Star with Bilinear Mapping

Zelin Peng[1,†], Yu Huang[1,†], Zhengqin Xu[1], Feilong Tang[2,3], Ming Hu[3],
Xiaokang Yang[1], and Wei Shen[1(✉)]
[1]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiaotong University
[2]MBZUAI, [3]Monash University.
{zelin.peng, yellowfish, fate311, changsong, xkyang, wei.shen}@sjtu.edu.cn
{Feilong.Tang, Ming.Hu}@monash.edu

Abstract

Contextual modeling is crucial for robust visual representation learning, especially in computer vision. Although Transformers have become a leading architecture for vision tasks due to their attention mechanism, the quadratic complexity of full attention operations presents substantial computational challenges. To address this, we introduce Star with Bilinear Mapping (SBM), a Transformer-like architecture that achieves global contextual modeling with linear complexity. SBM employs a bilinear mapping module (BM) with low-rank decomposition strategy and star operations (element-wise multiplication) to efficiently capture global contextual information. Our model demonstrates competitive performance on image classification and semantic segmentation tasks, delivering significant computational efficiency gains compared to traditional attention-based models. Code is available at https://github.com/SJTU-DeepVisionLab/SBM.

1. Introduction

Contextual modeling capabilities are fundamental for learning robust visual representations, serving as a cornerstone for advancements in the field of computer vision. Compared to classical CNNs, which have limited capacity for contextual understanding [18, 21, 25, 45, 63], Transformers [10, 11, 32, 33, 39, 47, 52] leverage a multi-head attention mechanism [50] to enable global interactions among tokens. This capability positions Transformers as a critical backbone for numerous foundational models [2, 11, 42, 43, 48], thus driving the ongoing AI revolution.

However, as shown in Fig.1, a critical challenge in deploying Transformers in these advancements is the quadratic complexity between matrix multiplications in the full attention mechanism [50], which significantly increases the computation cost as the token length grows. To address this issue, a promising direction is to achieve global contextual modeling with linear complexity, as demonstrated by the popular Mamba [12] and its variants [8, 31, 58, 60]. Despite their competitive performance, recent studies [14, 61] indicate that the core state space model may be somewhat redundant for certain vision tasks and still limits the training and inference efficiency unless equipped with a well-designed CUDA acceleration algorithm.

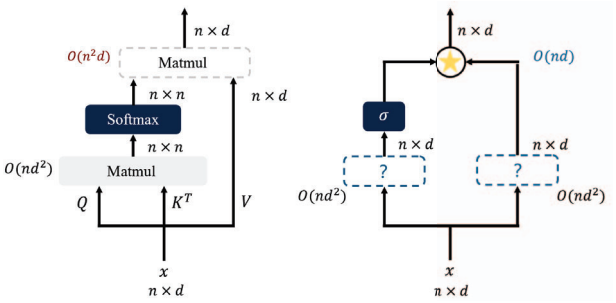

Figure 1. **Left:** Full attention architecture. **Right:** Architecture with star operation. The full attention mechanism computes a matrix multiplication between the query-key similarity matrix and the value matrix to capture global contextual information, resulting in a computational complexity of $O(n^2d)$. Our objective is to leverage the star operation, as shown on the right, to achieve efficient global receptive field coverage with linear complexity, thereby unlocking its potential for scalable contextual modeling.

In this paper, we aim to explore an alternative approach that not only achieves global contextual modeling with linear complexity but also facilitates fast training and inference. Existing research has theoretically demonstrated [37, 44, 59] that element-wise multiplication (i.e., star operation), as a linear complexity operation, can effectively capture high-dimensional contextual information when stacked across

✉ Corresponding Author: wei.shen@sjtu.edu.cn
† Indicates equal contribution.

multiple layers. Inspired by this, we hypothesize that there might be an efficient way to capture global contextual information by utilizing a series of star operations as a core module in constructing Transformer-like architectures.

Based on this motivation, we present the **S**tar with **B**linear **M**apping (SBM), a novel architecture that takes a step towards this goal by adopting an efficient bilinear mapping operation. Bilinear mapping [26] is a technique originally introduced to transform a global interaction, typically involving quadratic complexity, into multiplicative global interactions with linear complexity. We introduce the bilinear mapping module (BM) to capture complex contextual information in a feature space while maintaining linear complexity. In particular, we employ a low-rank matrix decomposition strategy for constructing the BM weights, ensuring that the complexity remains highly efficient. In each SBM, we use star operation to facilitate element-wise interactions between different feature spaces, allowing the model to learn complex relationships between features from the element level. Consequently, by stacking several SBM blocks, we are able to progressively enhance the expressive power of the star operation for general vision recognition tasks.

We validate the proposed SBM model on three major tasks: image classification using ImageNet-1K [25] and transer learning on CIFAR-100 [24], semantic segmentation on the ADE20K dataset [65], and object detection on the COCO 2017 dataset [30]. These evaluations illustrate that our approach successfully balances computational efficiency with strong performance in contextual modeling. Specifically, SBM achieves competitive accuracy across both classification and segmentation tasks, while substantially reducing computational overhead compared to traditional attention-based models. Our contributions are summarized as follows:

- We introduce **Star with Bilinear Mapping (SBM)**, a novel Transformer-like architecture that combines star operations with bilinear mapping to achieve global contextual modeling with linear computational complexity. This design overcomes the quadratic complexity limitations inherent in traditional attention mechanisms.
- By incorporating low-rank decomposition strategy into the BM module, we enhance the global receptive field and contextual modeling capabilities, allowing the model to capture complex feature relationships at the element level while maintaining high computational efficiency.
- We demonstrate the effectiveness of the SBM through extensive evaluations on image classification (ImageNet-1K [25] and CIFAR-100 [24]), semantic segmentation (ADE20K [65]) and object detection (COCO 2017 [30]). achieving competitive performance with significantly lower computational demands than recent models.

