ViG: Linear-complexity Visual Sequence Learning with Gated Linear Attention

Bencheng Liao[1, 2], **Xinggang Wang**[2, *], **Lianghui Zhu**[2], **Qian Zhang**[3], **Chang Huang**[3]

[1]Institute of Artificial Intelligence, Huazhong University of Science & Technology
[2]School of EIC, Huazhong University of Science & Technology
[3]Horizon Robotics
{bcliao, xgwang, lhzh}@hust.edu.cn, {qian01.zhang, chang.huang}@horizon.ai

Abstract

Recently, linear complexity sequence modeling networks have achieved modeling capabilities similar to Vision Transformers on a variety of computer vision tasks, while using fewer FLOPs and less memory. However, their advantage in terms of actual runtime speed is not significant. To address this issue, we introduce Gated Linear Attention (GLA) for vision, leveraging its superior hardware-awareness and efficiency. We propose direction-wise gating to capture 1D global context through bidirectional modeling and a 2D gating locality injection to adaptively inject 2D local details into 1D global context. Our hardware-aware implementation further merges forward and backward scanning into a single kernel, enhancing parallelism and reducing memory cost and latency. The proposed model, ViG, offers a favorable trade-off in accuracy, parameters, and FLOPs on ImageNet and downstream tasks, outperforming popular Transformer and CNN-based models.

Code — https://github.com/hustvl/ViG
Extended version — https://arxiv.org/abs/2405.18425

Introduction

Vision Transformer (ViT) (Dosovitskiy et al. 2020) has revolutionized computer vision by introducing an advanced sequence modeling layer Transformer (Vaswani et al. 2017) from natural language processing (NLP) to perform visual representation learning. It has proven highly successful across various vision tasks (Liu et al. 2021b; Fang et al. 2023; Radford et al. 2021; Fang et al. 2021; Song et al. 2024; Fang et al. 2024), serving as a versatile backbone. However, the quadratic complexity inherent in the Transformer's softmax attention presents significant challenges for its applications on high-resolution images. Numerous efforts (Liu et al. 2021b; Yang et al. 2021) have sought to address this limitation by drawing inspiration from the success of convolutional networks, such as constraining attention computations within local windows. While this approach achieves linear complexity similar to conventional CNNs, it falls short in capturing the global context. This raises a critical question: can we design a fundamental block that combines the best of both worlds—Transformers and CNNs—offering global receptive field and linear complexity?

The recent development of linear-time sequence modeling methods (Gu and Dao 2023; Peng et al. 2023; Yang et al. 2024b) from NLP provides a promising solution to the question. These methods operate similarly to RNNs by compressing all historical inputs into a fixed-size state and then attending to the current input based on this compressed state, unlike Transformers (Vaswani et al. 2017) which attend to all historical states. To further address the wall-time efficiency limitations of explicit recurrent forms, Mamba (Gu and Dao 2023), RWKV (Peng et al. 2023), and GLA (Yang et al. 2024b) introduce hardware-aware implementations, demonstrating superior accuracy and efficiency compared to highly-optimized Transformers. Mamba, in particular, has been widely adopted and adapted for vision tasks. While these methods achieve superior classification accuracy at a resolution of 224 and exhibit impressive efficiency in terms of FLOPs and memory at high resolutions (*e.g.*, 1248 resolution in Vision Mamba), their wall-time efficiency at lower and more common resolutions is often comparable to or even less than that of ViT/DeiT (Dosovitskiy et al. 2020; Touvron et al. 2021).

This limitation inspires us to explore the application of the superior hardware-efficient GLA (Yang et al. 2024b; Yang and Zhang 2024) to push the efficiency and accuracy envelope of linear-complexity visual sequence learning, making it more practical and competitive with the well-established and highly-optimized Transformers and CNNs. Different from Mamba based on SISO (single-input-single-output) state space model (Gu, Goel, and Ré 2022), GLA originates from linear attention (Katharopoulos et al. 2020), which has a simple and hardware-friendly matrix-multiply form by approximating softmax in standard attention with a linear kernel. To further enhance the expressiveness, GLA introduces a novel data-dependent gating mechanism to adaptively control the forget rate of compressed state.

However, the vanilla GLA model is designed for unidirectional modeling, which has temporal dependence. This makes its ability for global perception in NLP not truly global when applied to vision tasks. Despite some follow-up works (Hu et al. 2024; Huang et al. 2024) on Mamba in vision that explore additional scanning directions beyond bidirectional modeling of 1D visual sequences to approx-

*Correspondence to: Xinggang Wang

imate vanilla attention's ability to interact with visual tokens in any direction and position, these approaches suffer from significant memory inefficiencies due to frequent, non-sequential memory access patterns, making them less practical in terms of wall-time efficiency. We adhere to bidirectional modeling for its simplicity and memory-friendly access pattern. To further harness the inherent directional sensitivity in vision data (*i.e.*, information from different visual directions varies significantly in importance), we propose a bidirectional GLA (BiGLA) layer by designing a direction-wise gating mechanism to adaptively select the global context from different directions. This design shares most parameters between the forward and backward directions. Though the proposed directional design achieves global context along the 1D visual sequence, it still fails to capture the 2D nature of visual data. To address this, we propose a 2D gating locality injection to adaptively compress 2D local information extracted by convolution into the 1D global context extracted by the sequence modeling BiGLA layer. Moreover, the proposed parameter-efficient bidirectional design allows us to merge bidirectional scanning into a single kernel, enhancing the hardware-awareness of the implementation and reducing memory cost and latency introduced by the extra direction.

The main contributions of this paper can be summarized as follows:

- We present ViG, a generic vision backbone network that combines the linear complexity of Gated Linear Attention (GLA) with the hardware-awareness needed for efficient visual sequence learning. ViG addresses the limitations of previous Transformer-based and CNN-based methods, combining the best of both worlds to offer an efficient and scalable solution.
- To better adapt to vision tasks, we propose three key designs with minimal overhead: a bidirectional gated linear attention mechanism to capture the global 1D context of visual sequences, a direction-wise gating mechanism to adaptively select global context from different directions, and a 2D gating locality injection to integrate 2D local information into the 1D global context. We further provide a hardware-aware implementation that merges bidirectional scanning into a single kernel, enhancing parallelism and reducing memory cost and latency.
- Our models achieve superior performance in terms of accuracy and parameters compared to state-of-the-art non-hierarchical and hierarchical models on ImageNet, as shown in Fig. 1. For downstream dense prediction tasks (Zhou et al. 2019; Lin et al. 2014), ViG outperforms ViT and VRWKV. The wall-time efficiency of ViG outperforms the counter-part linear-complexity visual sequence learning methods (Zhu et al. 2024; Duan et al. 2024; Liu et al. 2024) and matches the well-established and highly-optimized ConvNeXt (Liu et al. 2022b) and SwinTransformer (Liu et al. 2021b).

