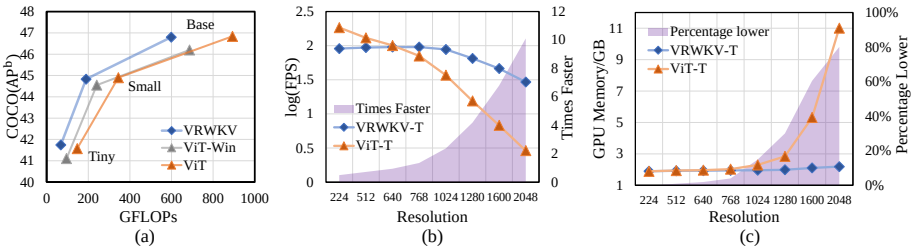


Fig. 1: Performance and efficiency comparison of Vision-RWKV (VRWKV) and ViT. (a) AP^b comparison of VRWKV and ViT [48] with window attention and global attention on the COCO [29] dataset. (b) Inference speed comparison of VRWKV-T and ViT-T across input resolutions ranging from 224 to 2048. (c) GPU memory comparison of VRWKV-T and ViT-T across input resolutions from 224 to 2048.

have demonstrated attributes similar to transformers [3,9,27,30,38–40,43,45] in NLP tasks, including the ability to handle long-range dependencies and parallel processing. Furthermore, they have also proven to be scalable, performing well with large-scale NLP datasets. Considering the significant differences between image and text modalities, it remains challenging to envision these methods entirely supplanting ViTs for vision-related tasks. It is imperative to conduct an in-depth analysis of how these models are applied to vision tasks, examining their scalability concerning data and parameters, their efficiency in handling sparse visual data, such as masked image modeling, and the necessary techniques to ensure model stability during scaling up.

In this work, we introduce Vision-RWKV (VRWKV), which is designed to adapt the RWKV architecture for vision tasks. This adaptation preserves the core structure and benefits of RWKV [37] while integrating critical modifications to tailor it for processing visual data. Specifically, we introduce a quad-directional shift (Q-Shift) tailed for vision tasks and modify the original causal RWKV attention mechanism to a bidirectional global attention mechanism. The Q-Shift operation expands the semantic range of individual tokens, while the bidirectional attention enables the calculation of global attention within linear computational complexity in an RNN form forward and backward. We primarily make modifications to the exponent in the RWKV attention mechanism, releasing the limitations of the decay vector and transforming the absolute positional bias into a relative bias. These changes enhance the model's capability while ensuring scalability and stability. In this way, our model inherits the efficiency of RWKV in handling global information and sparse inputs, while also being able to model the local concept of vision tasks. We implement layer scale [49] and layer normalization [2] where needed to stabilize the model's output across different scales. These adjustments significantly improve the model's stability when scaling up to a larger size.

Building on the aforementioned design, we develop a range of VRWKV models with different model scales, spanning from the VRWKV-Tiny (6M) to the VRWKV-Large (335M). These models are trained using large-scale datasets such

as ImageNet-1K [8] and ImageNet-22K [8]. We train them using both common supervised classification and sparse input MAE methods [19] and evaluate their performance on visual perception tasks, including classification, detection, and segmentation. Under the same settings, VRWKV has comparable performance to ViT in these tasks with lower computational costs while maintaining stable scalability. This achievement enables VRWKV training parallelism, high flexibility, excellent performance, and low inference cost simultaneously, making it a promising alternative to ViT in a wide range of vision tasks, particularly in high-resolution scenarios.

In this paper, our main contributions are:

(1) We propose VRWKV as a low-cost alternative to ViT, achieving comprehensive substitution with lower computational costs. Our model not only retains the advantages of ViT, including the capability to capture long-range dependencies and flexibility in handling sparse inputs but also reduces complexity to a linear level. This significant reduction eliminates the need for window-based attention in processing high-resolution images, making VRWKV a more efficient and scalable solution for vision tasks.

(2) To adapt to vision tasks, we introduce bidirectional global attention and a novel token shift method called Q-Shift, enabling the achievement of linear complexity in global attention. To ensure stable scalability, we make several efforts, including using a relative positional bias in the attention mechanism to avoid overflow, adopting layer scale in our model, and adding extra layer normalization in the calculation of key matrices.

(3) Our model surpasses window-based ViTs and is comparable to global attention ViTs, demonstrating lower FLOPs and faster processing speeds as resolution increases. Notably, VRWKV-T achieves 75.1% top-1 accuracy trained only on the ImageNet-1K [8], outperforming DeiT-T [48] by 2.9 points. With large-scale parameters (*i.e.*, 335M) and training data (*i.e.*, ImageNet-22K), the top-1 accuracy of VRWKV-L is further boosted to 85.3%, which is slightly higher than ViT-L [13] (85.28 vs. 85.15). In addition, on COCO [29], a challenging downstream benchmark, our best model VRWKV-L achieves 50.6% box mAP, 1.9 points better than ViT-L (50.6 vs. 48.7).

2 Related Works

2.1 Vision Encoder

Recent advances in vision encoders have significantly pushed the boundaries of computer vision, demonstrating remarkable performance across a range of tasks. Convolutional neural networks (CNNs) served as the foundational model in computer vision. The advancement of computational resources, such as GPUs, has enabled the successful training of stacked convolutional blocks like AlexNet [25] and VGG [41] on large-scale image classification datasets (*e.g.*, ImageNet [8]). This development paved the way for deeper and more sophisticated convolutional neural architectures, including GoogleNet [47], ResNet [21], and DenseNet [24].

