

MambaOut: Do We Really Need Mamba for Vision?

Weihaoyu Yu Xinchao Wang
National University of Singapore
weihaoyu@u.nus.edu xinchao@nus.edu.sg
Code: <https://github.com/yuweihaoyu/MambaOut>

In memory of Kobe Bryant

“What can I say, Mamba out.” — Kobe Bryant’s NBA farewell speech in 2016.

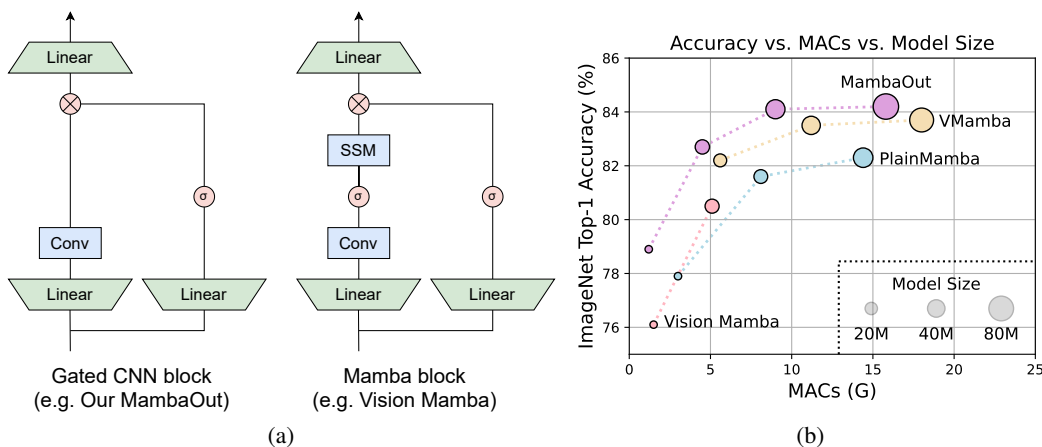


Figure 1: (a) Architecture of Gated CNN [18] and Mamba [25] blocks (omitting Normalization and shortcut). The Mamba block extends the Gated CNN with an additional state space model (SSM). As will be conceptually discussed in Section 3, SSM is not necessary for image classification on ImageNet [19, 66]. To empirically verify this claim, we stack Gated CNN blocks to build a series of models named *MambaOut*. (b) MambaOut outperforms visual Mamba models, e.g., Vision Mamba [104], VMamba [50] and PlainMamba [88], on ImageNet image classification.

Abstract — Mamba, an architecture with RNN-like token mixer of state space model (SSM), was recently introduced to address the quadratic complexity of the attention mechanism and subsequently applied to vision tasks¹. Nevertheless, the performance of Mamba for vision is often underwhelming when compared with convolutional and attention-based models. In this paper, we delve into the essence of Mamba, and conceptually conclude that Mamba is ideally suited for tasks with *long-sequence* and *autoregressive* characteristics. For vision tasks, as image classification does not align with either characteristic, we hypothesize that Mamba is not necessary for this task; Detection and segmentation tasks are also not *autoregressive*, yet they adhere to the *long-sequence* characteristic, so we believe it is still worthwhile to explore Mamba’s potential for these tasks. To empirically verify our hypotheses, we construct a series of models named *MambaOut* through stacking Mamba blocks while removing their core token mixer, SSM. Experimental results strongly support our hypotheses. Specifically, our MambaOut model surpasses all visual Mamba models on ImageNet image classification, indicating that Mamba is indeed unnecessary for this task. As for detection and segmentation, MambaOut cannot match the performance of state-of-the-art visual Mamba models, demonstrating the potential of Mamba for long-sequence visual tasks.

¹The vision tasks we discuss in this paper include image classification on ImageNet [19, 66], object detection & instance segmentation on COCO [48] and semantic segmentation on ADE20K [103].

1 Introduction

In recent years, Transformer [76] has become the mainstream backbone for various tasks, underpinning numerous prominent models such as BERT [20], GPT series [60, 61, 6, 1] and ViT [23]. However, the token mixer of Transformer, attention [3], incurs a quadratic complexity with respect to sequence length, posing major challenges for long sequences. To address this issue, a variety of token mixers with linear complexity to token length have been introduced [72], such as dynamic convolution [82, 84, 39], Linformer [78], Longformer [5], Big Bird [97], and Performer [12]. More recently, a new wave of RNN-like models has emerged [40, 98, 26, 59, 25], drawing significant interest from the community for their capability of parallelizable training and performing efficient inference on long sequences. Notably, models like RWKV [59] and Mamba [25] are proven to be effective as the backbone for large language models (LLMs) [59, 47].

Motivated by the promising capabilities of RNN-like models, various research endeavors have attempted to introduce Mamba [25] into visual recognition tasks, exemplified by the pioneering works of Vision Mamba [104], VMamba [50], LocalMamba [37], and PlainMamba [88], *etc.* The token mixer of Mamba is the structured state space models (SSM) [27, 26, 25], under the spirit of RNN. Nevertheless, their experiments show that the SSM based models for vision, in reality, lead to underwhelming performance compared with state-of-the-art convolutional [52, 21, 28, 64, 89, 49, 92, 35, 79, 93] and attention-based models [16, 74, 22, 95, 46, 75, 70, 92]. This gives rise to a compelling research question: *Do we really need Mamba for Vision?*

In this paper, we investigate the nature of Mamba, and conceptually summarize that Mamba is ideally suited for tasks with two key characteristics: *long-sequence* and *autoregressive*, because of the inherent RNN mechanism of SSM [27, 26, 25] (see explanation of Figure 2 and Figure 3). Unfortunately, not many vision tasks possess both characteristics. Image classification on ImageNet, for example, conforms to neither, while object detection & instance segmentation on COCO and semantic segmentation on ADE20K conform only to the *long-sequence*. *Autoregressive* characteristic, on the other hand, demands that each token aggregate information solely from preceding and current tokens, a concept denoted as *causal mode* for token mixing [63] (see Figure 3(a)). In fact, all visual recognition tasks fall within the understanding domain rather than the generative one, meaning that the model can see the entire image at once. As such, imposing additional causal constraints on token mixing in visual recognition models could lead to a performance drop (see Figure 3(b)). Although this issue can be mitigated via bidirectional branches [68], it is inevitable that the issue persists within each branch.

Based on the conceptual discussion above, we propose the two hypotheses as follows:

- *Hypothesis 1:* SSM is not necessary for image classification, since this task conforms to neither the *long-sequence* or *autoregressive* characteristic.
- *Hypothesis 2:* SSM may be potentially beneficial for object detection & instance segmentation and semantic segmentation, since they follow the *long-sequence* characteristic, though they are not *autoregressive*.

To experimentally validate our hypotheses, we developed a series of models termed *MambaOut* through stacking Gated CNN [18] blocks. The key distinction between Gated CNN and Mamba blocks lies in the existence of SSM, as illustrated in Figure 1(a). Experimental results demonstrate that the simpler MambaOut model, in reality, already surpasses the performance of visual Mamba models [104, 50, 37, 88], which in turn verifies our *Hypothesis 1*. We also show empirical results that MambaOut falls short of matching the performance of state-of-the-art visual Mamba models [50, 37] in detection and segmentation tasks (see Tables 2 and 3), which underscores the potential of SSM on these tasks and effectively validates our *Hypothesis 2*.