Related Work

ViT (Dosovitskiy et al. 2020) demonstrated that visual representation learning can be performed in a sequence manner by introducing the Transformer (Vaswani et al. 2017) from NLP. Many follow-up works (Wu et al. 2021; Wang et al. 2022; Fang et al. 2022; Dong et al. 2022; Liu et al. 2022a) focus on improving ViT's efficiency and performance without altering the softmax attention. Recently, another line of works (Qin, Yang, and Zhong 2024; Katharopoulos et al. 2020; Arora et al. 2024; Dai et al. 2019) have shown that quadratic softmax attention can be replaced by advanced RNN-like, linear-time sequence modeling methods. Vision Mamba (Zhu et al. 2024) builds upon the linear-time sequence modeling Mamba block (Gu and Dao 2023) by introducing an additional backward SSM layer. VMamba (Liu et al. 2024) introduces criss-cross scanning to Mamba and builds a hierarchical architecture. LocalMamba (Huang et al. 2024) optimizes scanning directions to exploit local priors for vision. Zigma (Hu et al. 2024) and PlainMamba (Yang et al. 2024a) introduce multiple scanning directions in a zigzag manner. Many works (Ma, Li, and Wang 2024; Li et al. 2024; Chen et al. 2024a; Liang et al. 2024; Xing et al. 2024; Zhao et al. 2024; Zhang et al. 2024; Patro and Agneeswaran 2024; Shen et al. 2024; Yang, Xing, and Zhu 2024; He et al. 2024; Fei et al. 2024; Chen et al. 2024b) have explored Mamba's effectiveness in various vision tasks. In contrast, VisionRWKV (Duan et al. 2024) forgoes the Mamba block, adapting the linear complexity RWKV block from NLP for use in vision.

Preliminary

In this section, we introduce the evolution from linear attention to advanced gated linear attention (GLA), omitting the multi-head mechanism for simplicity.

Linear Attention. Linear attention (Katharopoulos et al. 2020; Qin et al. 2022) replaces $\exp(\boldsymbol{q}_t\boldsymbol{k}_i^\top)$ in standard softmax attention with feature map dot-products $\phi(\boldsymbol{q}_t)\phi(\boldsymbol{k}_i)^\top$ and removes the normalizer. This simplifies the computation of $\boldsymbol{o}_t$ as:

$$\boldsymbol{o}_t = \boldsymbol{q}_t \sum_{i=1}^{t} \boldsymbol{k}_i^\top \boldsymbol{v}_i. \tag{1}$$

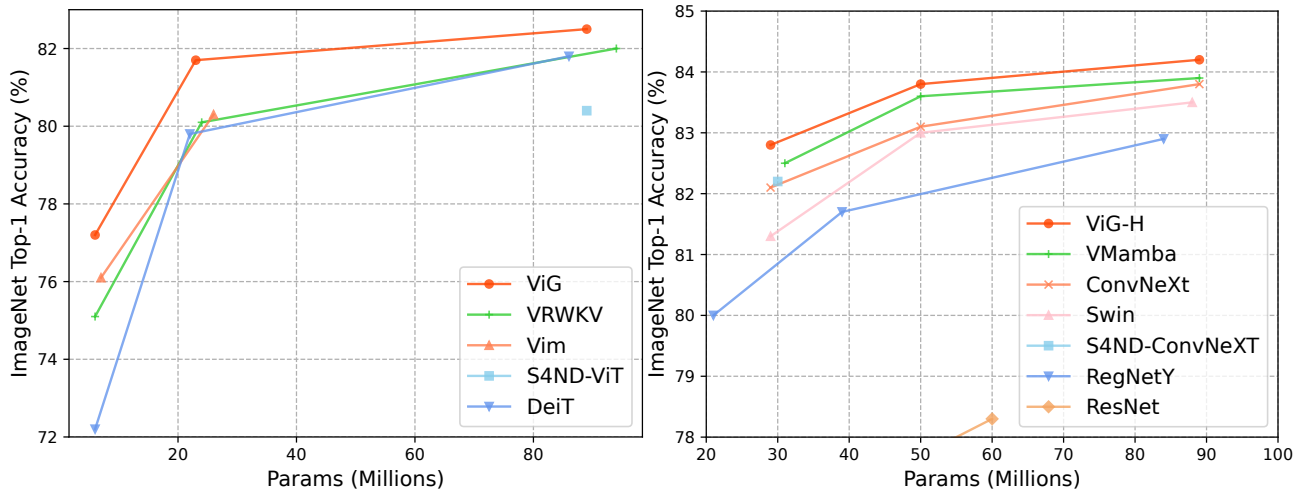
Letting hidden state $\mathbf{S}_t = \sum_{i=1}^{t} \boldsymbol{k}_i^\top \boldsymbol{v}_i \in \mathbb{R}^{d_k \times d_v}$, which is fixed-size and compresses the historical information, we can rewrite above computation as an RNN:

$$\mathbf{S}_t = \mathbf{S}_{t-1} + \boldsymbol{k}_t^\top \boldsymbol{v}_t, \quad \boldsymbol{o}_t = \boldsymbol{q}_t \mathbf{S}_t. \tag{2}$$

Gated Linear Attention. (Yang et al. 2024b) propose adding a data-dependent gating mechanism in linear attention to enhance expressiveness. The gated linear attention can be defined as:

$$\begin{aligned} \boldsymbol{\alpha}_t &= \mathrm{sigmoid}((\boldsymbol{x}_t \boldsymbol{W}_\alpha^1 \boldsymbol{W}_\alpha^2 + \boldsymbol{b}_\alpha))^{\frac{1}{\tau}} \in \mathbb{R}^{1\times d_k}, \\ \mathbf{G}_t &= \boldsymbol{\alpha}_t^\top \mathbf{1} \in (0,1)^{d_k \times d_v}, \\ \mathbf{S}_t &= \mathbf{G}_t \odot \mathbf{S}_{t-1} + \boldsymbol{k}_t^\top \boldsymbol{v}_t \in \mathbb{R}^{d_k \times d_v}, \boldsymbol{o}_t = \boldsymbol{q}_t \mathbf{S}_t, \end{aligned} \tag{3}$$

where $\boldsymbol{\alpha}_t$ is obtained from applying a low-rank linear layer on $\boldsymbol{x}_t$ followed by sigmoid activation, $\boldsymbol{W}_\alpha^1 \in \mathbb{R}^{d\times 16}$, $\boldsymbol{W}_\alpha^2 \in \mathbb{R}^{16\times d_k}$, $\boldsymbol{b}_\alpha \in \mathbb{R}^{1\times d_k}$ are trainable matrices and $\tau = 16$ is a temperature term to encourage the model to have a slower forgetting rate, $\mathbf{G}_t$ is matrix-form forget gate, expanded by outer-producting with matrice $\mathbf{1}$.