2. Related Work

Convolutional Neural Networks. Convolutional Neural Networks (CNNs) are a well-established architecture, renowned for their strong inductive bias towards capturing local features and ensuring translation invariance [18, 21, 25, 45]. This bias enables CNNs to effectively model spatial hierarchies and detect patterns at various scales. However, despite their effectiveness in local feature extraction, CNNs struggle with capturing long-range dependencies and global contextual information, which are often crucial for understanding complex relationships in data [3, 35, 49]. These limitations arise from the inherent locality of convolutional operations, which operate on fixed-sized regions and thus fail to model interactions between distant spatial locations. To address these challenges, Transformer-based models have emerged, offering significant advantages in modeling long-range dependencies. By leveraging self-attention mechanisms, Transformers are able to dynamically capture relationships between all positions in the input, regardless of their spatial distance, making them better suited for tasks that require a global understanding of the data.

Vision Transformers and Mamba. Since the introduction of ViT [11], Transformers and attention mechanisms have become widely adopted in computer vision tasks [10, 15, 32, 33, 46, 47, 51, 52, 54]. However, the quadratic complexity of the full attention mechanism poses significant challenges for scaling Transformers to high-resolution images. To address this, various approaches have been proposed to reduce the computational burden. One approach involves linear attention [23], which reduces complexity to linear time but often leads to performance degradation. Another line of research, inspired by the success of state space models in sequence modeling [13], has attempted to adapt the Mamba model [12] to vision tasks [22, 31, 38, 66] seeking to preserve the effectiveness of attention mechanisms while reducing computational costs. However, several studies have highlighted conceptual similarities between linear attention and the Mamba model [8, 14]. These studies reveal that Mamba suffers from memory inefficiency and lower throughput compared to transformers with linear attention, suggesting that the core state space model in Mamba may contain redundant operations or components for vision tasks. In contrast to previous studies, this paper introduces a novel architecture that incorporates a simple element-wise multiplication, i.e., the star operation, as a core module to achieve global contextual modeling with linear complexity.

3. Preliminaries

In this section, we briefly introduce two key concepts, i.e., star operation and bilinear mapping, that form the foundation of the novel architecture presented in this paper.

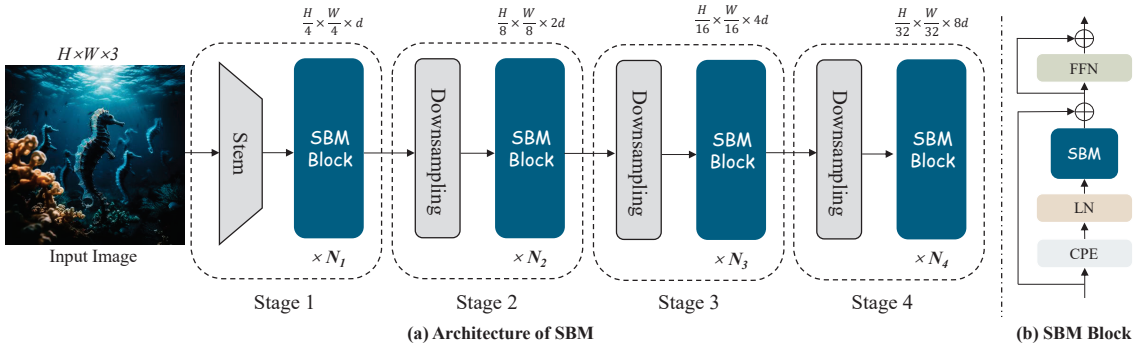

Figure 2. (a) Overview of the SBM model architecture. (b) Structure of the SBM Block. CPE stands for Conditional Positional Encoding, FFN represents the Feed Forward Network, and SBM denotes the SBM module, with detailed information provided in Fig. 3.

3.1. Star Operation

Element-wise multiplication refers to a mathematical operation where two matrices or vectors of the same size are multiplied together by multiplying their corresponding elements. In [37], a very recent study on network architecture design, element-wise multiplication is referred to as the star operation. The authors claim that the star operation enables exponential growth in feature dimensionality by fusing two linearly transformed features. More concretely, we can define this operation as $\left(\mathbf{W}_1^\top \mathbf{x}\right) * \left(\mathbf{W}_2^\top \mathbf{x}\right)$, where $\mathbf{W}_1$ and $\mathbf{W}_2$ are weight matrices, and $*$ denotes star operation. Assume that $\mathbf{w}_1, \mathbf{w}_2, \mathbf{x} \in \mathbb{R}^{(d+1)\times 1}$, with d representing the dimension of an input and the additional dimension represents the bias term. Then, the star operation expands as:

$$\begin{aligned}\mathbf{w}_1^\top \mathbf{x} * \mathbf{w}_2^\top \mathbf{x} &= \left(\sum_{i=1}^{d+1} w_1^i x^i\right) * \left(\sum_{j=1}^{d+1} w_2^j x^j\right) \\ &= \sum_{i=1}^{d+1}\sum_{j=1}^{d+1} w_1^i w_2^j x^i x^j,\end{aligned}$$

where this operation yields $\frac{(d+2)(d+1)}{2} \approx (\frac{d}{\sqrt{2}})^2$ unique terms ($d \gg 2$), creating a high-dimensional feature space. By stacking multiple star operations in different layers, star operations can substantially amplify the implicit dimensions exponentially. For a network with l layers and width d, the output feature space scales to $\mathbb{R}^{\left(\frac{d}{\sqrt{2}}\right)^{2^l}}$, greatly demonstrates its contextual modeling capability. This ability makes it a powerful tool for learning complex feature interactions efficiently. When combined with appropriate nonlinear activation functions and a well-designed network architecture, the star operation can significantly enhance the capacity of vision models to capture intricate relationships in data.

However, a notable limitation of the star operation is its restricted receptive field in each layer. While star operations can theoretically maintain nonlinearity and high-dimensional representation, similar to matrix multiplication, their element-wise nature prevents them from providing global interactions across the input. This lack of a global receptive field poses a significant challenge, as capturing long-range dependencies is crucial for many vision tasks. Therefore, the key challenge is how to augment the star operation to enable efficient global receptive field coverage, addressing this limitation and unlocking its full potential.