The contributions of our paper are threefold. Firstly, we analyze the RNN-like mechanism of SSM and conceptually conclude that Mamba is suited for tasks with long-sequence and autoregressive characteristics. Secondly, we examine the characteristics of visual tasks and hypothesize that SSM is unnecessary for image classification on ImageNet since this task does not meet either characteristic, yet exploring the potential of SSM for detection and segmentation tasks remains valuable since these tasks conform to long-sequence characteristic, though they are not autoregressive. Thirdly, we develop a series of models named MambaOut based on Gated CNN blocks but without SSM. Experiments show that MambaOut effectively surpasses visual Mamba models in ImageNet image

classification but does not reach the performance of state-of-the-art visual Mamba models in detection and segmentation tasks. These observations, in turn, validate our hypotheses. As such, MambaOut, because of its *Occam's razor* nature, may readily serve as a natural baseline for future research on visual Mamba models.

2 Related work

Transformer has been widely utilized across various domains, including BERT [20] and GPT series [60, 61, 6, 1] in NLP and ViT [23] in computer vision. However, the attention module in Transformers scales quadratically with sequence length, presenting a significant computational challenge. Numerous studies [72] have explored various strategies to mitigate this issue, including low-rank approaches [78], kernelization [40, 12], token mixing range limitation [5, 97, 51, 29], and history memory compression [62]. More recently, RNN-like methods [17, 40, 98], particularly RWKV [59] and Mamba [25], have garnered attention for their promising results in large language models [59, 47].

Eager exploratory researchers have quickly moved to incorporate SSM and Mamba [25] into visual recognition tasks [104, 50, 37, 88, 44, 57, 58, 99, 86]. For instance, Vision Mamba [104] integrates Mamba [25] to develop isotropic vision models akin to ViT [23]; VMamba [50] employs Mamba to construct hierarchical vision models similar to AlexNet [42] and ResNet [32]; LocalMamba [37] enhances visual Mamba models [104, 50] by incorporating local inductive biases; PlainMamba [88] aims to further enhance the performance of isotropic Mamba models; EfficientVMamba [58] focuses on efficiency through the introduction of atrous selective scan for lightweight visual Mamba models.

Unlike these initiatives, our work does not aim to design new visual Mamba models. Instead, we explore a pertinent research question about the necessity of Mamba [25] in visual recognition contexts [19, 66, 48, 103]. We hope this paper can provide insights for future research on visual Mamba models.

3 Conceptual discussion

In this section, we first discuss what characteristics of tasks the Mamba model is suited for. Next, we examine whether visual recognition tasks conform to these characteristics. Based on the examination results, we propose hypotheses regarding the necessity of Mamba for vision.

3.1 What tasks is Mamba suitable for?

The token mixer of Mamba is selective SSM [26, 25] which defines four input-dependent parameters $(\Delta, \mathbf{A}, \mathbf{B}, \mathbf{C})$ and transforms them to $(\bar{\mathbf{A}}, \bar{\mathbf{B}}, \mathbf{C})$ by

$$\bar{\mathbf{A}} = \exp(\Delta \mathbf{A}), \quad \bar{\mathbf{B}} = (\Delta \mathbf{A})^{-1}(\exp(\Delta \mathbf{A}) - \mathbf{I}) \cdot \Delta \mathbf{B}. \quad (1)$$

Then the sequence-to-sequence transformation of SSM can be expressed by

$$h_t = \bar{\mathbf{A}} h_{t-1} + \bar{\mathbf{B}} x_t, \quad (2)$$

$$y_t = \mathbf{C} h_t, \quad (3)$$

where t denotes the timestep, x_t represents the input, h_t signifies the hidden state, and y_t indicates the output. The recurrent property [34] of Equation 2 distinguishes RNN-like SSM from causal attention. The hidden state h can be seen as a fixed-size memory that stores all historical information. Through Equation 2, this memory is updated while its size remains constant. The fixed size means the memory is inevitably lossy, but it ensures that the computational complexity of integrating the memory with the current input remains constant. Conversely, causal attention stores all keys and values from previous tokens as its memory, which expands by adding the current token's key and value with each new input. This memory is theoretically lossless. However, as more tokens are inputted, the memory size grows, thereby increasing the complexity of integrating the memory with the current input. The differences in memory mechanisms between RNN-like models and causal attention are further illustrated in Figure 2.

Because SSM's memory is inherently lossy, it logically falls short of the lossless memory of attention. Consequently, Mamba cannot showcase its strengths in handling short sequences, an area where

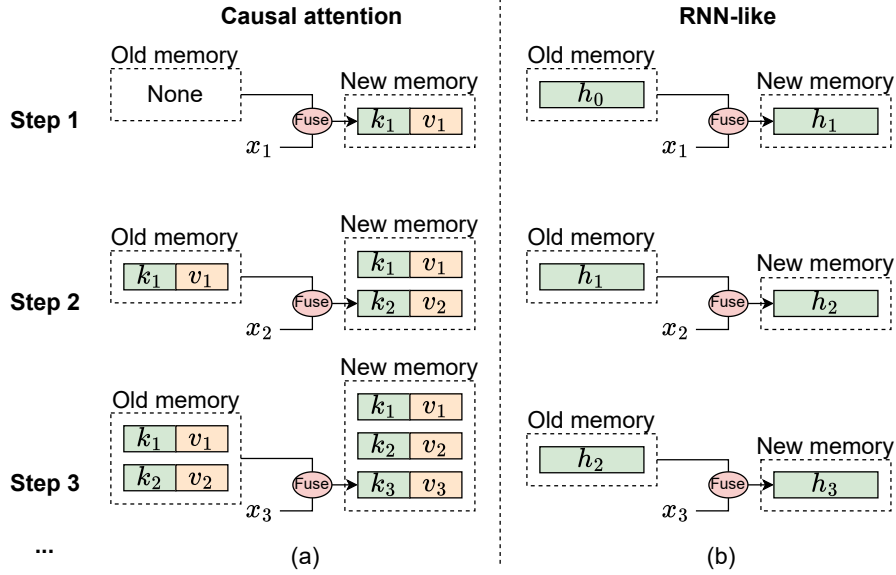


Figure 2: The mechanism illustration of causal attention and RNN-like models from memory perspective, where x_i denotes the input token of i -th step. (a) Causal attention stores all previous tokens’ keys k and values v as memory. The memory is updated by continuously adding the current token’s key and value, so the memory is lossless, but the downside is that the computational complexity of integrating old memory and current tokens increases as the sequence lengthens. Therefore, attention can effectively manage short sequences but may encounter difficulties with longer ones. (b) In contrast, RNN-like models compress previous tokens into fixed-size hidden state h , which serves as the memory. This fixed size means that RNN memory is inherently lossy, which cannot directly compete with the lossless memory capacity of attention models. Nonetheless, **RNN-like models can demonstrate distinct advantages in processing long sequences, as the complexity of merging old memory with current input remains constant, regardless of sequence length.**

attention performs well with ease. However, in scenarios involving long sequences, attention will falter due to its quadratic complexity. In this case, Mamba can distinctly highlight its efficiency in merging memory with the current input, thus managing long sequences smoothly. Therefore, Mamba is particularly well-suited for processing long sequences.

Although the recurrent nature of SSM (Equation 2) allows Mamba to handle long sequences efficiently, it introduces a significant limitation: h_t can only access information from the previous and current timesteps. As illustrated in Figure 3, this type of token mixing is termed causal mode, which can be formulated as:

$$y_t = f(x_1, x_2, \dots, x_t), \quad (4)$$

where x_t and y_t represent the input and output of the t -th token, respectively. Due to its causal nature, this mode is well-suited for autoregressive generation tasks.