(a) Comparison with non-hierarchical architectures

(b) Comparison with hierarchical architectures

Figure 1: Performance comparisons of (a) non-hierarchical architectures (Duan et al. 2024; Touvron et al. 2021; Zhu et al. 2024; Nguyen et al. 2022) and (b) hierarchical architectures (Liu et al. 2024, 2022b, 2021b; Nguyen et al. 2022; Radosavovic et al. 2020; He et al. 2016) on ImageNet-1K. Our proposed non-hierarchical ViG and hierarchical ViG-H demonstrate superior performance compared to the popular models in terms of parameters and accuracy. Particularly, the proposed basic ViG block achieves global receptive field with linear complexity, while the CNN (He et al. 2016; Radosavovic et al. 2020; Liu et al. 2022b), vanilla softmax attention (Touvron et al. 2021) and window-attention-based (Liu et al. 2021b) blocks cannot.

Method

Overall Architecture

The overall architecture of our model is depicted in Fig. 2. We first transform the $H \times W \times 3$ image into $T = \frac{H \times W}{p^2}$ patch tokens with d dimensions, where p is the patch size. Before feeding the patch tokens into the stack of ViG blocks, we add learnable position embeddings. The output tokens of the last block are fed into a global average pooling layer followed by a linear classifier.

ViG Block ViG block, serving as a basic block, consists of: 1) a long-term BiGLA layer that can exploit the 1D-global context of the image in a linear-complexity manner; 2) a short-term depth-wise convolution layer that can capture the 2D-local details of the image; 3) a gating mechanism that can adaptively combine the global and local information; 4) a SwiGLU FFN layer for channel mixing.

Global Bidirectional Gated Linear Attention This work adopts bidirectional modeling for its inherent simplicity and memory-friendly access pattern. The crux of this design is the direction-wise gating mechanism, particularly the forget gate $\mathbf{G}_t$, which meticulously controls the flow of information. This is crucial, especially for tokens located at the boundaries of an object, where information from different directions varies significantly in importance. To harness this directional sensitivity effectively, we introduce the Bidirectional Gated Linear Attention (BiGLA) layer, as shown in Fig. 3. This layer is parameter-efficient by sharing all parameters except for the forget gate, which is tailored to each direction:

$$
\begin{aligned}
\overline{\boldsymbol{\alpha}}_t &= \mathrm{sigmoid}((\boldsymbol{x}_t \boldsymbol{W}_\alpha^1 \overline{\boldsymbol{W}}_\alpha^2 + \overline{\boldsymbol{b}}_\alpha))^{\frac{1}{\tau}} \in \mathbb{R}^{1\times 2d_k}, \\
\overrightarrow{\boldsymbol{\alpha}}_t, \overleftarrow{\boldsymbol{\alpha}}_t &= \mathrm{split}(\overline{\boldsymbol{\alpha}}_t) \in \mathbb{R}^{1\times d_k}, \\
\overrightarrow{\mathbf{G}}_t &= \overrightarrow{\boldsymbol{\alpha}}_t^\top \mathbf{1} \in (0,1)^{d_k \times d_v}, \\
\overleftarrow{\mathbf{G}}_t &= \overleftarrow{\boldsymbol{\alpha}}_t^\top \mathbf{1} \in (0,1)^{d_k \times d_v}, \\
\overrightarrow{\mathbf{S}}_t &= \overrightarrow{\mathbf{G}}_t \odot \overrightarrow{\mathbf{S}}_{t-1} + \boldsymbol{k}_t^\top \boldsymbol{v}_t \in \mathbb{R}^{d_k \times d_v}, \\
\overleftarrow{\mathbf{S}}_t &= \overleftarrow{\mathbf{G}}_t \odot \overleftarrow{\mathbf{S}}_{t+1} + \boldsymbol{k}_t^\top \boldsymbol{v}_t \in \mathbb{R}^{d_k \times d_v}, \\
\overrightarrow{\boldsymbol{o}}_t &= \boldsymbol{q}_t \overrightarrow{\mathbf{S}}_t, \overleftarrow{\boldsymbol{o}}_t = \boldsymbol{q}_t \overleftarrow{\mathbf{S}}_t, \\
\boldsymbol{o}_t &= (\overrightarrow{\boldsymbol{o}}_t + \overleftarrow{\boldsymbol{o}}_t)/2,
\end{aligned}
\tag{4}
$$

where $\overline{\square}$ indicates bidirectional modification, $\overrightarrow{\square}$ and $\overleftarrow{\square}$ indicate forward and backward direction respectively, $\overline{\boldsymbol{W}}_\alpha^2 \in \mathbb{R}^{16\times 2d_k}$ and $\overline{\boldsymbol{b}}_\alpha \in \mathbb{R}^{1\times 2d_k}$. The proposed BiGLA layer compresses the historical information of forward and backward directions into fixed-size hidden states $\overrightarrow{\mathbf{S}}_t$ and $\overleftarrow{\mathbf{S}}_t$, and attends $\boldsymbol{q}_t$ with the hidden states to obtain long-term global context in a linear-complexity manner.

The proposed design only introduces extra $17d_k$ parameters to render vanilla causal GLA layer into BiGLA layer for visual representation learning, which is minor compared to the total $4d^2$ parameters (roughly) by setting $d_q, d_k = \frac{d}{2}$ and $d_v = d$. The designed BiGLA layer has nearly the same number of parameters as the standard softmax attention while using much fewer FLOPs ($\Omega_{\text{BiGLA}} = 5Td^2 + 32Td$ *vs.* $\Omega_{\text{SoftmaxAttn}} = 4Td^2 + 2T^2d$).

2D Gating Locality Injection Though the hidden states $\overrightarrow{\mathbf{S}}_t$ and $\overleftarrow{\mathbf{S}}_t$ in the BiGLA layer, compressed along the 1D