In addition to these innovations, significant advancements have also been made with architectures like SENet [23], which introduced a channel attention mechanism to enhance model sensitivity to informative features. Similarly, SKNet [28] merged multiple kernel sizes to adjust the receptive field adaptively. Further extending the CNN paradigm, recent models such as RepLKNet [11] and ConvNeXt [33] have refined the convolutional layers to improve efficiency and accuracy, while InternImage [54] explored the strategies to scale up the convolution-based vision model.

Drawing inspiration from the effectiveness of self-attention layers and transformer architectures in the NLP field, the Vision Transformer (ViT) [13] applied a transformer framework on image patches, offering a global receptive field and dynamic spatial aggregation. Due to the quadratically increasing computational complexity of the vanilla attention mechanism, approaches like PVT [55,57] and Linformer [53] implemented global attention on down-sampled feature maps, whereas other approaches like Swin [58] and HaloNet [6,50] introduced sampling techniques to enlarge the receptive field.

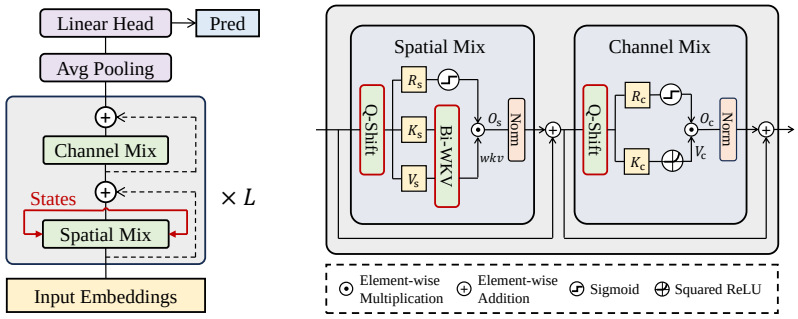
Another research direction involved replacing self-attention layers in models with linear complexity layers. Representative works in this domain include LongNet [10], RWKV [37], RetNet [46], and Mamba [16], though few have concentrated on vision-related applications. Concurrently, attempts like Vim [63] and VMamba [31] have sought to integrate these linear attention layers into the vision domain. However, these endeavors have only been experimented with on small-scale models, and it remains uncertain whether their efficiency can scale up to larger models.

2.2 Feature Aggregation Mechanism

The research on feature aggregation has received significant attention in the field of artificial intelligence. For visual data processing, convolutional operators [26], known for their parameter sharing and local perception, enabled efficient handling of large-scale data through sliding computation. Despite their advantages, traditional CNN operators faced challenges in modeling long-range dependencies. To overcome this issue, advanced convolutional operators, such as the deformable convolution [5,60,64], have improved the flexibility of CNN operators, enhancing their long-range modeling capability.

As for the field of NLP, RNN-based operators [14,36] have historically dominated because of their effectiveness in sequence modeling. RNNs and LSTMs excel in capturing temporal dependencies, making them suitable for tasks requiring an understanding of sequence dynamics. Subsequently, a significant shift occurred. The introduction of the transformer architecture [51] marked a turning point, with both NLP and computer vision fields shifting focus toward attention-based feature aggregation. The global attention mechanism overcomes the limitations of CNNs in modeling long-range dependencies and the shortcomings of RNNs in parallel computation while coming at a high computational cost.

To address the high computational cost of attention operators while modeling long sequences, researchers have introduced innovations such as window atten-



(a) Vision-RWKV Architecture (b) Vision-RWKV Encoder Layer

Fig. 2: Overall architecture of Vision-RWKV (VRWKV). (a) The VRWKV architecture includes L identical VRWKV encoder layers, an average pooling layer, and a linear prediction head. (b) The details of the VRWKV encoder layer. Q-Shift denotes the quad-directional shift method tailed for vision tasks. The "Bi-WKV" module served as a bidirectional RNN cell or a global attention mechanism.

tion and spatial reduction attention. Window attention [6, 32, 50] restricts the self-attention computation within local windows, drastically reducing the computational complexity while preserving the receptive field through window-level interaction. Spatial reduction attention [55, 57], on the other hand, reduces the dimensionality of the feature space before applying the attention mechanism, effectively decreasing the computational requirements without significantly degrading the model's performance.

In addition to the efforts to optimize the global attention mechanism, various operators with linear complexity [16,37,46] have also been explored. For instance, RWKV [37] and RetNet [46] employed exponential decay to model global information efficiently while SSMs [17, 18, 42, 52] also exhibited linear complexity concerning sequence length and modification in Mamba [16] enable them to be input-dependent. Besides, XCA [1] achieved linear complexity by calculating the cross-variance between input tokens. However, the low efficiency of information interaction between tokens makes the need for the assistance of additional modules necessary to complete a comprehensive feature aggregation. Despite some concurrent efforts [15, 31, 63], adapting these NLP-derived techniques to vision tasks remains a challenge in maintaining stable training across larger and more complex vision models.

3 Vision-RWKV

3.1 Overall Architecture

In this section, we propose Vision-RWKV (VRWKV), an efficient vision encoder with a linear complexity attention mechanism. Our principle is to preserve the

advantages of the original RWKV architecture [37], making only necessary modifications to enable its flexible application in vision tasks, supporting sparse input, and ensuring the stability of the training process after scaling up. An overview of our VRWKV is depicted in Fig. 2.

VRWKV adopts a block-stacked image encoder design like ViT, where each block consists of a spatial-mix module and a channel-mix module. The spatial-mix module functions as an attention mechanism, performing linear complexity global attention computation while the channel mix module serves as a feed-forward network (FFN), performing feature fusion in the channel dimension. The entire VRWKV includes a patch embedding layer and a stack of L identical VRWKV encoder layers, where each layer maintains the input resolution.