3.2. Bilinear Mapping

Bilinear mapping [26, 40, 41] refers to a mathematical operation that takes two input vectors (or tensors) and produces an output that is linear in both inputs. Specifically, a bilinear mapping is often realized in a matrix form to facilitate efficient computation. Given two input vectors $\mathbf{x} \in \mathbb{R}^n$ and $\mathbf{y} \in \mathbb{R}^m$, a bilinear mapping $B(\mathbf{x}, \mathbf{y})$ can be expressed using a matrix $\mathbf{A} \in \mathbb{R}^{n\times m}$ as follows:

$$B(\mathbf{x}, \mathbf{y}) = \mathbf{x}^\top \mathbf{A} \mathbf{y}, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{y} \in \mathbb{R}^m$, and $\mathbf{A} \in \mathbb{R}^{n\times m}$ is a matrix defining the pairwise interactions between the matrices of $\mathbf{x}$ and $\mathbf{y}$. Each element A_{ij} represents the weight or interaction strength between the i-th component of $\mathbf{x}$ and the j-th component of $\mathbf{y}$. Through the transformation matrix $\mathbf{A}$, a bilinear mapping facilitates the interaction between two vectors, i.e., $\mathbf{x}$ and $\mathbf{y}$, while maintaining linear computational complexity.

4. Methodology

In this section, we provide a comprehensive description of the design principles and computational strategies behind our proposed model. We begin by outlining the overall architecture in Sec. 4.1, detailing its hierarchical structure and the sequential processing of input tokens. In Sec. 4.2, we introduce the core Star with Bilinear Mapping (SBM) module, which integrates bilinear mapping with a low-rank decomposition strategy (Sec. 4.2.1) to efficiently model global

context. This is followed by a discussion of the mapping control mechanism in Sec. 4.2.3, which enhances model stability and spatial coherence through a multiplicative control branch. Lastly, we present the architecture variants in Sec. 4.3, which include configurations of SBM-T, SBM-S, and SBM-B designed to accommodate different computational budgets while maintaining high performance.

4.1. Overall Architecture

As depicted in Fig. 2 (a), our model adopts a four-stage processing pipeline, which is similar to the architecture of the Swin Transformer [32]. The process begins with the raw input image being processed in stage 1. In this stage, the image is first passed through the Stem block, where it is divided into non-overlapping patches, each with a spatial resolution of $\frac{H}{4} \times \frac{W}{4}$ and a deeper channel dimension of d. These patch tokens then pass through a series of SBM blocks, which preserve the resolution of the tokens while enriching their feature representations.

In the subsequent stages, patch tokens are subjected to downsampling operations, which merge patches as the model depth increases. These downsampling layers reduce the number of tokens by a factor of 2 while simultaneously doubling the channel depth, enabling the model to construct a hierarchical representation. The specific resolutions of the feature maps at each stage are annotated in Fig. 2 (a), illustrating the structured transition through the pipeline.

SBM Block. In each SBM block stage, patch tokens first pass through a conditional positional encoding [5] layer and layer normalization [1] to enhance relative positional information and ensure training stability. Following this, the tokens are processed by the Star with Bilinear Mapping (SBM) module, which serves as the core operation, providing global contextual modeling capabilities. Finally, the tokens are passed through a feed-forward network (FFN), consisting of a 2-layer MLP with `SiLU` nonlinearity. Shortcut connections are integrated into each SBM block to support stable and efficient training.

4.2. Star with Bilinear Mapping

Given input patch tokens of shape (n, d), where n represents the number of tokens and d is the embedding dimension, the Star with Bilinear Mapping (SBM) module, illustrated in Fig. 3, is formally defined by the following equations:

$$
\begin{aligned}
f(\mathbf{x}) &= \sigma(\mathbf{U}) * \mathbf{V}, \\
\mathbf{U} = \mathbf{W}_s \mathbf{x}_{\text{proj}} {\mathbf{W}_c}^{\top},\ \mathbf{V} &= \mathbf{x}_{\text{proj}} \mathbf{W}_v^{\top}, \\
\mathbf{x}_{\text{proj}} &= \texttt{Conv}(\mathbf{x} {\mathbf{W}_i}^{\top}), \\
G(\mathbf{x}) &= \sigma(\mathbf{x} \mathbf{W}_G^{\top}), \\
h(\mathbf{x}) &= G(\mathbf{x}) * f(\mathbf{x}).
\end{aligned}
\tag{2}
$$

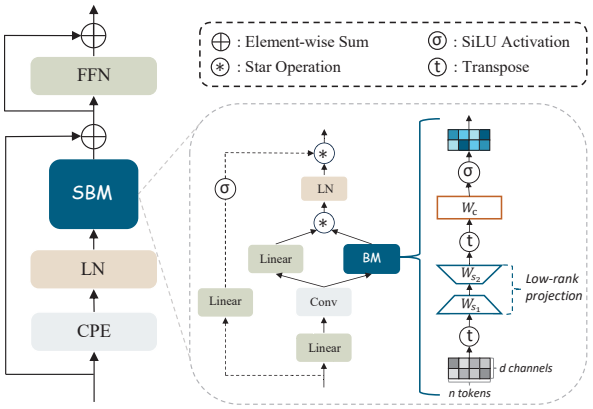

Figure 3. Details of the SBM module. BM refers to bilinear mapping combined with low-rank decomposition strategy. W_{s_1} and W_{s_2} represent the low-rank spatial projection matrices, and W_c is the channel projection matrix.

In the formulation above, $f(\mathbf{x})$ denotes the Star with Bilinear Mapping (SBM) operation, which computes the star operation between the bilinear mapping matrix $\mathbf{U}$ and the linearly projected matrix $\mathbf{V}$. Both $\mathbf{U}$ and $\mathbf{V}$ are derived from $\mathbf{x}_{\text{proj}}$, obtained by applying a linear projection $\mathbf{W}_i$ to $\mathbf{x}$, followed by a convolutional transformation. $G(\mathbf{x})$ represents the mapping control in the SBM module, which is derived from the original patch tokens $\mathbf{x}$ through $\mathbf{W}_G$ and an activation function. The symbol σ refers to the `SiLU` activation function, applied to $\mathbf{U}$ and $G(\mathbf{x})$ to introduce non-linear dynamics. The operator $*$ represents the star operation, which enables element-wise interactions between features, producing the final output $h(\mathbf{x})$ of the SBM module. Consequently, the SBM module consists of three branches: a bilinear mapping branch, a linear mapping branch and a mapping control branch. Each of these branches contributes to the overall functionality and performance of the module.