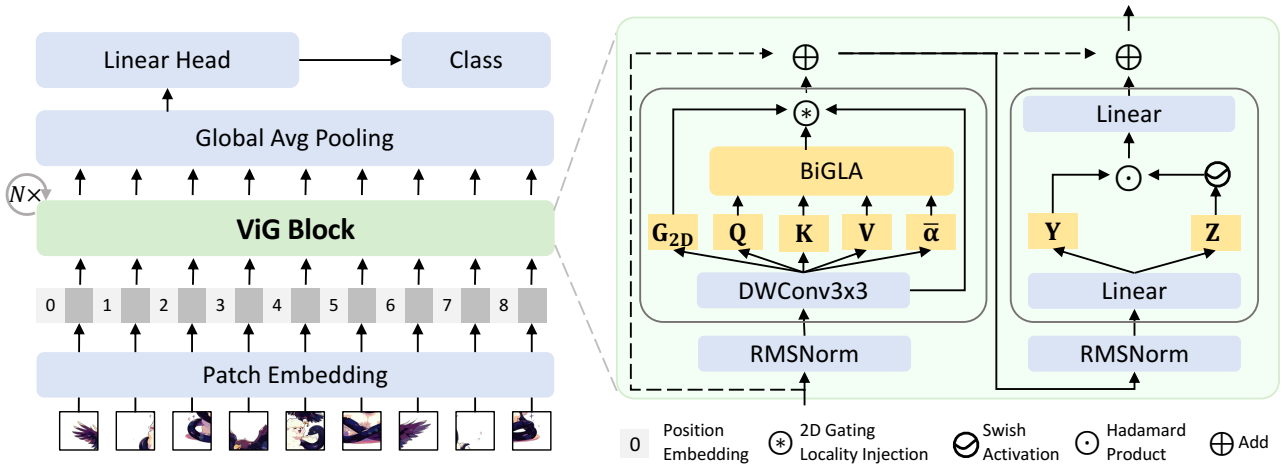


Figure 2: The overall architecture of ViG. We follow ViT to build architecture by first transforming the input image into a sequence of patches and then feeding it into N basic ViG blocks. The proposed ViG block consists of RMSNorm (Zhang and Sennrich 2019), the proposed linear complexity spatial mixing layer, and SwiGLU Feed Forward Network (Shazeer 2020).

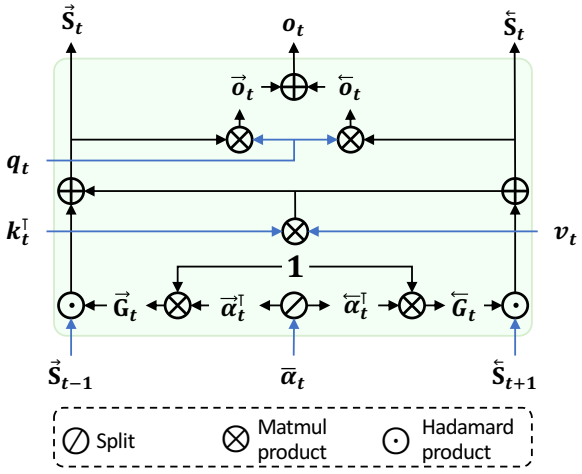


Figure 3: Illustration of BiGLA. $\overrightarrow{\mathbf{S}}_t$ and $\overleftarrow{\mathbf{S}}_t$ are the fixed-size hidden states for forward and backward directions respectively. $\overleftrightarrow{\boldsymbol{\alpha}}_t$ is the bidirectional gating.

visual sequences, can capture the long-term global context of the image, they may find it difficult in capturing the local details of 2D images. To address this issue, we inject 2D locality by introducing a short-term local convolution layer. In our case, we use 3×3 depthwise convolution for its efficiency in parameters and FLOPs, where the 3×3 convolutional filters are separated into each channel. Inspired by the data-dependent gating mechanism in GLA (Yang et al. 2024b), we propose a gating aggregation for 2D locality injection to interleave the global and local information:

$$\begin{aligned} \mathbf{O}_{\text{local}} &= \text{DWConv}_{3\times3}(X), \\ \mathbf{O}_{\text{global}} &= \text{BiGLA}(\mathbf{O}_{\text{local}}), \\ \mathbf{G}_{\text{2D}} &= \text{sigmoid}(\mathbf{O}_{\text{local}}\boldsymbol{W}_{\text{gate2D}} + \boldsymbol{b}_{\text{gate2D}}), \\ \mathbf{O} &= \mathbf{G}_{\text{2D}} \odot \mathbf{O}_{\text{local}} + (\mathbf{1} - \mathbf{G}_{\text{2D}}) \odot \mathbf{O}_{\text{global}}. \end{aligned} \tag{5}$$

Architecture Details Equipped with the proposed ViG block, we mainly investigate two kinds of variants of ViG: ViT-style non-hierarchical models with a fixed number of tokens in each block and CNN-style hierarchical models with gradually downsampled tokens.

For ViT-style models, we set the patch size p to 16 and stack 12 ViG blocks. Then, we obtain 3 variants of the model at different sizes (ViG-T, ViG-S, and ViG-B) by directly adjusting the embedding dimension d, which have similar parameters to DeiT-T, S, and B. For hierarchical models, we also propose 3 variants (ViG-H-T, ViG-H-S, and ViG-H-B) following the design of SwinTransformer. We set the patch size p to 4 and apply our proposed ViG block at different stages. The detailed architectures are provided in extended version.

Efficient Implementation

The practical efficiency of the model not only depends on the theoretical FLOPs but is mostly determined by the hardware-awareness of the implementation. It means the implemented model should: 1) be aware of memory hierarchy; 2) leverage the specialized compute unit (tensor cores on GPU for fast matrix multiplication); 3) have a high degree of parallelism. Thanks to the hardware-aware implementation of GLA, the most calculation-intensive parts of ViG can be represented in matrix multiplication and are performed on faster SRAM instead of slower high bandwidth memory (HBM), leading to superior wall-time efficiency by leveraging the tensor cores and reducing the HBM I/O cost.

Hardware-aware Bidirectional Design. Given the multi-directional nature of representing 2D images in 1D flattened sequences, the pioneering work Vim, based on bidirectional modeling, chooses to invoke two sequential kernels to process the forward and backward directions separately, which is inefficient in terms of parallelism. In this work, we propose a hardware-aware bidirectional design by fusing the

Method	Size	#P.	FLOPs	Tp.	Acc.
Transformers					
ViT-B/16 (Dosovitskiy et al. 2020)	384^2	86M	55.4G	-	77.9
ViT-L/16 (Dosovitskiy et al. 2020)	384^2	307M	190.7G	-	76.5
DeiT-T (Touvron et al. 2021)	224^2	6M	1.3G	5761	72.2
DeiT-S (Touvron et al. 2021)	224^2	22M	4.6G	2396	79.8
DeiT-B (Touvron et al. 2021)	224^2	86M	17.6G	837	81.8
MLP					
gMLP-T (Liu et al. 2021a)	224^2	6M	1.4G	3872	72.3
gMLP-S (Liu et al. 2021a)	224^2	20M	4.5G	1676	79.6
gMLP-B (Liu et al. 2021a)	224^2	73M	15.8G	647	81.6
SSMs					
S4ND-ViT-B (Nguyen et al. 2022)	224^2	89M	-	562	80.4
Vim-T (Zhu et al. 2024)	224^2	7M	1.5G	2561	76.1
Vim-S (Zhu et al. 2024)	224^2	26M	5.1G	1151	80.3
LocalVim-T (Huang et al. 2024)	224^2	8M	1.5G	885	76.2
LocalVim-S (Huang et al. 2024)	224^2	28M	4.8G	396	81.2
PlainMamba-L1 (Yang et al. 2024a)	224^2	7M	3.0G	995	77.9
PlainMamba-L2 (Yang et al. 2024a)	224^2	26M	8.1G	482	81.6
PlainMamba-L3 (Yang et al. 2024a)	224^2	51M	14.4G	279	82.3
Linear RNN					
VRWKV-T (Duan et al. 2024)	224^2	6M	1.2G	4551	75.1
VRWKV-S (Duan et al. 2024)	224^2	24M	4.6G	1724	80.1
VRWKV-B (Duan et al. 2024)	224^2	94M	18.2G	635	82.0
Linear Attention					
ViG-T	224^2	6M	0.9G	4645	77.2
ViG-S	224^2	23M	3.5G	1886	81.7
ViG-B	224^2	89M	13.8G	701	82.6

Table 1: Comparison with plain architectures on ImageNet-1K val set. "Size" means image size. "#P.", "Tp.", and "Acc." denote the number of parameters, throughput and top-1 accuracy respectively. Tp. (images/s) is measured on a single 4090 GPU with batch size 256 following (Liu et al. 2021b).

forward and backward directions of Eq. (4) into a single kernel to achieve higher parallelism. Moreover, owing to the parameter-efficient design of the BiGLA layer, we can reduce the materialization of the backward visual sequence in high-bandwidth memory (HBM), which saves memory.