Data Flow. First, we transform the $H \times W \times 3$ image into HW/p^2 patches, where p denotes the patch size. The patches after a linear projection add the position embedding to obtain image tokens of shape $T \times C$, where $T = HW/p^2$ denotes the total number of tokens. These tokens are then input into the VRWKV encoder with L layers.

In each layer, tokens are first fed into the spatial-mix module which plays the role of a global attention mechanism. Specifically, as shown in Fig. 2(b), the input tokens are first shifted and fed into three parallel linear layers to obtain the matrices $R_s, K_s, V_s \in \mathbb{R}^{T \times C}$:

$$R_{\mathrm{s}} = \text{Q-Shift}_R(X)W_R, \quad K_{\mathrm{s}} = \text{Q-Shift}_K(X)W_K, \quad V_{\mathrm{s}} = \text{Q-Shift}_V(X)W_V. \tag{1}$$

Here, K_{s} and V_{s} are passed to calculate the global attention result $wkv \in \mathbb{R}^{T \times C}$ by a linear complexity bidirectional attention mechanism and multiplied with $\sigma(R)$ which controls the output O_{s} probability:

$$\begin{aligned} O_{\mathrm{s}} &= (\sigma(R_{\mathrm{s}}) \odot wkv)W_O, \\ \text{where } wkv &= \text{Bi-WKV}(K_{\mathrm{s}}, V_{\mathrm{s}}). \end{aligned} \tag{2}$$

Operator σ denotes the sigmoid function, and $\odot$ means an element-wise multiplication is applied. The Q-Shift is a token shift function specially designed for the adaption of vision tasks. After an output linear projection, features are then stabilized using layer normalization [2].

Subsequently, the tokens are passed into the channel-mix module for a channel-wise fusion. R_{c}, K_{c} are obtained in a similar manner as spatial-mix:

$$R_{\mathrm{c}} = \text{Q-Shift}_R(X)W_R, \quad K_{\mathrm{c}} = \text{Q-Shift}_K(X)W_K. \tag{3}$$

Here, V_{c} is a linear projection of K after the activation function and the output O_{c} is also controlled by a gate mechanism $\sigma(R_{\mathrm{c}})$ before the output projection:

$$\begin{aligned} O_{\mathrm{c}} &= (\sigma(R_{\mathrm{c}}) \odot V_{\mathrm{c}})W_O, \\ \text{where } V_{\mathrm{c}} &= \text{SquaredReLU}(K_{\mathrm{c}})W_V. \end{aligned} \tag{4}$$

Simultaneously, residual connections [21] are established from the tokens to each normalization layer to ensure that training gradients do not vanish in deep networks.

3.2 Linear Complexity Bidirectional Attention

Different from the vanilla RWKV [37], we make the following modifications to its original attention mechanism to adapt it to vision tasks: (1) **Bidirectional attention**: We extend the upper limit of original RWKV attention from t (the current token) to $T-1$ (the last token) in the summation formula to ensure that all tokens are mutually visible in the calculation of each result. Thus, the original causal attention transforms into bidirectional global attention. (2) **Relative bias**: We compute the absolute value of the time difference $t-i$ and divide it by the total number of tokens (denoted as T) to represent the relative bias of tokens in images of different sizes. (3) **Flexible decay:** We no longer restrict the learnable decay parameter w to be positive in the exponential term allowing the exponential decay attention to focus on tokens further away from the current token in different channels. The simple yet necessary modification achieves global attention calculation and maximizes the preservation of RWKV's low complexity and adaptability to vision tasks.

Similar to the attention in RWKV, our bidirectional attention can also be equivalently expressed in a summation form (for the sake of clarity) and an RNN form (in practical implementation).

Summation Form. The attention calculation result for the t-th token is given by the following formula:

$$wkv_t = \text{Bi-WKV}(K, V)_t = \frac{\sum_{i=0, i\neq t}^{T-1} e^{-(|t-i|-1)/T\cdot w + k_i} v_i + e^{u+k_t} v_t}{\sum_{i=0, i\neq t}^{T-1} e^{-(|t-i|-1)/T\cdot w + k_i} + e^{u+k_t}}. \tag{5}$$

Here, T represents the total number of tokens, equal to HW/p^2, w and u are two C-dimensional learnable vectors that represent channel-wise spatial decay and the bonus indicating the current token, respectively. k_t and v_t denotes t-th feature of K and V.

The summation formula indicates that the output wkv_t is a weighted sum of V along the token dimension from 0 to $T-1$, resulting in a C-dimensional vector. It represents the result obtained by applying the attention operation to the t-th token. The weight is determined by the spatial decay vector w, the relative bias between tokens $(|t-i|-1)/T$, and k_i collectively.

RNN Form. In the practical implementation, the above Eq. (5) can be transformed into a recursive formula in the form of RNN that the result of each token can be obtained through a fixed number of FLOPs. By splitting the summation term of the denominator and numerator in Eq. (5) with t as the boundary, we can obtain 4 hidden states:

$$\begin{aligned} a_{t-1} &= \sum_{i=0}^{t-1} e^{-(|t-i|-1)/T\cdot w + k_i} v_i, \quad & b_{t-1} &= \sum_{i=t+1}^{T-1} e^{-(|t-i|-1)/T\cdot w + k_i} v_i, \\ c_{t-1} &= \sum_{i=0}^{t-1} e^{-(|t-i|-1)/T\cdot w + k_i}, & d_{t-1} &= \sum_{i=t+1}^{T-1} e^{-(|t-i|-1)/T\cdot w + k_i}, \end{aligned} \tag{6}$$

that can be recursively computed. The update of hidden states only requires adding or subtracting one summation term and multiplying or dividing $e^{-w/T}$. Then the t-th result can be given as:

$$wkv_t = \frac{a_{t-1} + b_{t-1} + e^{k_t+u}v_t}{c_{t-1} + d_{t-1} + e^{k_t+u}}. \tag{7}$$

Each update step yields an attention result (*i.e.*, wkv_t) for a token, so the entire wkv matrix requires T steps.