Another mode is called fully-visible mode, where each token can aggregate information from all preceding and subsequent tokens. This means the output of each token depends on the inputs from all tokens:

$$y_t = f(x_1, x_2, \dots, x_t, \dots, x_T), \quad (5)$$

where T represents the total number of tokens. The fully-visible mode is suitable for understanding tasks, where all inputs can be accessed by the model at once.

Attention is in fully-visible mode by default, but it can easily turn into causal mode by applying causal masks to the attention maps. RNN-like models inherently operate in causal mode due to their recurrent properties, as illustrated by Mamba’s Equation 2. Due to this inherent characteristic, RNN-like models cannot be transformed into fully-visible mode. Although RNNs can approximate a fully-visible mode using bidirectional branches, each branch still individually remains in causal

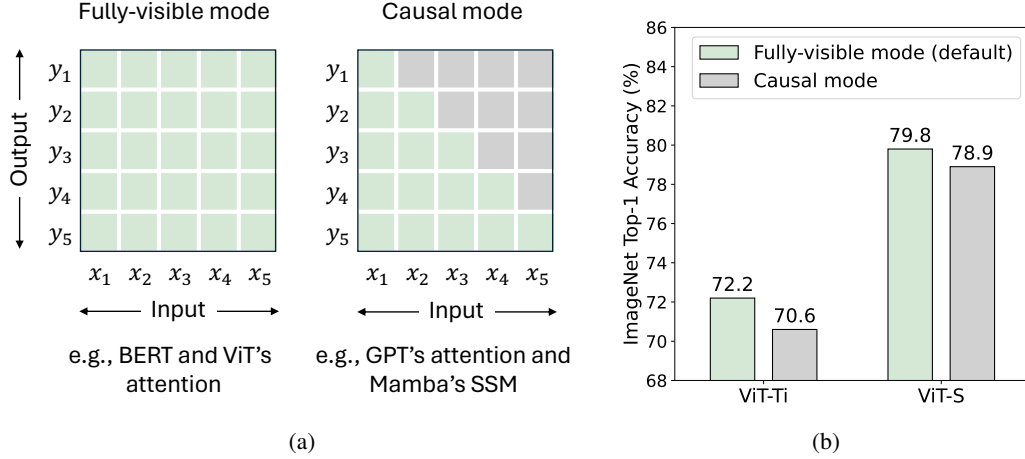


Figure 3: (a) Two modes of token mixing [63]. For a total of T tokens, the fully-visible mode allows token t to aggregate inputs from all tokens, i.e., $\{x_i\}_{i=1}^T$, to compute its output y_t . In contrast, the causal mode restricts token t to only aggregate inputs from preceding and current tokens $\{x_i\}_{i=1}^t$. By default, attention operates in fully-visible mode but can be adjusted to causal mode with causal attention masks. RNN-like models, such as Mamba’s SSM [25, 26], inherently operate in causal mode due to their recurrent nature. (b) **We modify the ViT’s attention [23, 73] from fully-visible to causal mode and observe performance drop on ImageNet, which indicates causal mixing is unnecessary for understanding tasks.**

mode. Therefore, Mamba is well-suited for tasks that require causal token mixing, due to the inherent limitations of its recurrent properties.

In summary, Mamba is ideally suited for tasks that display the following characteristics:

- *Characteristic 1:* The task involves processing long sequences.
- *Characteristic 2:* The task requires causal token mixing mode.

Next, we will discuss whether visual recognition tasks exhibit these two characteristics.

3.2 Do visual recognition tasks have very long sequences?

In this subsection, we explore whether visual recognition tasks necessitate long sequence modeling. We use the Transformer model [76] as a case study to facilitate our analysis. Consider a Transformer block with a common MLP ratio of 4; assuming its input $X \in \mathbb{R}^{L \times D}$ has a token length of L and channel (embedding) dimensions of D , the FLOPs for the block can be calculated as:

$$\text{FLOPs} = 24D^2L + 4DL^2. \quad (6)$$

From this, we derive the ratio of the quadratic term to the linear term in L as:

$$r_L = \frac{4DL^2}{24D^2L} = \frac{L}{6D}. \quad (7)$$

If $L > 6D$, the computational load of the quadratic term in L surpasses that of the linear term. This provides a simple metric to determine if the task involves long sequences. For instance, with 384 channels in ViT-S, the threshold $\tau_{\text{small}} = 6 \times 384 = 2304$, and for 768 channels in ViT-B, $\tau_{\text{base}} = 6 \times 768 = 4608$.

For image classification on ImageNet, the typical input image size is 224^2 , resulting in $14^2 = 196$ tokens with patch size of 16^2 . Clearly, 196 is much less than both τ_{small} and τ_{base} , indicating that image classification on ImageNet does not qualify as a long-sequence task.

For object detection & instance segmentation on COCO, with an inference image size of 800×1280 , and for semantic segmentation on ADE20K, with an inference image size of 512×2048 , the number of tokens is approximately 4K, given patch size of 16^2 . Since $4K > \tau_{\text{small}}$ and $4K \approx \tau_{\text{base}}$, both detection on COCO and segmentation on ADE20K can be considered long-sequence tasks.

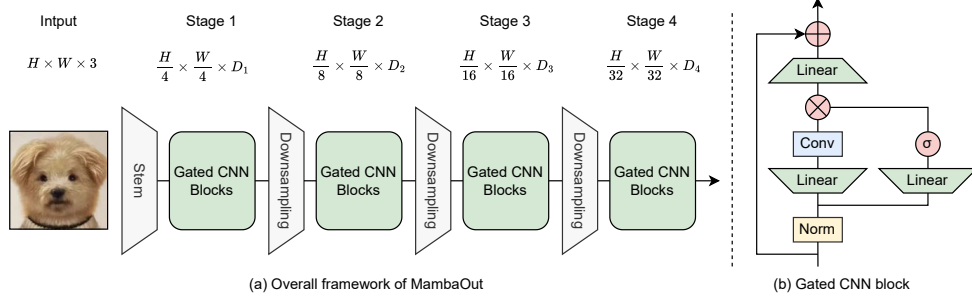


Figure 4: **(a) The overall framework of MambaOut for visual recognition.** Similar to ResNet [32], MambaOut adopts hierarchical architecture with four stages. D_i represents the channel dimensions at the i -th stage. **(b) The architecture of Gated CNN block.** The difference between the Gated CNN block [18] and the Mamba block [25] lies in the absence of the SSM (state space model) in the Gated CNN block.

3.3 Do visual recognition tasks need causal token mixing mode?

As discussed in Section 3.1 and illustrated in Figure 3, the fully-visible token mixing mode allows unrestricted range of mixing, whereas the causal mode limits the current token to only access information from preceding tokens. Visual recognition is categorized as an understanding task, wherein the model can see the entire image at once, eliminating the need for restrictions on token mixing. Imposing additional constraints on token mixing can potentially degrade model performance. As demonstrated in Figure 3(b), when causal restrictions are applied to Vision Transformers (ViT) [23, 73], a noticeable decline in performance is observed. Generally, the fully-visible mode is appropriate for understanding tasks, while the causal mode is better suited for autoregressive tasks. This claim can also be substantiated by the observation that BERT [20] and ViT [23] (BEiT [4] and MAE [30]) are used more for understanding tasks than GPT-1/2 [60, 61] and image GPT [9]. Therefore, visual recognition tasks do not need causal token mixing mode.

3.4 Hypotheses regarding the necessity of Mamba for vision

Based on our preceding discussion, we summarize our hypotheses regarding the necessity of introducing Mamba for visual recognition tasks as follows:

- *Hypothesis 1:* It is not necessary to introduce SSM for image classification on ImageNet, as this task does not meet *Characteristic 1* or *Characteristic 2*.
- *Hypothesis 2:* It is still worthwhile to further explore the potential of SSM for visual detection and segmentation since these tasks align with *Characteristic 1*, despite not fulfilling *Characteristic 2*.