Experiment

We conduct extensive experiments to validate the effectiveness of our proposed models. We present the main results on ImageNet (Deng et al. 2009). Additionally, we benchmark our model on downstream dense prediction tasks, including object detection on the COCO (Lin et al. 2014) dataset and semantic segmentation on ADE20K (Zhou et al. 2019).

Image Classification

Settings. We train classification experiments on ImageNet-1K dataset, which is a widely used large-scale benchmark for image classification. To fairly compare with previous works, we mainly follow the training and evaluation setting of DeiT and SwinTransformer (Touvron et al. 2021; Liu et al. 2021b). All the models are trained from scratch for 300 epochs. Further details are provided in extended version.

Method	Size	#P.	FLOPs	Tp.	Acc.
Convnets					
RegNetY-4G (Radosavovic et al. 2020)	224^2	12M	4G	-	80.0
RegNetY-8G (Radosavovic et al. 2020)	224^2	25M	8G	-	81.7
RegNetY-16G (Radosavovic et al. 2020)	224^2	45M	16G	-	82.9
EffNet-B3 (Tan and Le 2019)	300^2	12M	1.8G	-	81.6
EffNet-B4 (Tan and Le 2019)	380^2	19M	4.2G	-	82.9
EffNet-B5 (Tan and Le 2019)	456^2	30M	9.9G	-	83.6
EffNet-B6 (Tan and Le 2019)	528^2	43M	19.0G	-	84.0
EffNet-B7 (Tan and Le 2019)	528^2	66M	37.0G	-	84.3
ConvNeXt-T (Liu et al. 2022b)	224^2	29M	4.5G	1505	82.1
ConvNeXt-S (Liu et al. 2022b)	224^2	50M	8.7G	905	83.1
ConvNeXt-B (Liu et al. 2022b)	224^2	89M	15.4G	643	83.8
Transformers					
Swin-T (Liu et al. 2021b)	224^2	28M	4.6G	1511	81.3
Swin-S (Liu et al. 2021b)	224^2	50M	8.7G	915	83.0
Swin-B (Liu et al. 2021b)	224^2	88M	15.4G	661	83.5
SSMs					
S4ND-ConvNeXt-T (Nguyen et al. 2022)	224^2	30M	-	643	82.2
VMamba-T (Liu et al. 2024)	224^2	31M	4.9G	1161	82.5
VMamba-S (Liu et al. 2024)	224^2	50M	8.7G	779	83.6
VMamba-B (Liu et al. 2024)	224^2	89M	15.4G	557	83.9
EfficientVMamba-B (Pei, Huang, and Xu 2024)	224^2	33M	4.0G	1258	81.8
LocalVMamba-T (Huang et al. 2024)	224^2	26M	5.7G	330	82.7
LocalVMamba-S (Huang et al. 2024)	224^2	50M	11.4G	193	83.7
Linear Attention					
ViG-H-T	224^2	29M	4.5G	1480	82.8
ViG-H-S	224^2	50M	8.8G	890	83.8
ViG-H-B	224^2	89M	15.5G	621	84.2

Table 2: Comparison with hierarchical architectures on ImageNet-1K validation set.

Comparison with Non-hierarchical Architectures. Tab. 1 compares ViG with plain non-hierarchical architectures based on different sequence modeling layers, including Transformer, SSM, and linear RNN. The results show that the proposed ViG achieves superior trade-off in terms of parameters and accuracy across various model sizes, as shown in Fig. 1 (a). Remarkably, ViG-S has nearly the same number of parameters as DeiT-S and significantly outperforms it by 1.9% top-1 accuracy, which matches the performance of DeiT-B (only 0.1% lower) with 3.7× fewer parameters, 5× fewer FLOPs and 2× faster throughput. Moreover, ViG-B reaches 82.6% top-1 accuracy, surpassing DeiT-B by 0.8%, VRWKV-B by 0.6%, and S4ND-ViT-B by 2.2%.

In terms of practical throughput, ViG surpasses other linear-complexity sequence modeling methods, notably being 1.8× faster than Vim-T and 1.6× faster than Vim-S at tiny and small model sizes respectively.

Comparison with Hierarchical Architectures. Tab. 2 compares ViG-H with hierarchical architectures, including CNN-based RegNet and ConvNeXt, Transformer-based Swin Transformer, and SSM-based S4ND and VMamba. Thanks to linear complexity of the proposed VGLA block, ViG-H achieves similar FLOPs, but with the added advan-

Method	#Param.	FLOPs	AP^b	AP^m
ViT-T†	8M	95.4G	41.1	37.5
ViT-T	8M	147.1G	41.6	37.9
VRWKV-T	8M	67.9G	41.7	38.0
ViG-T	8M	**61.2G**	**43.3**	**39.1**
ViT-S†	28M	241.2G	44.6	39.7
ViT-S	28M	344.5G	44.9	40.1
VRWKV-S	29M	189.9G	44.8	40.2
ViG-S	28M	**164.1G**	**45.5**	**40.8**
ViT-B†	100M	686.7G	46.2	41.5
ViT-B	100M	893.3G	46.8	41.8
VRWKV-B	107M	599.0G	46.8	41.7
ViG-B	103M	**498.5G**	**47.3**	**42.2**

Table 3: Object detection and instance segmentation on COCO val2017. "#Param" denotes the number of backbone parameters. "FLOPs" denote the computational workload of the backbone with an input image of 1333×800. "†" means window attention is adopted in ViT layers.

tage of a global receptive field. As shown in Fig. 1 (b), ViG-H achieves the best accuracy across different model sizes.

The results of practical throughput demonstrate that ViG-H surpasses the SSM-based S4ND and VMamba, and matches the performance of well-established and highly-optimized ConvNeXt and SwinTransformer.

Object Detection

Settings. We conduct experiments for object detection and instance segmentation on the COCO 2017 dataset. We utilize Mask-RCNN as the detection head and follow VRWKV to integrate the ViT-Adapter (Chen et al. 2023) on our plain ViG models. Further details are provided in extended version.