4.2.1 Bilinear Mapping Branch

In the SBM module, the matrix $\mathbf{U}$, which serves as the primary branch of the SBM operation, is derived through a bilinear mapping (BM) operation on $\mathbf{x}_{\text{proj}}$, as depicted in Fig. 3. This process involves applying linear projections across both the spatial and channel dimensions, utilizing $\mathbf{W}_s$ and $\mathbf{W}_c$, respectively. Although a standard linear projection $\mathbf{W}_s$ is capable of learning global contextual information, it is time-consuming and results in substantial memory overhead. For fast matrix multiplication, following [20], we approximate the $\mathbf{W}_s$ by using a low-rank decomposition strategy. By controlling the dimensions of the low-rank projection, we can efficiently apply the BM to a range of vision tasks.

Low-rank Decomposition. To achieve this, the spatial projection matrix $\mathbf{W}_s$ is decomposed into two components: one with dimensions $n \times m$ and the other with dimensions $m \times n$, where $m \ll n$. This technique compresses the token em-

beddings into a more compact latent space to obtain more informative representations, which are then projected back to match the original token dimensions, thereby enhancing the understanding of the hierarchical spatial structure. Following this, the feature map undergoes a $d \times d$ channel-wise projection via $\mathbf{W}_c$, which combines global contextual information with channel-specific details to further facilitate feature interactions. Finally, the resulting feature map is passed through an activation function, completing the bilinear mapping operation.

4.2.2 Linear Mapping Branch

The goal of the linear mapping branch is to provide a different feature view for proceeding star operation. Specifically, the matrix $\mathbf{V}$ in the this branch is generated by applying the channel-wise projection $\mathbf{W}_v$ to $\mathbf{x}_{\text{proj}}$. This operation produces a feature matrix that preserves the complete spatial information and is able to capture the channel-wise contextual information. The resulting matrix $\mathbf{V}$ plays a crucial role in maintaining the representation of each token while facilitating further processing within the SBM framework.

To allow the model to learn complex relationships between features from the element level, the outputs $\mathbf{U}$ and $\mathbf{V}$ are then combined using the star operation, effectively aggregating information from both branches. This is followed by a layer normalization to ensure model stability.

4.2.3 Mapping Control Branch

To preserve the original spatial structures and ensure more stable training, we introduce a shortcut branch $G(\mathbf{x})$, i.e., the mapping control branch, which interacts with the main branch through a star operation. The final output feature map $h(\mathbf{x})$ of the SBM module is shown in Eq. 2 of equations in the last two rows, where $G(\mathbf{x})$ is derived from the former layer and passed through an activation function. Although the mapping control branch introduces a slight reduction in throughput, we observe that it plays a crucial role in enhancing performance. We carefully balance this trade-off, and the experimental details are provided in Sec. 5.

4.2.4 Computational Complexity

The computational complexity of the SBM module can be analyzed as follows:

$$\Omega(\text{SBM}) = 5nd^2 + 2ndm + 2nd, \tag{3}$$

where m represents the dimension of the latent space in the low-rank projection, and is a hyperparameter that depends on the dataset. In this work, we set m to be of the same order of magnitude as the embedding dimension d.

Specifically, the Bilinear Mapping (BM) operation contributes a complexity of $2ndm + nd^2$, due to the linear projections applied across both the spatial and channel dimensions. The term $2nd$ arises from the star operation, which is computationally efficient compared to traditional matrix multiplication. Additionally, the channel-wise projections contribute a complexity of $O(nd^2)$ in the SBM module.

In total, the SBM module exhibits linear time complexity, which is a significant improvement over the full attention mechanism, which has a complexity of $4nd^2 + 2n^2d$.

4.3. Architecture Variants

We design our base model, SBM-B, to achieve comparable parameter efficiency and computational cost to Swin-B [32] and Vmamba-B [31]. Additionally, we introduce smaller versions of the model, namely SBM-T and SBM-S, to cater to different computational requirements. All versions of the model use a latent space dimension $m = 64$. The main configurations of our models are as follows:

- SBM-T: $d = 64$, number of layers [3, 3, 8, 3]
- SBM-S: $d = 64$, number of layers [3, 8, 20, 4]
- SBM-B: $d = 96$, number of layers [3, 8, 20, 4]

where d refers to the embedding dimension in the first stage of each model. Further details on the model size, throughput, FLOPs, and performance are provided in Sec. 5.

5. Experiments

To comprehensively evaluate SBM, we conduct experiments across multiple datasets: ImageNet-1K [25] and CIFAR-100 [24] for image classification, ADE20K [65] for semantic segmentation, and COCO 2017 [30] for object detection. We systematically compare SBM against state-of-the-art models, assessing classification performance in Sec.5.1 and Sec.5.2, segmentation in Sec.5.3, and object detection in Sec.5.4. Finally, we conduct detailed ablation studies in Sec. 5.5 to analyze the contribution of each component.

5.1. Image Classification

We evaluate SBM on image classification using the ImageNet-1K dataset [25], which consists of 1.28 million training images and 50,000 validation images across 1,000 categories. Following the Swin Transformer [32] training pipeline, we train SBM for 300 epochs with the AdamW [36] optimizer, using a cosine decay scheduler for learning rate adjustment and a 30-epoch warm-up phase to enhance model stability. All experiments are conducted on 8 NVIDIA A800 GPUs, leveraging widely adopted data augmentations to improve generalization.