When the input K and V are matrices with the shape of $T \times C$, the computational cost of calculating the wkv matrix is given by:

$$\text{FLOPs}(\text{Bi-WKV}(K, V)) = 13 \times T \times C. \tag{8}$$

Here, the number 13 is approximately from the updates of (a, b, c, d), the computation of the exponential, and the calculation of wkv_t. T is the total number of update steps and is equal to the number of image tokens. The above approximation shows that the complexity of the forward process is $O(TC)$. The backward propagation of the operator can still be represented as a more complex RNN form, with a computational complexity of $O(TC)$. The specific formula for backward propagation is provided in the Appendix.

3.3 Quad-Directional Token Shift

By introducing an exponential decay mechanism, the complexity of global attention can be reduced from quadratic to linear, greatly enhancing the computational efficiency of the model in high-resolution images. However, the one-dimensional decay does not align with the neighboring relationships in two-dimensional images. Therefore, we introduce a quad-directional token shift (Q-Shift) in the first step of each spatial-mix and channel-mix module. The Q-Shift operation allows all tokens shifted and linearly interpolated with their neighboring tokens as follows:

$$\begin{aligned} &\text{Q-Shift}_{(*)}(X) = X + (1 - \mu_{(*)})X^{\dagger}, \\ &\text{where } X^{\dagger}[h, w] = \text{Concat}(X[h-1, w, 0 : C/4], X[h+1, w, C/4 : C/2], \\ &\qquad\qquad X[h, w-1, C/2 : 3C/4], X[h, w+1, 3C/4 : C]). \end{aligned} \tag{9}$$

Subscript $(*) \in \{R, K, V\}$ denotes 3 interpolation of X and $X^{\dagger}$ controlled by the learnable vectors $\mu_{(*)}$ for the later calculation of R, K, V, respectively. h and w denote the row and column index of token X, ":" is a slicing operation excluded the end index. The Q-Shift makes the attention mechanism of different channels obtain the prior of focusing on neighboring tokens internally without introducing many additional FLOPs. The Q-Shift operation also increases the receptive field of each token which greatly enhances the coverage of the token in the posterior layers.

3.4 Scale Up Stability

Both the number of model layers increasing and the accumulation in the exponent term during the recursion can lead to instability in the model output

Model	Emb Dim	Hidden Dim	Depth	Extra Norm	#Param
VRWKV-T	192	768	12	×	6.2M
VRWKV-S	384	1536	12	×	23.8M
VRWKV-B	768	3072	12	×	93.7M
VRWKV-L	1024	4096	24	✓	334.9M

Table 1: Default settings for Vision-RWKV of different scales. We report the embedding dimension, hidden dimension, and model depth for VRWKV-T/S/B/L. "Extra Norm" means additional layer normalization layers are used to stabilize the model's outputs. "#Param" denotes the number of parameters.

and affect the stability of the training process. To mitigate the instability, we employ two simple but effective modifications to stabilize the scale-up of the model. (1) **Bounded exponential**: As the input resolution increases, both exponential decay and growth quickly exceed the range of floating-point numbers. Therefore, we divide the exponential term by the number of tokens (such as $\exp(-(|t-i|-1)/T \cdot w)$), making the maximum decay and growth bounded. (2) **Extra layer normalization**: When the model gets deeper, we directly add layer normalization [2] after the attention mechanism and Squared ReLU operation to prevent the model's output from overflowing. The two modifications enable stable scaling of both input resolution and model depth, allowing large models to train and converge stably. We also introduce layer scale [49] which contributes to the stability of the models as they scale up.

3.5 Model Details

Following ViT, the hyper-parameters for variants of VRWKV, including embedding dimension, hidden dimension in linear projection, and depth, are specified in Tab. 1. Due to the increased depth of the VRWKV-L model, additional layer normalizations as discussed in Sec. 3.4, are incorporated at appropriate positions to ensure output stability.

4 Experiments

We comprehensively evaluate the substitutability of our VRWKV method for ViT in performance, scalability, flexibility, and efficiency. We validate the effectiveness of our model on the widely-used image classification dataset ImageNet [8]. For downstream dense prediction tasks, we select detection tasks on the COCO [29] dataset and semantic segmentation on the ADE20K [62] dataset.