4 Experimental verification

4.1 Gated CNN and MambaOut

Next, we aim to validate our hypotheses empirically. As depicted in Figure 1(a), Mamba block [25] is based on the Gated CNN block [18]. The meta-architecture of Gated CNN and Mamba can be considered as a simplified integration of the MetaFormer’s [91] token mixer and an MLP, akin to MetaNeXt [93]. Formally, given the input $X \in \mathbb{R}^{N \times D}$, the meta-architecture is formulated as:

$$X' = \text{Norm}(X), \quad (8)$$

$$Y = (\text{TokenMixer}(X'W_1) \odot \sigma(X'W_2))W_3 + X, \quad (9)$$

where $\text{Norm}(\cdot)$ represents normalization [38, 2, 83]; $\text{TokenMixer}(\cdot)$ refers to the module to conduct token mixing [92]; $W_1 \in \mathbb{R}^{D \times rD}$, $W_2 \in \mathbb{R}^{D \times rD}$ and $W_3 \in \mathbb{R}^{rD \times D}$ are learnable parameters with

Algorithm 1 PyTorch code of Gated CNN block

```
import torch
import torch.nn as nn

class GatedCNNBlock(nn.Module):
    def __init__(self, dim, expansion_ratio=8/3, kernel_size=7, conv_ratio=1.0,
                 norm_layer=partial(nn.LayerNorm, eps=1e-6),
                 act_layer=nn.GELU,
                 drop_path=0.):
        super().__init__()
        self.norm = norm_layer(dim)
        hidden = int(expansion_ratio * dim)
        self.fc1 = nn.Linear(dim, hidden * 2)
        self.act = act_layer()
        conv_channels = int(conv_ratio * dim)
        self.split_indices = (hidden, hidden - conv_channels, conv_channels)
        self.conv = nn.Conv2d(conv_channels, conv_channels, kernel_size=kernel_size, padding=kernel_size//2,
                              groups=conv_channels)
        self.fc2 = nn.Linear(hidden, dim)

    def forward(self, x):
        shortcut = x # [B, H, W, C] = x.shape
        x = self.norm(x)
        g, i, c = torch.split(self.fc1(x), self.split_indices, dim=-1)
        c = c.permute(0, 3, 1, 2) # [B, H, W, C] -> [B, C, H, W]
        c = self.conv(c)
        c = c.permute(0, 2, 3, 1) # [B, C, H, W] -> [B, H, W, C]
        x = self.fc2(self.act(g) * torch.cat((i, c), dim=-1))
        return x + shortcut
```

MLP expansion r ; σ is activation function [24, 33]. Token mixers of Gated CNN and Mamba are:

$$\text{TokenMixer}_{\text{GatedCNN}}(Z) = \text{Conv}(Z) \quad (10)$$

$$\text{TokenMixer}_{\text{Mamba}}(Z) = \text{SSM}(\sigma(\text{Conv}(Z))) \quad (11)$$

Comparing Equations 10 and 11, and referencing Figure 1(a), the primary distinction between the Gated CNN [60] and the Mamba block [25] lies in the presence of SSM. This prompts us to develop a series of models, termed MambaOut, which are based on the Gated CNN block without SSM. MambaOut will help us assess the necessity of Mamba for visual recognition tasks.

Specifically, we specify the token mixer of Gated CNN as depthwise convolution [11] of 7×7 kernel size, following ConvNeXt [52, 55]. Besides, to improve the practical speed, we only conduct depthwise convolution on partial channels [54, 93, 7], following InceptionNeXt [93]. As shown in Algorithm 1, the implementation of Gated CNN block is simple and elegant. Similar to ResNet, we adopt 4-stage framework to build MambaOut by stacking Gated CNN blocks at each stage, as depicted in Figure 4. The configuration details of each model size are shown in Table 4 in the appendix.

4.2 Image classification on ImageNet

Setup. ImageNet [19, 66] serves as the gold standard benchmark for image classification, encompassing a wide array of 1,000 common classes. It comprises approximately 1.3 million training images and 50,000 validation images. The training scheme follows DeiT [73] without distillation. Specifically, the used data augmentation contains random resized crop (input image size of 224^2), horizontal flip, RandAugment [15], Mixup [100], CutMix [96], Random Erasing [102] and color jitter; Regularization techniques include weight decay, stochastic depth [36] and label smoothing [71]. All our models are trained by AdamW [53, 41]. The learning rate scaling rule is $\text{lr} = \frac{\text{batchsize}}{1024} \times 10^{-3}$. In this paper, we set the batch size to 4096, so the learning rate is 0.004. Our MambaOut models are implemented with PyTorch [56] and timm [81] libraries and trained on TPU v3. More training hyper-parameters are shown in Table 5 in the appendix.

Results. The performance of our MambaOut models, visual Mamba models, and other various convolution and attention-based models on ImageNet [19, 66] is presented in Table 1. Notably, our MambaOut models, which do not incorporate SSM, consistently outperform visual Mamba models [104, 50, 37, 58, 88] that include SSM across all model sizes. For instance, the MambaOut-Small

Table 1: **Performance of models on ImageNet at the resolution of 224²**. Our MambaOut model employs the Gated CNN block [60]. The Mamba block [25], derived from the Gated CNN block, incorporates an additional SSM (state space model). It is evident that visual Mamba models fall short of MambaOut’s performance, let alone surpassing state-of-the-art convolutional or convolution-attention-hybrid models. *Note that VMambaV9 modifies the meta-architecture of the Mamba block to MetaFormer [92], different from other visual Mamba models and MambaOut.