The results in Table 1 demonstrate SBM's superiority over existing methods. With similar computational cost (FLOPs), SBM-T achieves a top-1 accuracy of 83.2

Method	Type	Params	FLOPs	Top-1
ConvNeXt-T [34]	CNN	29M	4.5G	82.1
MambaOut-T [61]	CNN	27M	4.5G	82.7
EffNet-B4[]	CNN	19M	4.2G	82.9
Swin-T [32]	Transformer	29M	4.5G	81.3
PVTv2-B2 [53]	Transformer	25M	4.0G	82.0
Focal-T [59]	Transformer	29M	4.9G	82.2
MViTv2-T [28]	Transformer	24M	4.7G	82.3
HiViT-T [64]	Transformer	19M	4.6G	82.1
DeiT-S [47]	Transformer	22M	4.6G	79.8
CSwin-T [10]	Transformer	23M	4.3G	82.7
DiNAT-T [16]	Transformer	28M	4.3G	82.7
CoAtNet-0 [7]	Transformer	25M	4.2G	81.6
T2T-ViT-14 [62]	Transformer	22M	4.8G	81.5
PlainMamba-L1 [58]	Mamba	7M	3.0G	77.9
Vim-S [66]	Mamba	26M	5.1G	80.3
LocalVim-S [22]	Mamba	28M	4.8G	81.2
PlainMamba-L2 [58]	Mamba	25M	8.1G	81.6
Mamba2D-S [27]	Mamba	24M	–	81.7
EfficientVMamba-B [38]	Mamba	33M	4.0G	81.8
VMamba-T [31]	Mamba	30M	4.9G	82.6
LocalVMamba-T [22]	Mamba	26M	5.7G	82.7
SBM-T	SBM	25M	4.5G	83.2

Method	Type	Params	FLOPs	Top-1
ConvNeXt-S [34]	CNN	50M	8.7G	83.1
MambaOut-S [61]	CNN	48M	9.0G	84.1
PVTv2-B3 [53]	Transformer	45M	7.9G	83.2
CSwin-S [10]	Transformer	35M	6.9G	83.6
Focal-S [59]	Transformer	51M	9.4G	83.6
MViTv2-S [28]	Transformer	35M	7.0G	83.6
HiViT-S [64]	Transformer	38M	9.1G	83.5
CoAtNet-1 [7]	Transformer	42M	8.4G	83.3
PlainMamba-L3 [58]	Mamba	50M	14.4G	82.3
VMamba-S [31]	Mamba	50M	8.7G	83.6
LocalVMamba-S [22]	Mamba	50M	11.4G	83.7
SBM-S	SBM	43M	8.2G	84.4
ConvNeXt-B [34]	CNN	89M	15.4G	83.8
MambaOut-B [61]	CNN	85M	15.8G	84.2
RepLKNet-31B [9]	CNN	79M	15.3G	83.5
PVTv2-B5 [53]	Transformer	82M	11.8G	83.8
Focal-B [59]	Transformer	90M	16.4G	84.0
CSwin-B [10]	Transformer	78M	15.0G	84.2
HiViT-B [64]	Transformer	66M	15.9G	83.8
DeiT-B [47]	Transformer	86M	17.5G	81.8
CoAtNet-2 [7]	Transformer	75M	15.7G	84.1
Mamba2D-B [27]	Mamba	94M	–	83.0
VMamba-B [31]	Mamba	89M	15.4G	83.9
SBM-B	SBM	93M	18.2G	85.1

Table 1. Comparison of state-of-the-art (SOTA) vision models on the ImageNet-1K dataset. This table provides a comprehensive comparison across multiple model architectures, including CNNs, Transformers, Mamba models, and our proposed SBM models.

Moreover, SBM surpasses MambaOut [61], which replaces the selective SSM module in Mamba with a gated convolution module. This highlights SBM's architectural advantages in maintaining both accuracy and efficiency. Beyond classification accuracy, SBM also excels in computational throughput. As illustrated in Fig. 4, SBM achieves higher image processing rates per second than both Vision Transformers and Mamba-based models, benefiting from its optimized element-wise multiplication design. This improved efficiency reduces training and inference time, making SBM particularly well-suited for large-scale deployment where both high accuracy and speed are essential.

Overall, these results solidify SBM as a competitive and efficient vision backbone, capable of delivering state-of-the-art performance while maintaining computational efficiency. Its balanced design enables superior accuracy and throughput, making it a promising solution for large-scale image classification and other vision applications.

5.2. Transfer Learning on CIFAR-100

Transfer learning enables models trained on large datasets like ImageNet to adapt to smaller ones such as CIFAR-100 (32×32 resolution) with minimal data and computation. We

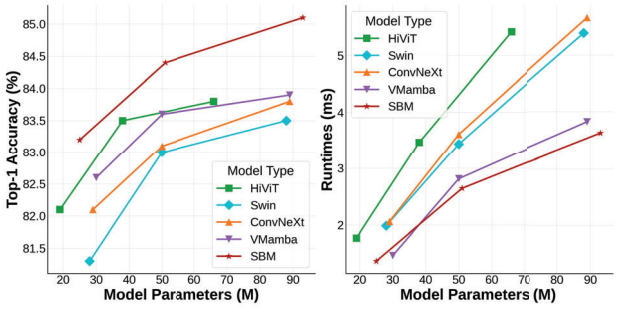

Figure 4. Left: Top-1 accuracy versus model parameters, depicting the classification performance of different architectures. Right: Runtime per image versus model parameters, measured using a RTX3090 GPU to illustrate computational efficiency.

evaluate SBM-B's transferability on CIFAR-100 using fine-tuning and linear probing. In fine-tuning, all parameters of the pre-trained SBM-B model are updated, while in linear probing, only a linear classifier is trained on top of a frozen backbone to assess feature quality.

Table 2 summarizes the results. SBM-B outperforms prior methods in both paradigms. In fine-tuning, it achieves 91.9% top-1 accuracy, surpassing DeiT-B (90.5%) by 1.4%, demonstrating its bilinear attention's effectiveness in captur-

Method	Fine-tuning	Linear probing
Deit-B [47]	90.5	80.6
Evo-ViT [57]	90.1	79.1
EViT [29]	90.0	80.2
TPS [55]	90.1	76.5
SBM-B	91.9	81.7

Table 2. Results of transfer learning on CIFAR-100 using two methods: fine-tuning and linear probing. The model is pretrained on ImageNet-1K.