Results. In Tab. 3, for high-resolution 1333×800 input images, ViT needs to resort to window attention to ensure efficiency, sacrificing accuracy. Unlike ViT, ViG can efficiently process high-resolution images directly with a global receptive field. The results demonstrate that ViG outperforms both ViT and VRWKV in terms of FLOPs and accuracy. Specifically, ViG-T uses only half the backbone FLOPs of ViT-T but achieves 1.7 higher AP^b and 1.2 higher AP^m, surpassing VRWKV-T by 1.6 in AP^b and 1.1 in AP^m. For the base model size, though VRWKV-B is more efficient than ViT-B in FLOPs, it still falls short of ViT-B by 0.1 in AP^m. Meanwhile, our ViG-B achieves 0.5 higher AP^b and 0.4 higher AP^m than ViT-B with 44% lower backbone FLOPs.

Semantic Segmentation

Settings. We train experiments for semantic segmentation on the ADE20K (Zhou et al. 2019) dataset. We use UperNet (Xiao et al. 2018) as the segmentation head and adopt the ViT-Adapter following VRWKV to adapt our plain ViG for segmentation. Training details are the same as VRWKV (Duan et al. 2024) and Vim (Zhu et al. 2024).

Method	#Param.	FLOPs	mIoU
Vim-T‡	-	-	41.0
ViT-T	8M	20.9G	42.6
VRWKV-T	8M	16.6G	43.3
ViG-T	8M	**14.9G**	**43.8**
Vim-S‡	-	-	44.9
ViT-S	28M	54.0G	46.2
VRWKV-S	29M	46.3G	47.2
ViG-S	28M	**40.0G**	**47.9**
ViT-B	100M	157.9G	48.8
VRWKV-B	107M	146.0G	49.2
ViG-B	103M	**121.5G**	**49.4**

Table 4: Semantic segmentation on ADE20K val set. "FLOPs" denote the computational workload of the backbone with an input image of 512×512. "‡" denotes that the results are directly taken from its paper (Zhu et al. 2024).

Method	Roadmap	#Param.	Tp.	Memory	Acc.
DeiT	-	5.72M	5761	627MB	72.2
GLA	-	5.72M	6662	582MB	66.1
+	improved patch embedding layer	5.75M	6634	582MB	73.8
+	direct bidirectional modeling	5.75M	4564	748MB	75.2
+	absolute position embedding	5.79M	4571	748MB	75.4
+	direction-wise $\overline{\alpha}$	5.81M	4566	785MB	76.3
+	2D gating locality injection	5.83M	3812	842MB	77.2
+	hardware-aware bidirection impl.	5.83M	4645	730MB	77.2

Table 5: Roadmap of ViG. Throughput and memory are test on 4090 GPU with batch size 256 and image size 224.

Results. As shown in Tab. 4, for medium-resolution 512×512 input images, ViG outperforms the quadratic-complexity Transformer-based ViT and the linear-complexity RNN-based VRWKV across different model sizes in both FLOPs and segmentation accuracy. For instance, in the tiny size models, ViG outperforms ViT by 1.2 mIoU and 5G FLOPs, and VRWKV by 0.5 mIoU and 1.7G FLOPs. In the small size models, ViG surpasses ViT by 1.7 mIoU and 14G FLOPs, and VRWKV by 0.7 mIoU and 6.3G FLOPs. These results demonstrate ViG's superiority for dense prediction tasks compared to VRWKV.

Ablation Study

Roadmap. In Tab. 5, we show the roadmap of how to introduce minimal cost to render the causal sequence modeling GLA into proposed ViG. The "improved patch embedding layer" of Row 3 means that we adopt a 9×9 convolution with stride as 8 followed by 3×3 convolution with stride as 2, which only adds 0.03 M parameters but boost the accuracy with nearly no affects on inference efficiency. The ViG-T introduces only 0.11M parameters and significantly outperforms vanilla GLA by 11.1% top-1 accuracy and DeiT by 5.0% top-1 accuracy. By further introducing the hardware-aware bidirectional implementation, we significantly boost the efficiency (enhance throughput by 21.9% and save 13.3% GPU memory) and close the gap with DeiT, even at the low-resolution 224×224 image.