Model	Token Mixing Type	Param (M)	Test@224 ²		Model	Token Mixing Type	Param (M)	Test@224 ²	
			MAC (G)	Acc (%)				MAC (G)	Acc (%)
VAN-B0 [28]	Conv	4	0.9	75.4	ConvNeXt-S [52]	Conv	50	8.7	83.1
MogaNet-T [45]	Conv	5	1.1	79.0	VAN-B3 [28]	Conv	45	9.0	83.9
FasterNet-T1 [7]	Conv	8	0.9	76.2	ConvFormer-S36 [92]	Conv	40	7.6	84.1
InceptionNeXt-A [93]	Conv	4	0.5	75.3	InternImage-S [79]	Conv	50	8	84.2
DeiT-Ti [73]	Attn	6	1.3	72.2	MogaNet-B [45]	Conv	44	9.9	84.3
T2T-ViT-7 [94]	Attn	4	1.1	71.7	T2T-ViT-19 [94]	Attn	39	8.5	81.9
PVTv2-B0 [80]	Conv + Attn	3	0.6	70.5	Swin-S [51]	Attn	50	8.7	83.0
MobileViTv3-XS [77]	Conv + Attn	3	0.9	76.7	Focal-Small [90]	Attn	51	9.1	83.5
EMO-6M [101]	Conv + Attn	6	1.0	79.0	CSWin-S [22]	Attn	35	6.9	83.6
Vim-Ti [104]	Conv + SSM	7	1.5	76.1	MViTv2-S [46]	Attn	35	7.0	83.6
LocalVim-T [37]	Conv + SSM	8	1.5	76.2	CoAtNet-1 [16]	Conv + Attn	42	8.4	83.3
EfficientVMamba-T [58]	Conv + SSM	6	0.8	76.5	UniFormer-B [43]	Conv + Attn	50	8.3	83.9
EfficientVMamba-S [58]	Conv + SSM	11	1.3	78.7	CAFormer-S36 [92]	Conv + Attn	39	8.0	84.5
MambaOut-Femto	Conv	7	1.2	78.9	SG-Former-M [65]	Conv + Attn	39	7.5	84.1
PoolFormer-S24 [91]	Pool	21	3.4	80.3	TransNeXt-Small [69]	Conv + Attn	50	10.3	84.7
ConvNeXt-T [52]	Conv	29	4.5	82.1	VMamba-S [50]	Conv + SSM	44	11.2	83.5
VAN-B2 [28]	Conv	27	5.0	82.8	LocalVMamba-S [37]	Conv + SSM	50	11.4	83.7
ConvFormer-S18 [92]	Conv	27	3.9	83.0	PlainMamba-L2 [88]	Conv + SSM	25	8.1	81.6
MogaNet-S [45]	Conv	25	5.0	83.4	VMambaV9-S [50]	Conv + SSM	50	8.7	83.6
InternImage-T [79]	Conv	30	5	83.5	MambaOut-Small	Conv	48	9.0	84.1
InceptionNeXt-T [93]	Conv	28	4.2	82.3	ConvNeXt-B [52]	Conv	89	15.4	83.8
DeiT-S [73]	Attn	22	4.6	79.8	RepLkNet-31B [21]	Conv	79	15.3	83.5
T2T-ViT-14 [94]	Attn	22	4.8	81.5	ConvFormer-M36 [92]	Conv	57	12.8	84.5
Swin-T [51]	Attn	29	4.5	81.3	HorNet-B [64]	Conv	88	15.5	84.3
Focal-Tiny [90]	Attn	29	4.9	82.2	MogaNet-L [45]	Conv	83	15.9	84.7
CSWin-T [22]	Attn	23	4.3	82.7	InternImage-B [79]	Conv	97	16	84.9
CoAtNet-0 [16]	Conv + Attn	25	4.2	81.6	DeiT-B [73]	Attn	86	17.5	81.8
iFormer-S [70]	Conv + Attn	20	4.8	83.4	T2T-ViT-24 [94]	Attn	64	13.8	82.3
MOAT-0 [87]	Conv + Attn	28	5.7	83.3	Swin-B [51]	Attn	88	15.4	83.5
CAFormer-S18 [92]	Conv + Attn	26	4.1	83.6	CSWin-B [22]	Attn	78	15.0	84.2
SG-Former-S [65]	Conv + Attn	23	4.8	83.2	MViTv2-B [46]	Attn	52	10.2	84.4
TransNeXt-Tiny [69]	Conv + Attn	28	5.7	84.0	CoAtNet-2 [16]	Conv + Attn	75	15.7	84.1
Vim-S [104]	Conv + SSM	26	5.1	80.5	iFormer-L [70]	Conv + Attn	87	14.0	84.8
VMamba-T [50]	Conv + SSM	22	5.6	82.2	MOAT-2 [87]	Conv + Attn	73	17.2	84.7
Mamba-2D-S [44]	Conv + SSM	24	—	81.7	CAFormer-M36 [92]	Conv + Attn	56	13.2	85.2
LocalVim-S [37]	Conv + SSM	28	4.8	81.2	TransNeXt-Base [69]	Conv + Attn	90	18.4	84.8
LocalVMamba-T [37]	Conv + SSM	26	5.7	82.7	VMamba-B [50]	Conv + SSM	75	18.0	83.7
EfficientVMamba-B [58]	Conv + SSM	33	4.0	81.8	Mamba-2D-B [44]	Conv + SSM	92	—	83.0
PlainMamba-L1 [88]	Conv + SSM	7	3.0	77.9	PlainMamba-L3 [88]	Conv + SSM	50	14.4	82.3
VMambaV9-T* [50]	Conv + SSM	31	4.9	82.5	VMambaV9-B [50]	Conv + SSM	89	15.4	83.9
MambaOut-Tiny	Conv	27	4.5	82.7	MambaOut-Base	Conv	85	15.8	84.2

model achieves top-1 accuracy of 84.1%, 0.4% higher than that of LocalVMamba-S [37], while requiring only 79% of the MACs. These results strongly support our *Hypothesis 1*, which posits that introducing SSM for image classification on ImageNet is unnecessary, aligning with the principle of Occam’s razor.

Additionally, visual Mamba models currently exhibit a significant performance gap when compared to state-of-the-art convolution and attention models. For instance, the CAFormer-M36 [92], which employs old-fashioned token mixers of simple separable convolutions from MobileNetV2 [67] and vanilla attention from Transformer [76] invented more than 7 years ago, outperforms all visual Mamba models of comparable size by more than 1% accuracy. Should future research aim to challenge our

Table 2: **Performance of object detection and instance segmentation on COCO with Mask R-CNN.** The MACs are measured with input size of 800×1280 .

Backbone	Token Mixing Type	Param (M)	MAC (G)	Mask R-CNN $1 \times$ schedule					
				AP ^b	AP ^b ₅₀	AP ^b ₇₅	AP ^m	AP ^m ₅₀	AP ^m ₇₅
ConvNeXt-T [49]	Conv	48	262	44.2	66.6	48.3	40.1	63.3	42.8
FocalNet-T [89]	Conv	49	268	46.1	68.2	50.6	41.5	65.1	44.5
Swin-T [51]	Attn	48	267	42.7	65.2	46.8	39.3	62.2	42.2
ViT-Adapter-S [10]	Attn	48	403	44.7	65.8	48.3	39.9	62.5	42.8
CSWin-T [22]	Attn	42	279	46.7	68.6	51.3	42.2	65.6	45.4
PVTv2-B2 [80]	Conv + Attn	45	309	45.3	67.1	49.6	41.2	64.2	44.4
SG-Former-S [65]	Conv + Attn	41	—	47.4	69.0	52.0	42.6	65.9	46.0
TransNeXt-Tiny [69]	Conv + Attn	48	356	49.9	71.5	54.9	44.6	68.6	48.1
VMamba-T [50]	Conv + SSM	42	286	46.5	68.5	50.7	42.1	65.5	45.3
LocalVMamba-T [37]	Conv + SSM	45	291	46.7	68.7	50.8	42.2	65.7	45.5
EfficientVMamba-B [58]	Conv + SSM	53	252	43.7	66.2	47.9	40.2	63.3	42.9
VMambaV9-T [50]	Conv + SSM	50	270	47.4	69.5	52.0	42.7	66.3	46.0
PlainMamba-L1 [88]	Conv + SSM	31	388	44.1	64.8	47.9	39.1	61.6	41.9
MambaOut-Tiny	Conv	43	262	45.1	67.3	49.6	41.0	64.1	44.1
ConvNeXt-S [49]	Conv	70	348	45.4	67.9	50.0	41.8	65.2	45.1
FocalNet-S [89]	Conv	72	365	48.3	70.5	53.1	43.1	67.4	46.2
Swin-S [51]	Attn	69	354	44.8	66.6	48.9	40.9	63.2	44.2
CSWin-S [22]	Attn	54	342	47.9	70.1	52.6	43.2	67.1	46.2
PVTv2-B3 [80]	Conv + Attn	65	397	47.0	68.1	51.7	42.5	65.7	45.7
SG-Former-M [65]	Conv + Attn	51	—	48.2	70.3	53.1	43.6	66.9	47.0
TransNeXt-Small [69]	Conv + Attn	69	516	51.1	72.6	56.2	45.5	69.8	49.1
VMamba-S [50]	Conv + SSM	64	400	48.2	69.7	52.5	43.0	66.6	46.4
LocalVMamba-S [37]	Conv + SSM	69	414	48.4	69.9	52.7	43.2	66.7	46.5
VMambaV9-S [50]	Conv + SSM	64	357	48.7	70.0	53.4	43.7	67.3	47.0
MambaOut-Small	Conv	65	354	47.4	69.1	52.4	42.7	66.1	46.2
ConvNeXt-B [49]	Conv	108	486	47.0	69.4	51.7	42.7	66.3	46.0
FocalNet-B [89]	Conv	111	507	49.0	70.9	53.9	43.5	67.9	46.7
Swin-B [51]	Attn	107	496	46.9	—	—	42.3	—	—
ViT-Adapter-B [10]	Attn	102	557	47.0	68.2	51.4	41.8	65.1	44.9
CSWin-B [22]	Attn	97	526	48.7	70.4	53.9	43.9	67.8	47.3
PVTv2-B5 [80]	Conv + Attn	102	557	47.4	68.6	51.9	42.5	65.7	46.0
TransNeXt-Base [69]	Conv + Attn	109	728	51.7	73.2	56.9	45.9	70.5	49.7
VMamba-B [50]	Conv + SSM	96	540	48.5	69.6	53.0	43.1	67.0	46.4
PlainMamba-L2 [88]	Conv + SSM	53	542	46.0	66.9	50.1	40.6	63.8	43.6
VMambaV9-B [50]	Conv + SSM	108	485	49.2	70.9	53.9	43.9	67.7	47.6
MambaOut-Base	Conv	100	495	47.4	69.3	52.2	43.0	66.4	46.3

