V“Mean”ba: Visual State Space Models only need 1 hidden dimension

TienYu Chi
National Yang Ming Chiao Tung University
b03902059@ntu.edu.tw

Hung-Yueh Chiang
The University of Texas at Austin
hungyueh.chiang@utexas.edu

Chi-Chih Chang
Cornell University
cc2869@cornell.edu

Ning-Chi Huang
National Yang Ming Chiao Tung University
nchuang@cs.nycu.edu.tw

Kai-Chiang Wu
National Yang Ming Chiao Tung University
kcw@cs.nctu.edu.tw

Abstract

Vision transformers dominate image processing tasks due to their superior performance. However, the quadratic complexity of self-attention limits the scalability of these systems and their deployment on resource-constrained devices. State Space Models (SSMs) have emerged as a solution by introducing a linear recurrence mechanism, which reduces the complexity of sequence modeling from quadratic to linear. Recently, SSMs have been extended to high-resolution vision tasks. Nonetheless, the linear recurrence mechanism struggles to fully utilize matrix multiplication units on modern hardware, resulting in a computational bottleneck. We address this issue by introducing *VMeanba*, a training-free compression method that eliminates the channel dimension in SSMs using mean operations. Our key observation is that the output activations of SSM blocks exhibit low variances across channels. Our *VMeanba* leverages this property to optimize computation by averaging activation maps across the channel to reduce the computational overhead without compromising accuracy. Evaluations on image classification and semantic segmentation tasks demonstrate that *VMeanba* achieves up to a 1.12x speedup with less than a 3% accuracy loss. When combined with 40% unstructured pruning, the accuracy drop remains under 3%.

1 Introduction

Computer vision has advanced significantly due to deep learning and the availability of large-scale datasets. Convolutional Neural Networks (CNNs) have become foundational for tasks such as image classification [11, 22, 9] and object detection [6, 5, 21]. However, CNNs struggle to capture long-range dependencies. Vision Transformers (ViTs) [3, 18, 24] which utilize self-attention mechanisms, effectively address this limitation but suffer from high computational costs due to quadratic complexity. To mitigate these costs, research has focused on reducing ViT complexity [25, 1, 18, 17, 15], applying model compression techniques [19, 14, 30, 27, 24, 13], and exploring alternative architectures like RWKV and State Space Models (SSMs) [20, 8, 4, 7].

State Space Models (SSMs) have recently garnered attention in computer vision as efficient and effective alternatives to Vision Transformers (ViTs), demonstrating competitive performance across

Machine Learning for Systems Workshop at (NeurIPS 2024).

various tasks [16, 29, 12, 23]. For example, VMamba [16] achievesd 82.6% top-1 accuracy on ImageNet-1k [2], surpassing Swin Transformer [18] by 1.3% with comparable FLOPs. However, despite reducing computational complexity, SSMs still fail to fully utilize matrix multiplication units on GPUs, creating a bottleneck in vision-based SSM models.

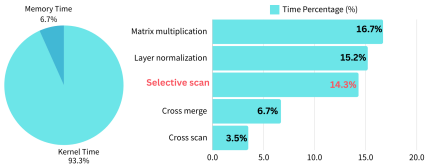

Figure 1: The GPU kernel time of each operation in a VMamba block. The latency is measured using feature maps with an input resolution of 224×224. We rank the kernels by their latency and highlights the top-5 time-consuming kernels on the bar chart. The selective scan operation is one of the major contributors in the VMamba block.

To this end, we first analyze the latency breakdown of VMamba [16] and identify the selective scan operation [7] as one of the key bottlenecks in inference. Figure 1 shows that the selective scan accounts for 14.3% of the total kernel time in a VMamba block. While optimizing selective scan operation is critical for enhancing SSM efficiency, few research works address this problem and optimizing the efficiency of SSMs remains unexplored.

In this paper, we propose *VMeanba*, a novel activation compression method designed to optimize the selective scan operation in VMamba blocks. The high-level overview of *VMeanba* is presented in Figure 2. The key idea is to reduce the input tensor's channel dimensions in the associate scan operation by applying a mean operation. Through analysis of the weight and activation distributions in the trained VMamba model, we identified a smooth pattern with small variances that allows for dimensional reduction. Based on this observation, we developed the *VMeanba* block to exploit this pattern, resulting in a more efficient associate scan operation without compromising accuracy. Experimental results demonstrate that *VMeanba* achieves up to a 1.12x speedup with less than a 3% accuracy loss. To the best of our knowledge, this is the first work optimizing of the selective scan operation in VMamba.