4.1 Image Classification

Settings. For -Tiny/Small/Base models, we conduct supervised training from scratch on ImageNet-1K [8]. Following the training strategy and data augmentation of DeiT [48], we use a batch size of 1024, AdamW [35] with a base learning rate of 5e-4, weight decay of 0.05, and cosine annealing schedule [34]. Images

	Method	Size	#Param	FLOPs	Top-1 Acc
hierarchical	ResNet-18 [22]	224^2	11.7M	1.8G	69.9
	PVT-T [56]	224^2	13.2M	1.9G	75.1
	ResNet-50 [22]	224^2	25.6M	4.1G	76.6
	Swin-T [32]	224^2	28.3M	4.4G	81.2
	PVT-M [56]	224^2	44.2M	6.7G	81.2
	ResNet-101 [22]	224^2	44.6M	7.9G	78.0
	Swin-S [32]	224^2	49.6M	8.7G	83.0
	PVT-L [56]	224^2	61.4M	9.8G	81.7
	Swin-B [32]	224^2	87.8M	15.1G	83.4
non-hierarchical	DeiT-T [48]	224^2	5.7M	1.3G	72.2
	DeiT-S [48]	224^2	22.1M	4.6G	79.9
	XCiT-S12 [1]	224^2	26.0M	4.8G	82.0
	DeiT-B [48]	224^2	86.6M	17.6G	81.8
	XCiT-L24 [1]	224^2	189.0M	36.1G	82.9
	ViT-L [13]	384^2	309.5M	191.1G	85.2
	VRWKV-T	224^2	6.2M	1.2G	75.1
	VRWKV-S	224^2	23.8M	4.6G	80.1
	VRWKV-B	224^2	93.7M	18.2G	82.0
	VRWKV-L	384^2	334.9M	189.5G	85.3
	VRWKV-L*	384^2	334.9M	189.5G	86.5

Table 2: Validation results on ImageNet-1K. VRWKV-T/S/B are trained from scratch using ImageNet-1K, while VRWKV-L is pre-trained on Imagenet-22K and fine-tuned on Imagenet-1K. "#Param" denotes the number of parameters, and "FLOPs" means the computational workload for processing an image at the resolution specified in the "Size" column. "$\star$" denotes Bamboo-47K [61] is used in the pre-training.

are cropped to the resolution of 224×224 for training and validation. For the -Large models, we first pre-train them for 30 epochs on ImageNet-22K with a batch size of 4096 and resolution of 192×192, and then fine-tune them for 20 epochs on ImageNet-1K to a higher resolution of 384×384.

Results. We compare the results of our VRWKV with other hierarchical and non-hierarchical backbones on the ImageNet-1K dataset. As shown in Tab. 2, with the same number of parameters, computational complexity, and training/testing resolutions, VRWKV achieves comparable results to ViT. For example, compared to ViT-L, VRWKV-L achieves a similar top-1 accuracy of 85.3% at a resolution of 384×384, with a slightly reduced computational cost. When the model size is smaller, VRWKV demonstrates higher baseline performance. In the case where both VRWKV-T and DeiT-T have FLOPs of 1.3G, VRWKV-T achieves a 2.9 point higher than DeiT-T. The exploration and utilization of linear attention mechanisms in VRWKV demonstrate its potential in vision tasks, making it a viable alternative to traditional ViT models that use global attention mechanisms. The comparable performance from tiny to large-size models also demonstrates that the VRWKV model possesses the scalability as ViT.

Method	#Param	FLOPs	AP^b	AP^m
ViT-T† [48]	8.0M	95.4G	41.1	37.5
ViT-T [48]	8.0M	147.1G	41.6	37.9
VRWKV-T (ours)	8.4M	67.9G	41.7	38.0
ViT-S† [48]	27.5M	241.2G	44.6	39.7
ViT-S [48]	27.5M	344.5G	44.9	40.1
VRWKV-S (ours)	29.3M	189.9G	44.8	40.2
ViT-B† [48]	99.5M	686.7G	46.2	41.5
ViT-B [48]	99.5M	893.3G	46.8	41.8
VRWKV-B (ours)	106.6M	599.0G	46.8	41.7
ViT-L† [44]	327.0M	1799.3G	48.7	43.3
VRWKV-L (ours)	351.9M	1730.6G	50.6	44.9

Table 3: Object detection and instance segmentation on COCO val2017. All models adopt the ViT-Adapter [4] to generate multi-scale features for detection heads. -T/S/B models are initialized with ImageNet-1K pre-trained weights, and all -L models with ImageNet-22K weights. "#Param" denotes the number of backbone parameters. "FLOPs" denotes the computational workload of the backbone with an input image of 1333×800. "†" means window attention is adopted in ViT layers.

4.2 Object Detection

Settings. In the detection tasks, we adopt Mask R-CNN [20] as the detection head. For the -Tiny/Small/Base models, the backbones use weights pre-trained on ImageNet-1K for 300 epochs. For the -Large model, weights pre-trained on ImageNet-22K are used. All models use a $1\times$ training schedule (*i.e.*, 12 epochs) with a batch size of 16, and AdamW [35] optimizer with an initial learning rate of 1e-4 and weight decay of 0.05.

Results. In Tab. 3, we report the detection results on the COCO val [29] dataset using VRWKV and ViT as backbones. As the results showed in Fig. 1 (a) and Tab. 3, due to the use of window attention in dense prediction tasks, VRWKV with global attention can achieve better performance than ViT with lower FLOPs. For example, VRWKV-T has approximately 30% lower backbone FLOPs compared to ViT-T†, with an improvement of AP^b by 0.6 points. Similarly, VRWKV-L achieves a 1.9-point increase in AP^b with lower FLOPs compared to ViT-L†. Additionally, we compare the performance of VRWKV and ViT using global attention. For instance, VRWKV-S achieves similar performance to ViT-S with 45% lower FLOPs. This demonstrates the effectiveness of VRWKV's global attention mechanism in dense prediction tasks and the advantage of lower computational complexity compared to the original attention mechanism.