Semantic Segmentation on ADE20K			
Backbone	**mIoU**	**#Params.**	**FLOPs**
ResNet-50 [18]	42.1	67M	953G
DeiT-S + MLN [47]	43.8	58M	1217G
Swin-T [32]	44.5	60M	945G
ConvNeXt-T [34]	46.0	60M	939G
Focal-T [59]	47.0	62M	998G
Twins-S [4]	46.2	54M	901G
NAT-T [17]	47.1	58M	934G
Vim-S [66]	44.9	46M	-
SBM-T	47.2	64M	967G
ResNet-101 [18]	43.8	86M	1030G
DeiT-B + MLN [47]	45.5	144M	2007G
Swin-S [32]	47.6	81M	1039G
ConvNeXt-S [34]	48.7	82M	1027G
Focal-S [59]	48.0	85M	1130G
Twins-B [4]	47.7	89M	1020G
NAT-S [17]	48.0	82M	1010G
SBM-S	49.1	86M	1081G
Swin-B [32]	48.1	121M	1188G
ConvNeXt-B [34]	49.1	122M	1170G
Focal-B [17]	49.0	126M	1354G
Twins-L [4]	48.8	133M	1164G
NAT-B [17]	48.5	123M	1137G
RepLKNet-31B [9]	49.9	112M	1170G
MambaOut-B [61]	49.6	112M	1178G
SBM-B	50.2	139M	1196G

Table 3. Semantic segmentation results on ADE20K. FLOPs are calculated with an input size of 512×2048.

ing fine-grained features. In linear probing, SBM-B attains 81.7%, exceeding DeiT-B's 80.6% by 1.1%, confirming the superiority of its pre-trained features.

Compared to Evo-ViT [57] and TPS [55], SBM-B exhibits notable improvements, reinforcing its robustness and adaptability. These results highlight SBM-B's strong generalization in transfer learning, making it a compelling choice for efficient and scalable vision tasks.

5.3. Semantic Segmentation on ADE20K

We evaluate SBM's performance on the challenging task of semantic segmentation using the widely recognized ADE20K [65] dataset, which includes 150 diverse semantic categories, covering a broad range of indoor and outdoor scenes. The ADE20K dataset comprises 20,000 images for training and 2,000 for validation, each labeled at the pixel level to facilitate detailed scene parsing. For this evaluation, we use UperNet [56], a standard framework for semantic segmentation in the mmsegmentation library [6], as our baseline model, allowing a fair comparison with other backbone architectures in this domain.

Consistent with the trends observed in previous experiments, SBM achieves superior performance across all model scales compared to other contemporary architectures. Specifically, SBM-T achieves a mean Intersection over Union (mIoU) of 47.2%, outperforming Swin-T [32], ConvNeXt-T [34], and NAT-T [17], while maintaining comparable model size and computational cost. Similarly, SBM-S attains an mIoU of 49.1%, surpassing ConvNeXt-S [34] and NAT-S [17] by a notable margin. For larger models, SBM-B achieves the highest mIoU of 50.2% among the compared methods, further demonstrating its scalability and effectiveness in handling complex semantic segmentation tasks.

Overall, the results demonstrate that SBM not only maintains computational efficiency but also delivers high accuracy across varying model sizes, confirming its robustness and adaptability in capturing fine-grained semantic information.

5.4. Object Detection on COCO 2017

COCO 2017 (Mask R-CNN 1× schedule)					
Backbone	Type	AP^b	AP^m	Params	FLOPs
Swin-T	Transformer	42.7	39.3	48M	267G
VMamba-T	Mamba	46.5	42.1	42M	262G
SBM-T	SBM	**47.2**	**42.6**	49M	263G
Swin-S	Transformer	44.8	40.9	69M	354G
VMamba-S	Mamba	48.2	43.0	64M	357G
SBM-S	SBM	**49.1**	**44.1**	65M	337G
Swin-B	Transformer	46.9	42.3	107M	496G
Vmamba-B	Mamba	48.5	43.1	96M	482G
SBM-B	SBM	**50.2**	**44.8**	112M	497G

Table 4. Object detection results on COCO 2017.

We evaluate SBM's performance on object detection using the COCO 2017 [30] dataset, a widely used benchmark for evaluating detection and instance segmentation models. COCO 2017 contains over 118K training images and 5K validation images, annotated with bounding boxes and instance masks spanning 80 object categories. We adopt the Mask R-CNN [19] framework with the standard $1\times$ training schedule and compare SBM against leading backbone architectures, including Swin [32] and VMamba [31].

As shown in Table 4, SBM consistently outperforms competing methods across all model scales. Specifically, SBM-T achieves an AP^b of 47.2 and an AP^m of 42.6, exceeding Swin-T by 4.5 and 3.3 points, respectively, while maintaining comparable computational cost. Similarly, SBM-S attains 49.1 AP^b and 44.1 AP^m, surpassing VMamba-S by 0.9

Architecture	#Params.	FLOPs	Throughput	Top-1
Block design all.	25M	4.5G	732	83.2
− Linear mapping branch	23M	4.1G	821	82.3 ↓ 0.9
− Bilinear mapping	21M	3.9G	914	81.6 ↓ 1.6
− Mapping control	24M	4.3G	826	82.6 ↓ 0.6

Table 5. Ablation study on the main branches of the SBM module. "Block design all" refers to the complete SBM block utilized in SBM-T, while "Linear mapping branch," "Bilinear mapping," and "Mapping control" correspond to the V matrix, U matrix, and $G(\mathbf{x})$ as described in Section 4.2. These terms represent the ablated versions where each respective branch is removed, as illustrated in Fig.3. The table demonstrates the impact of omitting each component on model performance and throughput, highlighting the contribution of each branch to the effectiveness of the SBM block. The Top-1 accuracy drop is indicated by (↓) for each ablation.

and 1.1 points, respectively, demonstrating its efficiency in object detection tasks. At the large model scale, SBM-B achieves the highest performance, with an AP^b of 50.2 and an AP^m of 44.8, outperforming VMamba-B by 1.7 and 1.7 points, respectively, further highlighting its scalability and effectiveness in complex detection scenarios.