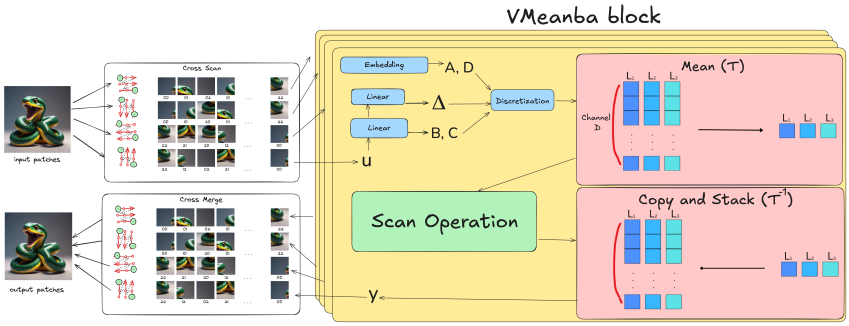

Figure 2: Overview of the *VMeanba* block. *VMeanba* reduced the channel dimension of the inputs to the associated scan operation by applying a transform T, thereby simplifying the computation. The proposed *VMeanba* components are highlighted in red, while the original selective scan components are shown in blue and green, with the green block indicating the main area of optimization.

2 Methods

2.1 Distribution Analysis of VMamba

We conduct an in-depth investigation into the characteristics of each layer's output within the Mamba block of VMamba. The output is denoted as $y_{layer} \in \mathbb{R}^{B \times D \times L}$, where B is the batch size, D is the inner channel dimension utilized by the scan algorithm within the Mamba block, and L is 4x of the feature map size HW. Our analysis revealed that for each y_{layer}, the distribution of values across the inner channel dimension is remarkably consistent across different data points, as illustrated in figure 3. This observation raised a critical question: Is the full dimensionality D necessary for each y_{layer}?

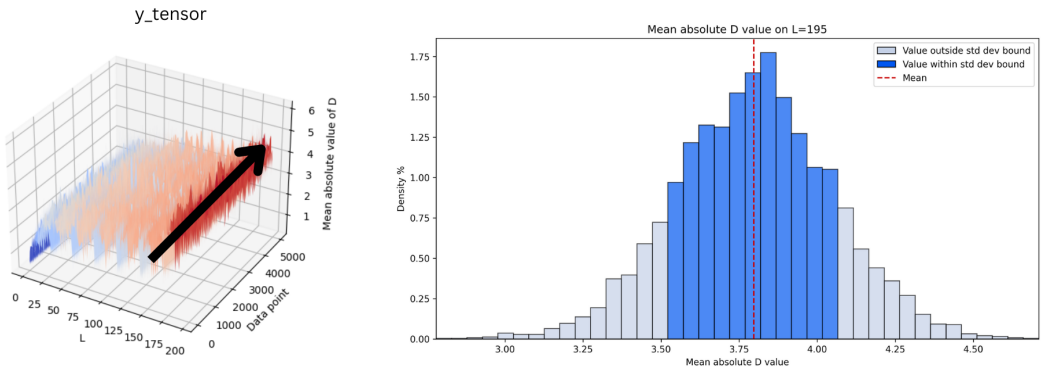

Figure 3: The figure illustrates the distribution of inner dimension values of y_{layer} across various data points as a function of sequence length. Notably, the distribution remains remarkably consistent across different data points for identical l values, as indicated by the arrows. The distribution for $l = 195$, shown on the right, provides further evidence of this concentration.

To explore this, we hypothesized a property described by equation (1):

$$y_{\text{layer}}[:, d, :] \sim y_{layer}[:, d', :] \quad \forall d, d' \in [1, D], d \neq d' \tag{1}$$

Given that the scan algorithm in the Mamba block performs a linear transformation, this unique property of y_{layer} can be attributed to the inputs $\bar{A}$, $\bar{B}u_t$ and C to the SSM system. Consequently, we propose that a reduced set of inputs $(\bar{A}_{reduce}, (\bar{B}u_t)_{reduce}, C_{reduce})$, referred to collectively as I_{basis}, can effectively represent the original inputs $(\bar{A}, \bar{B}u_t, C)$. By leveraging these reduced inputs, we can optimize the computational efficiency of each Mamba block.

2.2 VMeanba

Building on the findings from section 2.1, we indroduce a new model inference efficiency optimization method called *VMeanba*, which computes I_{basis} for each Mamba block using mean operators. We further design a pipeline to select which layers in the model will undergo this optimization.