4.3 Semantic Segmentation

Settings. In the semantic segmentation task, we use UperNet [59] as the segmentation head. Specifically, all ViT models use global attention in the segmentation

Method	#Param	FLOPs	mIoU
ViT-T [48]	8.0M	20.9G	42.6
VRWKV-T (ours)	8.4M	16.6G	43.3
ViT-S [48]	27.5M	54.0G	46.2
VRWKV-S (ours)	29.3M	46.3G	47.2
ViT-B [48]	99.5M	157.9G	48.8
VRWKV-B (ours)	106.6M	146.0G	49.2
ViT-L [44]	327.0M	446.8G	53.4
VRWKV-L (ours)	351.9M	421.9G	53.5

Table 4: Semantic segmentation on the ADE20K val set. All models used ViT-Adapter [4] for multi-scale feature generation and are trained with UperNet [59] as the segmentation heads. For consistency in comparison, all -T/S/B models are initialized using ImageNet-1K pre-training, whereas -L models utilize ImageNet-22K pre-training. "#Param" refers to the number of parameters of the backbone. We report the FLOPs of backbones with the input size of 512×512.

task. For the -Tiny/Small/Base models, the backbones use weights pre-trained on ImageNet-1K. And for the -Large model, weights pre-trained on ImageNet-22K are used. We employ the AdamW optimizer with an initial learning rate of 6e-5 for the -Small/Base/Large models and 12e-5 for the -Tiny model, a batch size of 16, and a weight decay of 0.01. All models are trained for 160k iterations on the training set of the ADE20K dataset [62].

Results. As shown in Tab. 4, when using UperNet for semantic segmentation, models based on VRWKV consistently outperform those based on ViT with global attention, while also being more efficient. For example, VRWKV-S achieves 1 point higher than ViT-S with a 14% FLOPs decrease. VRWKV-L creates a result of 53.5 mIoU similar to ViT-L while the computation of the backbone has a 25G FLOPs lower. These results demonstrate that our VRWKV backbones can extract better features for semantic segmentation compared to ViT backbones while also being more efficient, benefiting from the linear complexity attention mechanism.

4.4 Ablation Study

Settings. We conduct ablation studies of the tiny-size VRWKV on ImageNet-1K [8] to validate the effectiveness of various key components like Q-Shift and bidirectional attention. The experimental settings are consistent with Sec. 4.1.

Token Shift. We compare the performance of not using token shift, using the original shift method in RWKV [37], and our Q-Shift. As shown in Tab. 5, the variation in the shift method shows performance differences. Variant 1 without token shift leads to a poor performance of 71.5, which is 3.6 points lower than our model. Even with the use of our global attention, the model using the original token shift still has a 0.7-point gap with our model.

Method	Token Shift	Bidirectional Attention	Top-1 Acc
RWKV [37]	original	×	71.1 (-4.0)
Variant 1	none	✓	71.5 (-3.6)
Variant 2	original	✓	74.4 (-0.7)
Variant 3	Q-Shift	×	72.8 (-2.3)
VRWKV-T (ours)	Q-Shift	✓	75.1

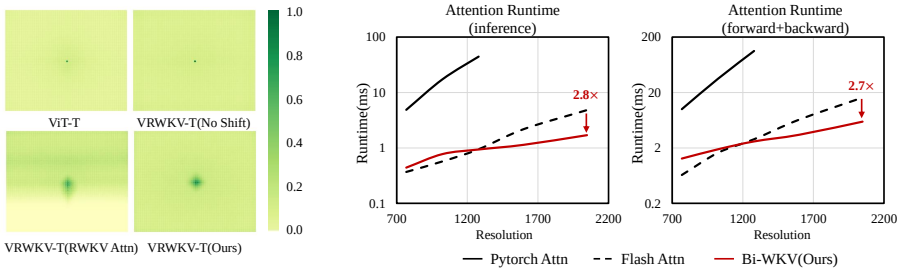
Table 5: Ablation on key components of the proposed VRWKV. All models are trained from scratch on ImageNet-1K. The "original" means the original token shift in RWKV [37], which mixes the token in a single direction.

Bidirectional Attention. The bidirectional attention mechanism enables the model to achieve global attention while the original RWKV attention has a causal mask internally. The result of Variant 3 shows that the global attention mechanism brings a 2.3 points increase in the top-1 accuracy.

Effective Receptive Field (ERF). We analyze the impact of different designs on the ERF of models based on [11] and visualize it in Fig. 3(a). We visualize the ERF of the central pixel with an input size of 1024×1024. In Fig. 3(a), "No Shift" represents the absence of the token shift method (Q-Shift), and "RWKV Attn" indicates using the original RWKV attention mechanism without our modifications for vision tasks. From the comparison in the figure, all models except the "RWKV Attn" one achieved global attention while the global capacity of the VRWKV-T model is better than that of ViT-T. Despite the assistance of Q-Shift, the central pixel in "RWKV Attn" still cannot attend to the pixels on the bottom of the image due to the large input resolution. The results of "No Shift" and Q-Shift show that the Q-Shift method expands the core range of the receptive field, enhancing the inductive bias of global attention.

Efficiency Analysis. We gradually increase the input resolution from 224×224 to 2048×2048 and compare the inference and memory efficiency of VRWKV-T and ViT-T. The results were tested on an Nvidia A100 GPU, as shown in Fig. 1. From the curves presented in Fig. 1(b), it is observed that at lower resolutions, such as 224×224 with around 200 image tokens, VRWKV-T and ViT-T exhibit comparable memory usage, though VRWKV-T has a slightly lower FPS compared to ViT-T. However, with increasing resolution, VRWKV-T's FPS rapidly exceeds that of ViT-T, thanks to its linear attention mechanism. Additionally, VRWKV-T's RNN-like computational framework ensures a slow increase in memory usage. By the time the resolution hits 2048×2048 (equivalent to 16384 tokens), VRWKV-T's inference speed is 10 times faster than ViT-T, and its memory consumption is reduced by 80% compared to ViT-T.