5.5. Ablation Study

To evaluate the impact of each distinct component in our model, we conduct an ablation study that systematically removes or modifies key elements. This study is divided into two parts. First, we analyze the importance of bilinear mapping, linear mapping branch, and mapping control within the SBM module as shown in Fig 3. Then, we perform additional ablations to further investigate the impact of components within the SBM Block as shown in Fig. 2(b). Throughput is measured using a RTX3090 GPU. The results of these experiments are summarized in Table 5 and Table 6.

5.5.1 Ablation on SBM Module

To assess the contribution of each branch in the SBM module, we isolate each component in separate experiments, and the results are presented in Table 5. Removing the linear mapping branch results in an increase in throughput, but leads to a noticeable drop in top-1 accuracy, from 83.2% to 82.3% (↓ 0.9). The removal of bilinear mapping causes a more substantial decline in accuracy, with top-1 accuracy decreasing to 81.6% (↓ 1.6), highlighting the essential role of bilinear mapping in capturing global contextual information. Finally, removing mapping control leads to a smaller decrease in accuracy, from 83.2% to 82.6% (↓ 0.6), suggesting that while its impact is more modest, it still contributes positively to performance without significant loss in efficiency.

5.5.2 Ablation on Key Components in the SBM Block

In this section, we evaluate the importance of key components in the SBM block by keeping all three main branches:

Architecture	#Params	FLOPs	Throughput	Top-1
Baseline	22M	4.3G	860	82.3
+ APE	22M	4.3G	801	82.6
+ CPE	23M	4.4G	754	82.9
+ Shortcut	22M	4.3G	853	82.4
− Normalization	22M	4.3G	786	73.4
+ Block design all.	25M	4.5G	732	83.2

Table 6. Ablation study on the key components of the SBM block. The symbols "+" and "−" indicate the addition and removal of specific components, respectively, relative to the baseline configuration. "Baseline" refers to the SBM block without any additional enhancements. APE and CPE represent absolute and conditional positional encoding. Each component's effect on the model's parameters, FLOPs, throughput, and Top-1 accuracy is shown.

positional encoding, layer normalization (LN), and shortcut connections. The baseline model in Table 6 represents the SBM block without any form of positional encoding or shortcuts but includes layer normalization.

We begin by analyzing the baseline model, which has 22M parameters, a throughput of 860, and a top-1 accuracy of 82.3%. Adding Absolute Positional Encoding (APE) and Conditional Positional Encoding (CPE) slightly reduces throughput but improves performance by 0.3% and 0.6%, respectively, without introducing a significant increase in parameters. However, removing all LN layers in SBM block results in a substantial drop in performance, with top-1 accuracy plummeting to 73.4%. This highlights the critical role of layer normalization in stabilizing the training process. Finally, we test the full block design, incorporating all components together. This increases the model's parameter count to 25M and FLOPs to 4.5G, while the throughput decreases to 732. However, this configuration leads to an improved top-1 accuracy of 83.2%.

6. Conclusion

This paper presents Star with Bilinear Mapping (SBM), a Transformer-like architecture designed to overcome the quadratic complexity of traditional attention mechanisms through a bilinear mapping module combined with star operations, achieving global contextual modeling with linear complexity. Our experiments on image classification and semantic segmentation tasks, demonstrate that SBM provides competitive accuracy with significantly improved computational efficiency. Ablation studies further confirm the effectiveness of its components, underscoring SBM's potential as a scalable and high-performance model for efficient contextual modeling in the computer vision community.

7. Acknowledgments

This work was supported by the NSFC under Grant 62322604, 62176159 and 62401361, and in part by the Shanghai Municipal Science and Technology Major Project under Grant 2021SHZDZX0102.

References

[1] Jimmy Lei Ba. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[2] Tom B Brown. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 1

[3] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2017. 2

[4] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. *Advances in neural information processing systems*, 34: 9355–9366, 2021. 7

[5] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021. 4

[6] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. https://github.com/open-mmlab/mmsegmentation, 2020. 7

[7] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *Advances in neural information processing systems*, 34:3965–3977, 2021. 6

[8] Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024. 1, 2

[9] Xiaohan Ding, Xiangyu Zhang, Jungong Han, and Guiguang Ding. Scaling up your kernels to 31x31: Revisiting large kernel design in cnns. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11963–11975, 2022. 6, 7

[10] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. In *CVPR*, 2022. 1, 2, 6

[11] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2020. 1, 2

[12] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023. 1, 2

[13] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021. 2

[14] Dongchen Han, Ziyi Wang, Zhuofan Xia, Yizeng Han, Yifan Pu, Chunjiang Ge, Jun Song, Shiji Song, Bo Zheng, and Gao Huang. Demystify mamba in vision: A linear attention perspective. *arXiv preprint arXiv:2405.16605*, 2024. 1, 2

[15] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Advances in neural information processing systems*, 34:15908–15919, 2021. 2

[16] Ali Hassani and Humphrey Shi. Dilated neighborhood attention transformer. *arXiv preprint arXiv:2209.15001*, 2022. 6

[17] Ali Hassani, Steven Walton, Jiachen Li, Shen Li, and Humphrey Shi. Neighborhood attention transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6185–6194, 2023. 7

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 2, 7

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn, 2018. 7

[20] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021. 4

[21] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708, 2017. 1, 2

[22] Tao Huang, Xiaohuan Pei, Shan You, Fei Wang, Chen Qian, and Chang Xu. Localmamba: Visual state space model with windowed selective scan. *arXiv preprint arXiv:2403.09338*, 2024. 2, 6

[23] Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International conference on machine learning*, pages 5156–5165. PMLR, 2020. 2