VMeanba block. The I_{basis} is derived by having a transform function T that maps the original inputs $(\bar{A}, \bar{B}u_t, C)$ to reduced dimension inputs. After processing by the original Mamba block, the output is recovered using an inverse transform function T^{-1}. This entire process can be expressed as equation (2).

$$y_{layer} = T^{-1}(Mamba(T(\bar{A}, \bar{B}u_t, C))) \tag{2}$$

In this process, T is defined as the mean operator applied along the inner channel dimension axis, and T^{-1} is defined as the broadcast operator. While the mean transform may lead to a loss of information, it significantly reduces the dimensionality of the inputs from D to 1, with our experiments demonstrating that model performance is maintained. The computational complexity analysis is provided in B.

Layer Selection. We developed a pipeline to replace K Mamba blocks with *VMeanba* blocks. We treat the choices of layers as a hyperparameter, determined using the validation set. Specifically, we calculate the layer impact score S_{layer} for each layer, and select the layers with the K smallest scores to apply the VMeanba optimization. The impact score is defined by equation (3):

$$S_{layer} = Acc(OriginalModel) - Acc(VMeanba\ on\ layer) \tag{3}$$

where Acc represents the model accuracy on the validation set. The algorithm for this process is detailed in C.

3 Experiments

We apply the proposed *VMeanba* method to two different tasks: image classification and semantic segmentation. The details experiment setup and more experiments are provided in appendix D, E

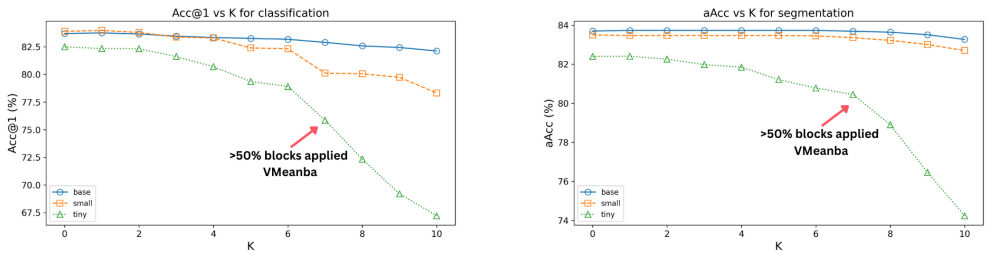

Figure 4: Accuracy versus K Analysis on classification and segmentation tasks by using *VMeanba*. This figure illustrates the trade-off between the value of K and the associated accuracy drop. By carefully selecting an appropriate K, the model's accuracy can be largely preserved.

3.1 Results on image classification and semantic segmentation

Accuracy versus K Analysis. We applied the *VMeanba* method to VMamba backbone models for both image classification and semantic segmentation tasks, varying the parameter K, as shown in Figure 4. Our results indicate that the model accuracy remains largely unaffected when an appropriate K value is chosen. However, there is a trade-off exists: increasing K reduces inference time but leads to a more pronounced accuracy decline, as indicated by the arrows in the figure. Striking an optimal balance between accuracy and K is essential. For example, selecting $K = 10$ for the `base` model in image classification and semantic segmentation appears reasonable. In cases where accuracy drop is deemed unacceptable, one could still opt for a larger K and retrain the model to recover performance. Since this study focuses on a training-free approach, retraining strategies are left for future work.

3.2 Combined with Other Optimization Techniques

We demonstrated that our *VMeanba* method can be seamlessly integrated with other optimization techniques to enhance model efficiency. Specifically, we explored the effectiveness of combining *VMeanba* with unstructured pruning on the VMamba `base` model for the image classification task using value $K = 8$. The results are summarized in Table 1. Pruning was applied to weight of linear layer or convolution 2D layer using the $l1$ norm, with a consistent pruning ratio of 40%. Our findings indicate that the *VMeanba* method is orthogonal to pruning, as it enhances efficiency while maintaining comparable accuracy, demonstrating that the two techniques can be combined without interference.

Pruning Target	K	Acc@1
Linear Layers	0	83.5%
	8	81.6%
Conv2D Layers	0	80.1%
	8	77.5%

Table 1: Accuracy comparison of *VMeanba* with pruning on Linear and Conv2D layers using the `base` backbone.

4 Conclusion

In this work, we introduced `VMeanba`, a novel, training-free model compression technique that reduces the inference time of the Mamba block in VMamba by applying a mean operation to reduce the dimensionality of input channel tensors in the associate scan operation. Our experimental results demonstrate that `VMeanba` enhances inference speed and throughput while maintaining competitive accuracy in VMamba.