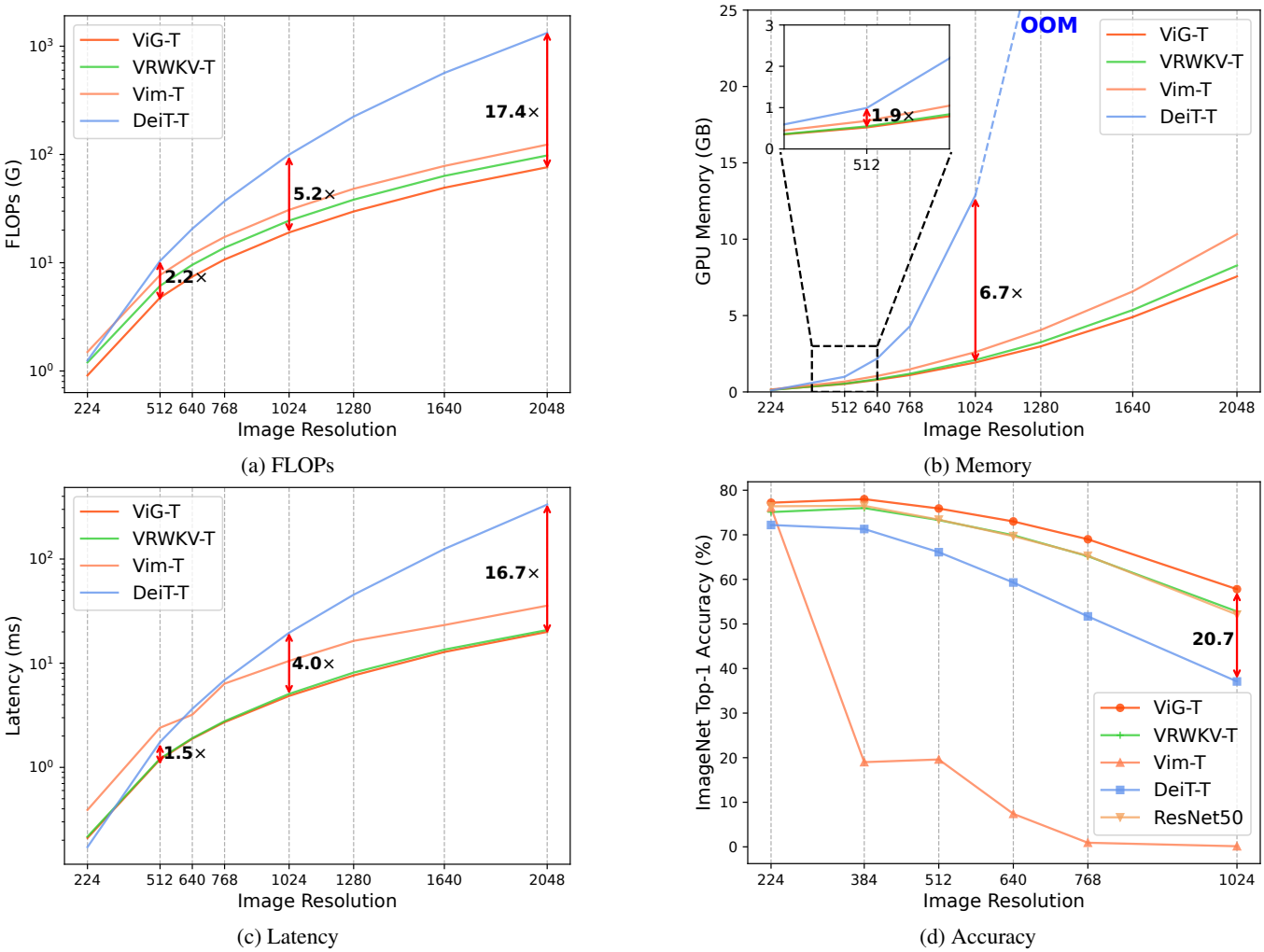


Figure 4: Comparison among ViG, Vim, VRWKV, and ViT in (a) FLOPs, (b) memory, (c) latency, and (d) accuracy with respect to increasing image resolution during inference on ImageNet-1K val set. The blue dashed line indicates the estimated values when the GPU memory has run out.

Efficiency of ViG. In Fig. 4, we compare ViG-T with Vim-T, VRWKV-T, and DeiT-T, focusing on theoretical FLOPs, actual latency, and memory usage on a 4090 GPU. We test the model across increasing input image resolutions from 224×224 to 2048×2048. Thanks to the linear-complexity sequence modeling, the advantage of the proposed architecture over ViT grows as the resolution increases. When resolution reaches 1024×1024, ViG-T uses $5.2\times$ lower FLOPs, saves 90% GPU memory, and runs $4.8\times$ faster than DeiT-T. Compared to its linear-complexity Vim, ViG also demonstrates $1.6\times$ lower FLOPs, saves 26% GPU memory, and runs $2.2\times$ faster. Additionally, ViG surpasses VRWKV in terms of FLOPs, latency, and memory.

Accuracy *vs*. Resolution. In Fig. 4 (d), we test the accuracy of models trained on 224×224 resolution across different resolutions. ViG outperforms ViT, Vim, VRWKV, and hierarchical CNN-based ResNet50. The results demonstrate that ViG benefits from better 2D-awareness and demonstrates superior generalization in resolution extrapolation.

Conclusion

In this paper, we introduced ViG, a generic vision backbone network that introduces Gated Linear Attention (GLA) to the vision field, achieving efficient visual representation learning. Our approach addresses the inherent limitations of traditional Transformers and CNNs by maintaining a global receptive field while operating with linear complexity. We propose direction-wise gating through bidirectional GLA modeling and 2D gating locality injection to capture both global context and local details, leading to significant improvements. Additionally, our hardware-aware implementation reduces the overhead brought by extra direction, enhancing efficiency. The superior results of ViG in low-resolution image classification, medium-resolution segmentation, and high-resolution detection highlight it as a competitive alternative to the existing vision backbones.

Acknowledgments

This work was partially supported by the National Science and Technology Major Project under Grant No. 2023YFF0905400 and the National Natural Science Foundation of China (NSFC) under Grant No. 62276108. We acknowledge Yuxin Fang for helpful feedback on the draft.

References

Arora, S.; Eyuboglu, S.; Zhang, M.; Timalsina, A.; Alberti, S.; Zinsley, D.; Zou, J.; Rudra, A.; and Ré, C. 2024. Simple linear attention language models balance the recall-throughput tradeoff. *arXiv preprint arXiv:2402.18668*.

Chen, G.; Huang, Y.; Xu, J.; Pei, B.; Chen, Z.; Li, Z.; Wang, J.; Li, K.; Lu, T.; and Wang, L. 2024a. Video mamba suite: State space model as a versatile alternative for video understanding. *arXiv preprint arXiv:2403.09626*.

Chen, H.; Song, J.; Han, C.; Xia, J.; and Yokoya, N. 2024b. Changemamba: Remote sensing change detection with spatio-temporal state space model. *arXiv preprint arXiv:2404.03425*.

Chen, Z.; Duan, Y.; Wang, W.; He, J.; Lu, T.; Dai, J.; and Qiao, Y. 2023. Vision transformer adapter for dense predictions. In *ICLR*.

Dai, Z.; Yang, Z.; Yang, Y.; Carbonell, J.; Le, Q. V.; and Salakhutdinov, R. 2019. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

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

Dong, X.; Bao, J.; Chen, D.; Zhang, W.; Yu, N.; Yuan, L.; Chen, D.; and Guo, B. 2022. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *CVPR*.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. In *ICLR*.

Duan, Y.; Wang, W.; Chen, Z.; Zhu, X.; Lu, L.; Lu, T.; Qiao, Y.; Li, H.; Dai, J.; and Wang, W. 2024. Vision-rwkv: Efficient and scalable visual perception with rwkv-like architectures. *arXiv preprint arXiv:2403.02308*.

Fang, J.; Xie, L.; Wang, X.; Zhang, X.; Liu, W.; and Tian, Q. 2022. Msg-transformer: Exchanging local spatial information by manipulating messenger tokens. In *CVPR*.

Fang, Y.; Liao, B.; Wang, X.; Fang, J.; Qi, J.; Wu, R.; Niu, J.; and Liu, W. 2021. You only look at one sequence: Rethinking transformer in vision through object detection. In *NeurIPS*.

Fang, Y.; Sun, Q.; Wang, X.; Huang, T.; Wang, X.; and Cao, Y. 2024. Eva-02: A visual representation for neon genesis. *Image and Vision Computing*, 149: 105171.

Fang, Y.; Wang, W.; Xie, B.; Sun, Q.; Wu, L.; Wang, X.; Huang, T.; Wang, X.; and Cao, Y. 2023. Eva: Exploring the limits of masked visual representation learning at scale. In *CVPR*.

Fei, Z.; Fan, M.; Yu, C.; and Huang, J. 2024. Scalable Diffusion Models with State Space Backbone. *arXiv preprint arXiv:2402.05608*.

Gu, A.; and Dao, T. 2023. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*.

Gu, A.; Goel, K.; and Ré, C. 2022. Efficiently modeling long sequences with structured state spaces. In *ICLR*.

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

He, X.; Cao, K.; Yan, K.; Li, R.; Xie, C.; Zhang, J.; and Zhou, M. 2024. Pan-Mamba: Effective pan-sharpening with State Space Model. *arXiv preprint arXiv:2402.12192*.