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2, 5

[25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 2012. 1, 2, 5

[26] Carlos S Kubrusly. *Bilinear Maps and Tensor Products in Operator Theory*. Springer, 2023. 2, 3

[27] Shufan Li, Harkanwar Singh, and Aditya Grover. Mamba-nd: Selective state space modeling for multi-dimensional data. *arXiv preprint arXiv:2402.05892*, 2024. 6

[28] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Mvitv2: Improved multiscale vision transformers for classification and detection. In *CVPR*, 2022. 6

[29] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. Not all patches are what you need: Expediting vision transformers via token reorganizations. In *Proceedings of ICLR*, 2022. 7

[30] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. 2, 5, 7

[31] Yue Liu, Yunjie Tian, Yuzhong Zhao, Hongtian Yu, Lingxi Xie, Yaowei Wang, Qixiang Ye, and Yunfan Liu. Vmamba: Visual state space model. *arXiv preprint arXiv:2401.10166*, 2024. 1, 2, 5, 6, 7

[32] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021. 1, 2, 4, 5, 6, 7

[33] Ze Liu, Han Hu, Yutong Lin, Zhuliang Yao, Zhenda Xie, Yixuan Wei, Jia Ning, Yue Cao, Zheng Zhang, Li Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In *CVPR*, 2022. 1, 2

[34] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *CVPR*, 2022. 6, 7

[35] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 2

[36] I Loshchilov. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 5

[37] Xu Ma, Xiyang Dai, Yue Bai, Yizhou Wang, and Yun Fu. Rewrite the stars. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5694–5703, 2024. 1, 3

[38] Xiaohuan Pei, Tao Huang, and Chang Xu. Efficientvmamba: Atrous selective scan for light weight visual mamba. *arXiv preprint arXiv:2403.09977*, 2024. 2, 6

[39] Zelin Peng, Guanchun Wang, Lingxi Xie, Dongsheng Jiang, Wei Shen, and Qi Tian. Usage: A unified seed area generation paradigm for weakly supervised semantic segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 624–634, 2023. 1

[40] Zelin Peng, Zhengqin Xu, Zhilin Zeng, Yaoming Wang, Lingxi Xie, Qi Tian, and Wei Shen. Parameter-efficient fine-tuning in hyperspherical space for open-vocabulary semantic segmentation. *arXiv preprint arXiv:2405.18840*, 2024. 3

[41] Zelin Peng, Zhengqin Xu, Zhilin Zeng, Lingxi Xie, Qi Tian, and Wei Shen. Parameter efficient fine-tuning via cross block orchestration for segment anything model. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3743–3752, 2024. 3

[42] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 1

[43] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021. 1

[44] Yongming Rao, Wenliang Zhao, Yansong Tang, Jie Zhou, Ser Nam Lim, and Jiwen Lu. Hornet: Efficient high-order spatial interactions with recursive gated convolutions. *NeurIPS*, 35:10353–10366, 2022. 1

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 1, 2

[46] Yunjie Tian, Lingxi Xie, Jihao Qiu, Jianbin Jiao, Yaowei Wang, Qi Tian, and Qixiang Ye. Fast-itpn: Integrally pre-trained transformer pyramid network with token migration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024. 2

[47] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 1, 2, 6, 7

[48] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023. 1

[49] Aäron Van Den Oord, Nal Kalchbrenner, and Koray Kavukcuoglu. Pixel recurrent neural networks. In *International conference on machine learning*, pages 1747–1756. PMLR, 2016. 2

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 1

[51] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020. 2

[52] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *ICCV*, 2021. 1, 2

[53] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 2022. 6

[54] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7794–7803, 2018. 2

[55] Siyuan Wei, Tianzhu Ye, Shen Zhang, Yao Tang, and Jiajun Liang. Joint token pruning and squeezing towards more aggressive compression of vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2092–2101, 2023. 7

[56] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *ECCV*, 2018. 7

[57] Yifan Xu, Zhijie Zhang, Mengdan Zhang, Kekai Sheng, Ke Li, Weiming Dong, Liqing Zhang, Changsheng Xu, and Xing Sun. Evo-vit: Slow-fast token evolution for dynamic vision transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 2964–2972, 2022. 7

[58] Chenhongyi Yang, Zehui Chen, Miguel Espinosa, Linus Ericsson, Zhenyu Wang, Jiaming Liu, and Elliot J Crowley. Plainmamba: Improving non-hierarchical mamba in visual recognition. *arXiv preprint arXiv:2403.17695*, 2024. 1, 6

[59] Jianwei Yang, Chunyuan Li, Xiyang Dai, and Jianfeng Gao. Focal modulation networks. *Advances in Neural Information Processing Systems*, 35:4203–4217, 2022. 1, 6, 7

[60] Songlin Yang, Bailin Wang, Yika7ng Shen, Rameswar Panda, and Yoon Kim. Gated linear attention transformers with hardware-efficient training. *arXiv preprint arXiv:2312.06635*, 2023. 1

[61] Weihao Yu and Xinchao Wang. Mambaout: Do we really need mamba for vision? *arXiv preprint arXiv:2405.07992*, 2024. 1, 6, 7

[62] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis E.H. Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 558–567, 2021. 6

[63] Xiangrong Zhang, Zelin Peng, Peng Zhu, Tianyang Zhang, Chen Li, Huiyu Zhou, and Licheng Jiao. Adaptive affinity loss and erroneous pseudo-label refinement for weakly supervised semantic segmentation. In *Proceedings of the 29th ACM international conference on multimedia*, pages 5463–5472, 2021. 1

[64] Xiaosong Zhang, Yunjie Tian, Lingxi Xie, Wei Huang, Qi Dai, Qixiang Ye, and Qi Tian. Hivit: A simpler and more efficient design of hierarchical vision transformer. In *The Eleventh International Conference on Learning Representations*, 2023. 6

[65] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *CVPR*, 2017. 2, 5, 7

[66] Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: Efficient visual representation learning with bidirectional state space model. *arXiv preprint arXiv:2401.09417*, 2024. 2, 6, 7