Hypothesis 1, it will be necessary to develop visual Mamba models with token mixers of convolution and SSM to achieve state-of-the-art performance on ImageNet.

4.3 Object detection & instance segmentation on COCO

Setup. COCO 2017 [48] serves as a widely recognized benchmark for object detection and instance segmentation. In our experiments, MambaOut is employed as the backbone within Mask R-CNN [31], initialized with weights pre-trained on ImageNet. We adhere to the standard $1 \times$ training schedule of 12 epochs. The training images are resized such that the shorter side measures 800 pixels, while the longer side does not exceed 1333 pixels. The AdamW optimizer [53, 41] is used with a learning rate of 0.0001 and a total batch size of 16. Our implementation leverages the PyTorch [56] and mmdetection [8] libraries. We utilize FP16 precision to save training costs. The experiments are conducted on 4 GPUs of NVIDIA 4090.

Results. Although MambaOut can surpass some visual Mamba models [58, 88] in object detection and instance segmentation on COCO [48], it still lags behind the state-of-the-art visual Mambas, such as VMamba [50] and LocalVMamba [50]. For instance, the performance of MambaOut-Tiny as the backbone for Mask R-CNN trails VMamba-T [50] by 1.4 AP^b and 1.1 AP^m. This performance disparity underscores the benefits of integrating Mamba in long-sequence visual tasks, reinforcing our *Hypothesis 2*. However, visual Mamba still exhibits a significant performance gap when compared to the state-of-the-art convolution-attention-hybrid models, TransNeXt [69]. Visual Mamba needs to further validate its effectiveness by outperforming other state-of-the-art models in the visual detection task.

Table 3: **Performance of Semantic segmentation with UperNet [85] on ADE20K [103] validation set.** The MACs are measured with input size of 512×2048 .

Backbone	Token Mixing Type	UperNet			
		Param (M)	MAC (G)	mIoU (SS)	mIoU (MS)
ConvNeXt-T [49]	Conv	60	939	46.0	46.7
HorNet-T [64]	Conv	55	924	49.2	49.3
ConvFormer-S18 [92]	Conv	54	925	47.5	48.6
InternImage-T [79]	Conv	59	944	47.9	48.1
Swin-T [51]	Attn	60	945	44.4	45.8
Twins-S [13]	Attn	54	901	46.2	47.1
Focal-T [90]	Attn	62	998	45.8	47.0
CSWin-T [22]	Attn	60	959	49.3	50.7
UniFormer-S [43]	Conv + Attn	52	955	47.0	48.5
CAFormer-S18 [92]	Conv + Attn	54	1024	48.1	48.9
SG-Former-S [65]	Conv + Attn	53	989	49.9	51.5
TransNeXt-Tiny [69]	Conv + Attn	59	978	51.1	51.2
VMamba-T [50]	Conv + SSM	55	964	47.3	48.3
LocalVMamba-T [37]	Conv + SSM	57	970	47.9	49.1
EfficientVMamba-B [58]	Conv + SSM	65	930	46.5	47.3
PlainMamba-L2 [88]	Conv + SSM	55	285	–	46.8
PlainMamba-L3 [88]	Conv + SSM	81	419	–	49.1
VMambaV9-T [50]	Conv + SSM	62	948	48.3	48.6
MambaOut-Tiny	Conv	54	938	47.4	48.6
ConvNeXt-S [49]	Conv	82	1027	48.7	49.6
HorNet-S [64]	Conv	85	1027	50.0	50.5
ConvFormer-S36 [92]	Conv	67	1003	49.6	50.7
InternImage-S [79]	Conv	80	1017	50.1	50.9
Swin-S [51]	Attn	81	1038	47.6	49.5
Twins-B [13]	Attn	89	1020	47.7	48.9
Focal-S [90]	Attn	85	1130	48.0	50.0
CSWin-S [22]	Attn	65	1027	50.4	51.5
CAFormer-S36 [92]	Conv + Attn	67	1197	50.6	50.8
SG-Former-M [65]	Conv + Attn	68	1114	51.2	52.1
TransNeXt-Small [69]	Conv + Attn	80	1089	52.2	52.3
VMamba-S [50]	Conv + SSM	76	1081	49.5	50.5
LocalVMamba-S [37]	Conv + SSM	81	1095	50.0	51.0
VMambaV9-S [50]	Conv + SSM	82	1039	50.6	51.2
MambaOut-Small	Conv	76	1032	49.5	50.6
ConvNeXt-B [49]	Conv	122	1170	49.1	49.9
HorNet-B [64]	Conv	126	1171	50.5	50.9
ConvFormer-M36 [92]	Conv	85	1113	50.4	51.3
InternImage-B [79]	Conv	128	1185	50.8	51.3
Swin-B [51]	Attn	121	1188	48.1	49.7
Twins-L [13]	Attn	133	1164	48.8	50.2
Focal-B [90]	Attn	126	1354	49.0	50.5
CSWin-B [22]	Attn	110	1222	51.1	52.2
UniFormer-B [43]	Conv + Attn	80	1106	49.5	50.7
CAFormer-M36 [92]	Conv + Attn	84	1346	51.7	51.7
SG-Former-B [65]	Conv + Attn	109	1304	52.0	52.7
TransNeXt-Base [69]	Conv + Attn	121	1268	53.0	53.4
VMamba-B [50]	Conv + SSM	110	1226	50.0	51.3
VMambaV9-B [50]	Conv + SSM	122	1170	51.0	51.6
MambaOut-Base	Conv	112	1178	49.6	51.0

4.4 Semantic segmentation on ADE20K

Setup. ADE20K [103], a widely-used benchmark for the semantic segmentation task, encompasses 150 semantic categories. It includes 20,000 images in the training set and 2,000 images in the validation set. In our experiments, Mamba is employed as the backbone for UperNet [85], with initialization from ImageNet pre-trained weights. The training is conducted using the AdamW optimizer [41, 53] with learning rate of 0.0001 and batch size of 16 for 160,000 iterations. Our implementation utilizes the PyTorch [56] and mmsegmentation [14] libraries. Experiments are performed on four GPUs of NVIDIA 4090, with FP16 precision to enhance the training speed.