Hu, V. T.; Baumann, S. A.; Gui, M.; Grebenkova, O.; Ma, P.; Fischer, J.; and Ommer, B. 2024. Zigma: Zigzag mamba diffusion model. *arXiv preprint arXiv:2403.13802*.

Huang, T.; Pei, X.; You, S.; Wang, F.; Qian, C.; and Xu, C. 2024. Localmamba: Visual state space model with windowed selective scan. *arXiv preprint arXiv:2403.09338*.

Katharopoulos, A.; Vyas, A.; Pappas, N.; and Fleuret, F. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *ICML*.

Li, K.; Li, X.; Wang, Y.; He, Y.; Wang, Y.; Wang, L.; and Qiao, Y. 2024. Videomamba: State space model for efficient video understanding. *arXiv preprint arXiv:2403.06977*.

Liang, D.; Zhou, X.; Wang, X.; Zhu, X.; Xu, W.; Zou, Z.; Ye, X.; and Bai, X. 2024. PointMamba: A Simple State Space Model for Point Cloud Analysis. *arXiv preprint arXiv:2402.10739*.

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

Liu, H.; Dai, Z.; So, D.; and Le, Q. V. 2021a. Pay attention to mlps. In *NeurIPS*.

Liu, Y.; Tian, Y.; Zhao, Y.; Yu, H.; Xie, L.; Wang, Y.; Ye, Q.; and Liu, Y. 2024. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*.

Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; et al. 2022a. Swin transformer v2: Scaling up capacity and resolution. In *CVPR*.

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

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

Ma, J.; Li, F.; and Wang, B. 2024. U-mamba: Enhancing long-range dependency for biomedical image segmentation. *arXiv preprint arXiv:2401.04722*.

Nguyen, E.; Goel, K.; Gu, A.; Downs, G. W.; Shah, P.; Dao, T.; Baccus, S. A.; and Ré, C. 2022. S4nd: Modeling images and videos as multidimensional signals using state spaces. *arXiv preprint arXiv:2210.06583*.

Patro, B. N.; and Agneeswaran, V. S. 2024. Simba: Simplified mamba-based architecture for vision and multivariate time series. *arXiv preprint arXiv:2403.15360*.

Pei, X.; Huang, T.; and Xu, C. 2024. Efficientvmamba: Atrous selective scan for light weight visual mamba. *arXiv preprint arXiv:2403.09977*.

Peng, B.; Alcaide, E.; Anthony, Q.; Albalak, A.; Arcadinho, S.; Cao, H.; Cheng, X.; Chung, M.; Grella, M.; GV, K. K.; et al. 2023. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*.

Qin, Z.; Han, X.; Sun, W.; Li, D.; Kong, L.; Barnes, N.; and Zhong, Y. 2022. The devil in linear transformer. *arXiv preprint arXiv:2210.10340*.

Qin, Z.; Yang, S.; and Zhong, Y. 2024. Hierarchically gated recurrent neural network for sequence modeling. In *NeurIPS*.

Radford, A.; Kim, J. W.; Hallacy, C.; Ramesh, A.; Goh, G.; Agarwal, S.; Sastry, G.; Askell, A.; Mishkin, P.; Clark, J.; et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

Radosavovic, I.; Kosaraju, R. P.; Girshick, R.; He, K.; and Dollár, P. 2020. Designing network design spaces. In *CVPR*.

Shazeer, N. 2020. Glu variants improve transformer. *arXiv preprint arXiv:2002.05202*.

Shen, Q.; Yi, X.; Wu, Z.; Zhou, P.; Zhang, H.; Yan, S.; and Wang, X. 2024. Gamba: Marry gaussian splatting with mamba for single view 3d reconstruction. *arXiv preprint arXiv:2403.18795*.

Song, Y.; Wang, X.; Yao, J.; Liu, W.; Zhang, J.; and Xu, X. 2024. ViTGaze: Gaze Following with Interaction Features in Vision Transformers. *Visual Intelligence*.

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

Touvron, H.; Cord, M.; Douze, M.; Massa, F.; Sablayrolles, A.; and Jégou, H. 2021. Training data-efficient image transformers & distillation through attention. In *ICML*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NeurIPS*.

Wang, W.; Xie, E.; Li, X.; Fan, D.-P.; Song, K.; Liang, D.; Lu, T.; Luo, P.; and Shao, L. 2022. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*.

Wu, H.; Xiao, B.; Codella, N.; Liu, M.; Dai, X.; Yuan, L.; and Zhang, L. 2021. Cvt: Introducing convolutions to vision transformers. In *ICCV*.

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

Xing, Z.; Ye, T.; Yang, Y.; Liu, G.; and Zhu, L. 2024. Segmamba: Long-range sequential modeling mamba for 3d medical image segmentation. *arXiv preprint arXiv:2401.13560*.

Yang, C.; Chen, Z.; Espinosa, M.; Ericsson, L.; Wang, Z.; Liu, J.; and Crowley, E. J. 2024a. Plainmamba: Improving non-hierarchical mamba in visual recognition. *arXiv preprint arXiv:2403.17695*.

Yang, J.; Li, C.; Zhang, P.; Dai, X.; Xiao, B.; Yuan, L.; and Gao, J. 2021. Focal self-attention for local-global interactions in vision transformers. *arXiv preprint arXiv:2107.00641*.

Yang, S.; Wang, B.; Shen, Y.; Panda, R.; and Kim, Y. 2024b. Gated linear attention transformers with hardware-efficient training. In *ICML*.

Yang, S.; and Zhang, Y. 2024. FLA: A Triton-Based Library for Hardware-Efficient Implementations of Linear Attention Mechanism. https://github.com/sustcsonglin/flash-linear-attention.

Yang, Y.; Xing, Z.; and Zhu, L. 2024. Vivim: a video vision mamba for medical video object segmentation. *arXiv preprint arXiv:2401.14168*.

Zhang, B.; and Sennrich, R. 2019. Root mean square layer normalization. In *NeurIPS*.

Zhang, Z.; Liu, A.; Reid, I.; Hartley, R.; Zhuang, B.; and Tang, H. 2024. Motion mamba: Efficient and long sequence motion generation with hierarchical and bidirectional selective ssm. *arXiv preprint arXiv:2403.07487*.

Zhao, S.; Chen, H.; Zhang, X.; Xiao, P.; Bai, L.; and Ouyang, W. 2024. Rs-mamba for large remote sensing image dense prediction. *arXiv preprint arXiv:2404.02668*.

Zhou, B.; Zhao, H.; Puig, X.; Xiao, T.; Fidler, S.; Barriuso, A.; and Torralba, A. 2019. Semantic understanding of scenes through the ade20k dataset. *IJCV*.

Zhu, L.; Liao, B.; Zhang, Q.; Wang, X.; Liu, W.; and Wang, X. 2024. Vision mamba: Efficient visual representation learning with bidirectional state space model. In *ICML*.