This work contributes to the field by introducing a practical method for improving VMamba's efficiency and suggests future exploration of the dimensionality of input channel tensors and the kernel fusion of the discretization and selective scan operations to improve GPU utilization. Additionally, we envision extending *VMeanba* to other computer vision tasks to evaluate its broader applicability and scalability.

A Preliminaries

In this section, we introduce some preliminaries of the State Space Model, SSM [10], and two recently proposed methods using SSM, mainly selective state space model (Mamba)[7] and VMamba[16]

State Space Model (SSM). The SSM is a mathematical model that represents the evolution of a system over time. The model is specified as a set of equations that relate the state of the system to the observations at each time step. The most general form of the SSM is called continuous-time linear dynamical system, which is defined as equation (4).

$$\begin{aligned} h'(t) &= A(t)h(t) + B(t)u(t) \\ y(t) &= C(t)h(t) + D(t)u(t) \end{aligned} \tag{4}$$

$h(t) \in \mathbb{R}^n$ is the state variable at time step $t \in \mathbb{R}$, or usually called hidden variable in recent machine learning literature, $u(t) \in \mathbb{R}^m$ is the input, $y(t) \in \mathbb{R}^p$ is the output, and $A(t) \in \mathbb{R}^{n\times n}$, $B(t) \in \mathbb{R}^{n\times m}$, $C(t) \in \mathbb{R}^{p\times m}$, $D(t) \in \mathbb{R}^{p\times m}$ are the system matrices at each time step. Note that in the following context, we treat $u(t)$ and $y(t)$ as scalars, i.e., $m = p = 1$. The above continuous-time linear dynamical system can lead to a linear time-invariant (LTI) system when the system matrices $A(t)$, $B(t)$, $C(t)$, $D(t)$ are all time-invariant. This LTI SSM then can be written as equation (5). It can be discretized into a discrete-time linear dynamical system, which is defined as equation (6). One of the frequent ways for this transformation utilized in the literature related to SSM is zero-order hold (ZOH) discretization, which is defined as equation (7). Besides, it can further written as a convolution form (8).

$$\begin{aligned} h'(t) &= Ah(t) + Bu(t) \\ y(t) &= Ch(t) + Du(t) \end{aligned} \tag{5}$$

$$\begin{aligned} h_t &= \bar{A}h_{t-1} + \bar{B}u_t \\ y_t &= Ch_t + Du_t \end{aligned} \tag{6}$$

$$\begin{aligned} \bar{A} &= \exp(\Delta A) \\ \bar{B} &= (\Delta A)^{-1}\exp(\Delta A - I)\Delta B \end{aligned} \tag{7}$$

$$\begin{aligned} \bar{K} &= (C\bar{B}, C\bar{A}\bar{B}, ..., C\bar{A}^k\bar{B}, ...) \\ y &= x * \bar{K} \end{aligned} \tag{8}$$

Selective State Space Model (Mamba). Mamba is the discrete-time linear dynamical system with a timescale parameter Δ that transforms the continuous variables A, B to discrete variables $\bar{A}$, $\bar{B}$. In addition to discretization, Mamba also relax the time-invariant constraint of the system matrices by introducing *selection* mechanism, which simply makes several parameters Δ, B, C to be time-varying by functions s of the input u. Specifically defined as equation (9).

$$\begin{aligned} s_B(u) &= Linear_N(u) \\ s_C(u) &= Linear_N(u) \\ s_\Delta(u) &= Broadcast_D(Linear_1(u)) \\ \Delta &= \tau_\Delta(Parameter + s_\Delta(u)) \end{aligned} \tag{9}$$

The $Linear_d$ is a parameterized linear projection to dimension d, and the $\tau_\Delta = softplus$. As the selection mechanism loses the equivalence to convolution form (7), to avoid the sequential recurrence, Mamba further incorporates a work-efficient parallel algorithm, *associate scan*, into its GPU kernel implementation to facilitate parallel computation of the system.

VMamba The original Mamba block is designed for 1-dimensional input and output, which is not suitable for computer vision tasks. VMamba proposed a new module called 2D-Selective-Scan (SS2D) for adapting Mamba to 2D input and output. The SS2D module is composed of three steps: cross-scan, selective scan (Mamba block), and cross merge. The cross-scan unfold the input feature map along four directions, forming 4 sets of 1D sequences. Then the selective scan processes each 1D sequence in parallel. The cross-merge finally merges the 4 sets of 1D sequences back to 2D feature map. The cross-scan and cross-merge are called Cross Scan Module (CSM) together, and by this way, the model can have a global receptive field. VMamba further stack multiple SS2D blocks in a layer, and then stack layers to form the whole model.