We also compare the speed of our Bi-WKV and flash attention [7], reported in Fig. 3(b). Flash attention is highly efficient at low resolutions yet plagued by quadratic complexity, its speed decreases rapidly as the resolution increases. In



(a) ERF of ViT and VRWKV

(b) Attention Runtime

Fig. 3: Comparison of effective receptive field (ERF) and attention runtime. (a) ERF for ViT and VRWKV in different settings. "No Shift" means no shift is used in spatial-mix and channel-mix modules. "RWKV Attn" means the original RWKV attention [37] without our modifications. Our VRWKV with Q-Shift and Bi-WKV has a more comprehensive ERF than other counterparts. (b) Attention runtime of inference (left) and forward + backward (right) tested on an Nvidia A100 GPU.

high-resolution scenarios, our linear operator Bi-WKV demonstrates a significant speed advantage. For instance, when the input is 2048×2048 (*i.e.*, 16384 tokens) with the channel and head number settings according to ViT-B and VRWKV-B, our Bi-WKV operator is $2.8\times$ faster than flash attention in inference runtime and $2.7\times$ faster in the combined forward and backward pass.

MAE Pre-training. Similar to ViT, our VRWKV model can handle sparse inputs and benefits from the MAE pre-training [20]. By simply modifying the Q-Shift to perform a bidirectional shift operation, VRWKV can be pre-trained using MAE. The pre-trained weights can be directly fine-tuned for other tasks using a Q-Shift manner. Following the same MAE pre-training setting as ViT, and subsequent classification training similar to Sec. 4.1, our VRWKV-L achieves a top-1 accuracy improvement from 85.3% to 85.5% on ImageNet-1K val, showing the ability to acquire visual prior from masked image modeling.

5 Conclusion

We propose Vision-RWKV (VRWKV), a vision encoder with a linear computational complexity attention mechanism. We demonstrate its capability to be an alternative backbone to ViT in comprehensive vision tasks including classification, dense predictions, and masked image modeling pre-training. With comparable performance and scalability, VRWKV exhibits lower computational complexity and memory consumption. Benefiting from its low complexity, VRWKV can achieve better performance in the tasks that ViT struggling to afford the high computational overhead of global attention. We hope VRWKV will be an efficient and low-cost alternative to ViT, showcasing the powerful potential of linear complexity transformers in vision fields.

References

1. Ali, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., et al.: Xcit: Cross-covariance image transformers. NeurIPS **34** (2021) 5, 10
2. Ba, J.L., Kiros, J.R., Hinton, G.E.: Layer normalization. arXiv preprint arXiv:1607.06450 (2016) 2, 6, 9
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. NeurIPS **33**, 1877–1901 (2020) 2
4. Chen, Z., Duan, Y., Wang, W., He, J., Lu, T., Dai, J., Qiao, Y.: Vision transformer adapter for dense predictions (2023) 11, 12
5. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: ICCV. pp. 764–773 (2017) 4
6. Dai, J., Shi, M., Wang, W., Wu, S., Xing, L., Wang, W., Zhu, X., Lu, L., Zhou, J., Wang, X., et al.: Demystify transformers & convolutions in modern image deep networks. arXiv preprint arXiv:2211.05781 (2022) 4, 5
7. Dao, T., Fu, D., Ermon, S., Rudra, A., Ré, C.: Flashattention: Fast and memory-efficient exact attention with io-awareness. NeurIPS **35**, 16344–16359 (2022) 13
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255 (2009) 3, 9, 12
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018) 2
10. Ding, J., Ma, S., Dong, L., Zhang, X., Huang, S., Wang, W., Zheng, N., Wei, F.: Longnet: Scaling transformers to 1,000,000,000 tokens. arXiv preprint arXiv:2307.02486 (2023) 4
11. Ding, X., Zhang, X., Han, J., Ding, G.: Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In: CVPR. pp. 11963–11975 (2022) 4, 13
12. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2020) 1
13. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2020) 3, 4, 10
14. Elman, J.L.: Finding structure in time. Cognitive Science **14**(2), 179–211 (1990) 4
15. Fan, Q., Huang, H., Chen, M., Liu, H., He, R.: Rmt: Retentive networks meet vision transformers. arXiv preprint arXiv:2309.11523 (2023) 5
16. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752 (2023) 1, 4, 5
17. Gu, A., Goel, K., Ré, C.: Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:2111.00396 (2021) 5
18. Gu, A., Johnson, I., Goel, K., Saab, K., Dao, T., Rudra, A., Ré, C.: Combining recurrent, convolutional, and continuous-time models with linear state space layers. NeurIPS **34**, 572–585 (2021) 5
19. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.: Masked autoencoders are scalable vision learners. arXiv preprint arXiv:2111.06377 (2021) 1, 3
20. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: ICCV. pp. 2961–2969 (2017) 11, 14

21. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) 3, 6
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) 10
23. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: CVPR. pp. 7132–7141 (2018) 4
24. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR. pp. 4700–4708 (2017) 3
25. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. NeurIPS **25** (2012) 3
26. LeCun, Y., Bengio, Y., et al.: Convolutional networks for images, speech, and time series. The Handbook of Brain Theory and Neural Networks **3361**(10), 1995 (1995) 4
27. Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., Zettlemoyer, L.: Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. arXiv preprint arXiv:1910.13461 (2019) 2
28. Li, X., Wang, W., Hu, X., Yang, J.: Selective kernel networks. In: CVPR. pp. 510–519 (2019) 4
29. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. pp. 740–755 (2014) 2, 3, 9, 11
30. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019) 2
31. Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., Ye, Q., Liu, Y.: Vmamba: Visual state space model. arXiv preprint arXiv:2401.10166 (2024) 4, 5
32. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV. pp. 10012–10022 (2021) 5, 10
33. Liu, Z., Mao, H., Wu, C.Y., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: CVPR. pp. 11976–11986 (2022) 4
34. Loshchilov, I., Hutter, F.: Sgdr: Stochastic gradient descent with warm restarts. arXiv preprint arXiv:1608.03983 (2016) 9
35. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017) 9, 11
36. Memory, L.S.T.: Long short-term memory. Neural Computation **9**(8), 1735–1780 (2010) 4
37. Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Cao, H., Cheng, X., Chung, M., Grella, M., GV, K.K., et al.: Rwkv: Reinventing rnns for the transformer era. arXiv preprint arXiv:2305.13048 (2023) 1, 2, 4, 5, 6, 7, 12, 13, 14
38. Radford, A., Narasimhan, K., Salimans, T., Sutskever, I.: Improving language understanding by generative pre-training (2018) 2
39. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019) 2
40. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. arXiv preprint arXiv:1910.10683 (2019) 2
41. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556 (2014) 3

42. Smith, J.T., Warrington, A., Linderman, S.W.: Simplified state space layers for sequence modeling. arXiv preprint arXiv:2208.04933 (2022) 5
43. Smith, S., Patwary, M., Norick, B., LeGresley, P., Rajbhandari, S., Casper, J., Liu, Z., Prabhumoye, S., Zerveas, G., Korthikanti, V., et al.: Using deepspeed and megatron to train megatron-turing nlg 530b, a large-scale generative language model. arXiv preprint arXiv:2201.11990 (2022) 2
44. Steiner, A., Kolesnikov, A., Zhai, X., Wightman, R., Uszkoreit, J., Beyer, L.: How to train your vit? data, augmentation, and regularization in vision transformers. arXiv preprint arXiv:2106.10270 (2021) 1, 11, 12
45. Stickland, A.C., Murray, I.: Bert and pals: Projected attention layers for efficient adaptation in multi-task learning. In: ICML. pp. 5986–5995 (2019) 2
46. Sun, Y., Dong, L., Huang, S., Ma, S., Xia, Y., Xue, J., Wang, J., Wei, F.: Retentive network: A successor to transformer for large language models. arXiv preprint arXiv:2307.08621 (2023) 4, 5
47. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions. In: CVPR. pp. 1–9 (2015) 3
48. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: ICML. pp. 10347–10357 (2021) 1, 2, 3, 9, 10, 11, 12
49. Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G., Jégou, H.: Going deeper with image transformers. In: ICCV. pp. 32–42 (2021) 2, 9
50. Vaswani, A., Ramachandran, P., Srinivas, A., Parmar, N., Hechtman, B., Shlens, J.: Scaling local self-attention for parameter efficient visual backbones. In: CVPR. pp. 12894–12904 (2021) 4, 5
51. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. NeurIPS **30** (2017) 1, 4
52. Wang, J., Zhu, W., Wang, P., Yu, X., Liu, L., Omar, M., Hamid, R.: Selective structured state-spaces for long-form video understanding. In: CVPR. pp. 6387–6397 (2023) 5
53. Wang, S., Li, B.Z., Khabsa, M., Fang, H., Ma, H.: Linformer: Self-attention with linear complexity. arXiv preprint arXiv:2006.04768 (2020) 4
54. Wang, W., Dai, J., Chen, Z., Huang, Z., Li, Z., Zhu, X., Hu, X., Lu, T., Lu, L., Li, H., et al.: Internimage: Exploring large-scale vision foundation models with deformable convolutions. In: CVPR. pp. 14408–14419 (2023) 4
55. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: ICCV. pp. 568–578 (2021) 4, 5
56. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: ICCV. pp. 568–578 (2021) 10
57. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pvtv2: Improved baselines with pyramid vision transformer. CVMJ pp. 1–10 (2022) 4, 5
58. Wu, S., Wu, T., Tan, H., Guo, G.: Pale transformer: A general vision transformer backbone with pale-shaped attention. In: AAAI. vol. 36, pp. 2731–2739 (2022) 4
59. Xiao, T., Liu, Y., Zhou, B., Jiang, Y., Sun, J.: Unified perceptual parsing for scene understanding. In: ECCV. pp. 418–434 (2018) 11, 12
60. Xiong, Y., Li, Z., Chen, Y., Wang, F., Zhu, X., Luo, J., Wang, W., Lu, T., Li, H., Qiao, Y., et al.: Efficient deformable convnets: Rethinking dynamic and sparse operator for vision applications. arXiv preprint arXiv:2401.06197 (2024) 4

61. Zhang, Y., Sun, Q., Zhou, Y., He, Z., Yin, Z., Wang, K., Sheng, L., Qiao, Y., Shao, J., Liu, Z.: Bamboo: Building mega-scale vision dataset continually with human-machine synergy. arXiv preprint arXiv:2203.07845 (2022) 10
62. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: CVPR. pp. 633–641 (2017) 9, 12
63. Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., Wang, X.: Vision mamba: Efficient visual representation learning with bidirectional state space model. arXiv preprint arXiv:2401.09417 (2024) 4, 5
64. Zhu, X., Hu, H., Lin, S., Dai, J.: Deformable convnets v2: More deformable, better results. In: CVPR. pp. 9308–9316 (2019) 4