Results. The performance trend for semantic segmentation on ADE20K is similar to object detection on COCO. MambaOut can outperform some visual Mamba models but cannot match the results of state-of-the-art Mamba models. For instance, LocalVMamba-T [37] surpasses MambaOut-Tiny by 0.5 mIoU in both single scale (SS) and multi-scale (MS) evaluations, further corroborating our *Hypothesis 2* empirically. Additionally, visual Mamba models continue to exhibit notable performance deficits when compared to the more advanced hybrid models that integrate convolution and attention mechanisms, such as SG-Former [65] and TransNeXt [69]. Visual Mamba needs to further showcase its long-sequence modeling strengths by delivering stronger performance in visual segmentation task.

5 Conclusion

In this paper, we discuss the Mamba mechanism conceptually and conclude that it is ideally suited for tasks with long-sequence and autoregressive characteristics. We analyze common visual tasks against these criteria and argue that introducing Mamba for ImageNet image classification is unnecessary, as it meets neither characteristic. However, the potential of Mamba for visual detection and segmentation tasks, which align with at least the long-sequence characteristic, merits further exploration. To substantiate our claims empirically, we develop MambaOut models that employ Mamba blocks without their core token mixer, SSM. MambaOut surpasses all visual Mamba models on ImageNet, yet it exhibits a notable performance gap compared to state-of-the-art visual Mamba models, thereby validating our assertions. Due to computational resource limitations, this paper only verifies the Mamba concept for visual tasks. In the future, we may further explore Mamba and RNN concepts as well as the integration of RNN and Transformer for large language models (LLMs) and large multimodal models (LMMs).

Acknowledgement

Weihao was partly supported by Snap Research Fellowship, Google TPU Research Cloud (TRC), and Google Cloud Research Credits program. We thank Dongze Lian, Qiuhong Shen, Xingyi Yang, and Gongfan Fang for valuable discussions.

References

- [1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.
- [2] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- [3] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.
- [4] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. In *International Conference on Learning Representations*, 2021.
- [5] Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.
- [6] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.
- [7] Jierun Chen, Shiu-hong Kao, Hao He, Weipeng Zhuo, Song Wen, Chul-Ho Lee, and S-H Gary Chan. Run, don’t walk: Chasing higher flops for faster neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12021–12031, 2023.
- [8] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.
- [9] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya Sutskever. Generative pretraining from pixels. In *International conference on machine learning*, pages 1691–1703. PMLR, 2020.
- [10] Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. Vision transformer adapter for dense predictions. In *The Eleventh International Conference on Learning Representations*, 2022.
- [11] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017.

- [12] Krzysztof Choromanski, Valerii Likhoshesterov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Davis, David Belanger, Lucy Colwell, et al. Masked language modeling for proteins via linearly scalable long-context transformers. *arXiv preprint arXiv:2006.03555*, 2020.
- [13] 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.
- [14] MMSegmentation Contributors. MMSegmentation: Openmmlab semantic segmentation toolbox and benchmark. <https://github.com/open-mmlab/mms Segmentation>, 2020.
- [15] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020.
- [16] 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.
- [17] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime G Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, 2019.
- [18] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. Language modeling with gated convolutional networks. In *International conference on machine learning*, pages 933–941. PMLR, 2017.
- [19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [20] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [21] 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.
- [22] 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 *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12124–12134, 2022.
- [23] 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. *arXiv preprint arXiv:2010.11929*, 2020.
- [24] Kunihiko Fukushima. Visual feature extraction by a multilayered network of analog threshold elements. *IEEE Transactions on Systems Science and Cybernetics*, 5(4):322–333, 1969.
- [25] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.
- [26] Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.
- [27] Albert Gu, Isys Johnson, Karan Goel, Khaled Saab, Tri Dao, Atri Rudra, and Christopher Ré. Combining recurrent, convolutional, and continuous-time models with linear state space layers. *Advances in neural information processing systems*, 34:572–585, 2021.
- [28] Meng-Hao Guo, Cheng-Ze Lu, Zheng-Ning Liu, Ming-Ming Cheng, and Shi-Min Hu. Visual attention network. *Computational Visual Media*, 9(4):733–752, 2023.
- [29] 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.
- [30] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16000–16009, 2022.

- [31] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.
- [32] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [33] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.
- [34] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [35] Qibin Hou, Cheng-Ze Lu, Ming-Ming Cheng, and Jiashi Feng. Conv2former: A simple transformer-style convnet for visual recognition. *arXiv preprint arXiv:2211.11943*, 2022.
- [36] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14*, pages 646–661. Springer, 2016.
- [37] 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.
- [38] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.
- [39] Zi-Hang Jiang, Weihao Yu, Daquan Zhou, Yunpeng Chen, Jiashi Feng, and Shuicheng Yan. Conybert: Improving bert with span-based dynamic convolution. *Advances in Neural Information Processing Systems*, 33:12837–12848, 2020.
- [40] 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.
- [41] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [42] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 2012.
- [43] Kunchang Li, Yali Wang, Junhao Zhang, Peng Gao, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unifying convolution and self-attention for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.
- [44] Shufan Li, Harkanwar Singh, and Aditya Grover. Mamba-nd: Selective state space modeling for multi-dimensional data. *arXiv preprint arXiv:2402.05892*, 2024.
- [45] Siyuan Li, Zedong Wang, Zicheng Liu, Cheng Tan, Haitao Lin, Di Wu, Zhiyuan Chen, Jiangbin Zheng, and Stan Z. Li. Moganet: Multi-order gated aggregation network. In *The Twelfth International Conference on Learning Representations*, 2024.
- [46] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4804–4814, 2022.
- [47] Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, et al. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*, 2024.
- [48] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6–12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014.
- [49] Shiwei Liu, Tianlong Chen, Xiaohan Chen, Xuxi Chen, Qiao Xiao, Boqian Wu, Tommi Kärkkäinen, Mykola Pechenizkiy, Decebal Mocanu, and Zhangyang Wang. More convnets in the 2020s: Scaling up kernels beyond 51x51 using sparsity. *arXiv preprint arXiv:2207.03620*, 2022.

- [50] 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.
- [51] 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 *Proceedings of the IEEE/CVF international conference on computer vision*, pages 10012–10022, 2021.
- [52] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986, 2022.
- [53] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.
- [54] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)*, pages 116–131, 2018.
- [55] Franck Mamalet and Christophe Garcia. Simplifying convnets for fast learning. In *International Conference on Artificial Neural Networks*, pages 58–65. Springer, 2012.
- [56] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.
- [57] Badri N Patro and Vijay S Agneeswaran. Simba: Simplified mamba-based architecture for vision and multivariate time series. *arXiv preprint arXiv:2403.15360*, 2024.
- [58] Xiaohuan Pei, Tao Huang, and Chang Xu. Efficientvmamba: Atrous selective scan for light weight visual mamba. *arXiv preprint arXiv:2403.09977*, 2024.
- [59] Bo Peng, Eric Alcaide, Quentin Anthony, Alon Albalak, Samuel Arcadinho, Huanqi Cao, Xin Cheng, Michael Chung, Matteo Grella, Kranthi Kiran GV, et al. Rwkv: Reinventing rnns for the transformer era. *arXiv preprint arXiv:2305.13048*, 2023.
- [60] Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. 2018.
- [61] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.
- [62] Jack W Rae, Anna Potapenko, Siddhant M Jayakumar, and Timothy P Lillicrap. Compressive transformers for long-range sequence modelling. *arXiv preprint arXiv:1911.05507*, 2019.
- [63] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of machine learning research*, 21(140):1–67, 2020.
- [64] Yongming Rao, Wenliang Zhao, Yansong Tang, Jie Zhou, Ser Nam Lim, and Jiwen Lu. HorNet: Efficient high-order spatial interactions with recursive gated convolutions. *Advances in Neural Information Processing Systems*, 35:10353–10366, 2022.
- [65] Sucheng Ren, Xingyi Yang, Songhua Liu, and Xinchao Wang. Sg-former: Self-guided transformer with evolving token reallocation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6003–6014, 2023.
- [66] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.
- [67] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.
- [68] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681, 1997.
- [69] Dai Shi. Transnext: Robust foveal visual perception for vision transformers. *arXiv preprint arXiv:2311.17132*, 2023.

- [70] Chenyang Si, Weihao Yu, Pan Zhou, Yichen Zhou, Xinchao Wang, and Shuicheng Yan. Inception transformer. *Advances in Neural Information Processing Systems*, 35:23495–23509, 2022.
- [71] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2818–2826, 2016.
- [72] Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. Efficient transformers: A survey. *ACM Computing Surveys*, 55(6):1–28, 2022.
- [73] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *International conference on machine learning*, pages 10347–10357. PMLR, 2021.
- [74] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 32–42, 2021.
- [75] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxvit: Multi-axis vision transformer. In *European conference on computer vision*, pages 459–479. Springer, 2022.
- [76] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [77] Shakti N Wadekar and Abhishek Chaurasia. Mobilevitv3: Mobile-friendly vision transformer with simple and effective fusion of local, global and input features. *arXiv preprint arXiv:2209.15159*, 2022.
- [78] Sinong Wang, Belinda Z Li, Madian Khabsa, Han Fang, and Hao Ma. Linformer: Self-attention with linear complexity. *arXiv preprint arXiv:2006.04768*, 2020.
- [79] Wenhai Wang, Jifeng Dai, Zhe Chen, Zhenhang Huang, Zhiqi Li, Xizhou Zhu, Xiaowei Hu, Tong Lu, Lewei Lu, Hongsheng Li, et al. Internimage: Exploring large-scale vision foundation models with deformable convolutions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14408–14419, 2023.
- [80] 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*, 8(3):415–424, 2022.
- [81] Ross Wightman. Pytorch image models. <https://github.com/rwightman/pytorch-image-models>, 2019.
- [82] Felix Wu, Angela Fan, Alexei Baevski, Yann N Dauphin, and Michael Auli. Pay less attention with lightweight and dynamic convolutions. *arXiv preprint arXiv:1901.10430*, 2019.
- [83] Yuxin Wu and Kaiming He. Group normalization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.
- [84] Zhanghao Wu, Zhijian Liu, Ji Lin, Yujun Lin, and Song Han. Lite transformer with long-short range attention. *arXiv preprint arXiv:2004.11886*, 2020.
- [85] Tete Xiao, Yingcheng Liu, Bolei Zhou, Yuning Jiang, and Jian Sun. Unified perceptual parsing for scene understanding. In *Proceedings of the European conference on computer vision (ECCV)*, pages 418–434, 2018.
- [86] Rui Xu, Shu Yang, Yihui Wang, Bo Du, and Hao Chen. A survey on vision mamba: Models, applications and challenges. *arXiv preprint arXiv:2404.18861*, 2024.
- [87] Chenglin Yang, Siyuan Qiao, Qihang Yu, Xiaoding Yuan, Yukun Zhu, Alan Yuille, Hartwig Adam, and Liang-Chieh Chen. MOAT: Alternating mobile convolution and attention brings strong vision models. In *The Eleventh International Conference on Learning Representations*, 2023.
- [88] 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.

- [89] Jianwei Yang, Chunyuan Li, Xiyang Dai, and Jianfeng Gao. Focal modulation networks. *Advances in Neural Information Processing Systems*, 35:4203–4217, 2022.
- [90] Jianwei Yang, Chunyuan Li, Pengchuan Zhang, Xiyang Dai, Bin Xiao, Lu Yuan, and Jianfeng Gao. Focal attention for long-range interactions in vision transformers. *Advances in Neural Information Processing Systems*, 34:30008–30022, 2021.
- [91] Weihao Yu, Mi Luo, Pan Zhou, Chenyang Si, Yichen Zhou, Xinchao Wang, Jiashi Feng, and Shuicheng Yan. Metaformer is actually what you need for vision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10819–10829, 2022.
- [92] Weihao Yu, Chenyang Si, Pan Zhou, Mi Luo, Yichen Zhou, Jiashi Feng, Shuicheng Yan, and Xinchao Wang. Metaformer baselines for vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.
- [93] Weihao Yu, Pan Zhou, Shuicheng Yan, and Xinchao Wang. Inceptionnext: When inception meets convnext. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024.
- [94] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis EH 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*, pages 558–567, 2021.
- [95] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 45(5):6575–6586, 2022.
- [96] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019.
- [97] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.
- [98] Shuangfei Zhai, Walter Talbott, Nitish Srivastava, Chen Huang, Hanlin Goh, Ruixiang Zhang, and Josh Susskind. An attention free transformer. *arXiv preprint arXiv:2105.14103*, 2021.
- [99] Hanwei Zhang, Ying Zhu, Dan Wang, Lijun Zhang, Tianxiang Chen, and Zi Ye. A survey on visual mamba. *arXiv preprint arXiv:2404.15956*, 2024.
- [100] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations*, 2018.
- [101] Jiangning Zhang, Xiangtai Li, Jian Li, Liang Liu, Zhucun Xue, Boshen Zhang, Zhengkai Jiang, Tianxin Huang, Yabiao Wang, and Chengjie Wang. Rethinking mobile block for efficient attention-based models. In *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 1389–1400. IEEE Computer Society, 2023.
- [102] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 13001–13008, 2020.
- [103] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017.
- [104] 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.

A More details of MambaOut models

The MambaOut model configurations are shown in Table 4 and the hyper-parameters to train MambaOut on ImageNet are shown in Table 5.

Table 4: Configurations of MambaOut models. The contents in the tuples represent the configurations in the four stages of the models.

Size	Femto	Tiny	Small	Base
Stem	3×3 conv with stride 2; Norm; GELU; 3×3 conv with stride 2, Norm			
Downsampling layers	3×3 conv with stride 2			
Token mixer	7×7 depthwise conv			
MLP ratio	8/3			
Classifier head	Global average pooling, Norm, MLP			
# Blocks	(3, 3, 9, 3)	(3, 3, 9, 3)	(3, 4, 27, 3)	(3, 4, 27, 3)
# Channel	(48, 96, 192, 288)	(96, 192, 384, 576)	(96, 192, 384, 576)	(128, 256, 512, 768)
Parameters (M)	7.3	26.5	48.5	84.8
MACs (G)	1.2	4.5	9.0	15.8

Table 5: Hyper-parameters of MambaOut on ImageNet image classification.

	MambaOut			
	Femto	Tiny	Small	Base
Input resolution	224 ²			
Epochs	300			
Batch size	4096			
Optimizer	AdamW			
Adam ϵ	1e-8			
Adam (β_1, β_2)	(0.9, 0.999)			
Learning rate	4e-3			
Learning rate decay	Cosine			
Gradient clipping	None			
Warmup epochs	20			
Weight decay	0.05			
Rand Augment	9/0.5			
Repeated Augmentation	off			
Cutmix	1.0			
Mixup	0.8			
Cutmix-Mixup switch prob	0.5			
Random erasing prob	0.25			
Label smoothing	0.1			
Peak stochastic depth rate	0.025	0.2	0.4	0.6
Random erasing prob	0.25			
EMA decay rate	None			