B Computation Complexity

Complexity of SSM The computational complexity of the associated scan operation in Mamba block, measured in floating-point operations (FLOPs), is derived from processing a sequence of length L, which requires $2L$ operations. Furthermore, the input to the scan operation incurs an additional cost of 3 FLOPs, leading to a total of $3 \times 2BLD$, where B is the batch size, L is the sequence length, and D is the inner dimension.

In the context of the SSM system, computations involve multiplications for $\bar{B}u_t$ and Ch_t, which amount to $2BLD$, and additions for Ch_t and D, totaling BLD FLOPs. Consequently, the overall FLOPs for the SSM system is $3BLD$. The total FLOPs for the Mamba block, therefore, aggregate to $3 \times 2BLD + 3BLD$.

Complexity of reduced SSM The reduction in FLOPs can be achieved by employing the I_{basis}, which consists of $9BLd$ FLOPs, and additional FLOPs for the reduce operation and broadcast operation. The total reduction in FLOPs is summarized by the equation (10):

$$\begin{aligned} FLOP_{Mamba} &= FLOP_{scan} + FLOP_{SSM} \\ FLOP_{original} &= 3 \times 2BLD + 3BLD \\ FLOP_{reduction} &= 9BLd + FLOP_{reduce\_op} \\ &\quad + FLOP_{broadcast} \end{aligned} \tag{10}$$

Complexity of *VMeanba* The mean operator contribute only $BLD + BL$ FLOPs, and the broadcast operator is just a memory operation. The reduced FLOPs is then $B(10 + D)L$ FLOPs, comparing to the original $9BDL$ FLOPs, we achieve 89% FLOPs reduction ($10 << D$).

C Algorithm

Algorithm 1 VMeanba Layer Selection Pipeline

Input: $Model$, D_{val}, K, $CalculateScore$
Output: $layersToApply$

$Scores \leftarrow [\,]$
for $layer$ in $Layers$ **do**
  $s \leftarrow CalculateScore(layer, Model, D_{val})$
  $Scores \leftarrow Scores + s$
end for
$Layers \leftarrow$ Sort(S_{layer})
$layersToApply \leftarrow Layers[: K]$
return $layersToApply$

D Experiments Setup

Datasets. The datasets we use for our *VMeanba* experiments are the ImageNet-1k dataset [2] for image classification and the ADE20k dataset [28] for semantic segmentation. We only use the validation set of them for the experiments. The ImageNet-1k dataset contains 50k validation images from 1k classes, and the ADE20k dataset contains 2k images for validation, with pixel-level annotations.

Models. We use the VMamba pre-trained backbone models [16] for both tasks. The backbone models is first trained on the ImageNet-1k training dataset. It is then used as the pre-trained backbone models for downstream task. The segmentation task use the UperNet [26] on top of the VMamba pre-trained backbone models, and trained on the ADE20k training dataset. The VMamba backbone models have three different versions: *tiny*, *small*, and *base*. There are two mainly differences between these versions: the number of layers and the dimension of the L and D in the SS2D block. All of the backbone models have four layers and the *tiny* version is stack as [2, 2, 8, 2], while the other two versions are stack as [2, 2, 20, 2]. The dimension of the L and D is different across two tasks, both of

them remain the same inside each layer. However, the dimension of the D grows by a factor of 2, and the dimension of the L scale down by a factor of 4 along the layers.

Kernel Implementation. The original CUDA kernel for the Mamba block includes both the discretization and scan operations, dividing the GPU multiprocessor into a 2D grid blocks based on the batch size and inner dimension. In this configuration, multiple threads within the block handle the scan operation. However, since the discretization process is not the focus of this study, and the original approach of dividing the inner dimension across blocks is not compatible with our *VMeanba* method, we developed a new CUDA kernel. This new kernel exclusively handles the scan operation, with the discretization process executed outside the kernel. All experiments conducted in this paper are based on this optimized kernel. Future work includes integrating the discretization and scan operations into a single kernel for further optimization.

Additional Information. The evaluation metric for the image classification task is top-1 accuracy, while for the semantic segmentation task, we utilized all pixel accuracy (aAcc). The batch size for the image classification task is set to 128, whereas for the semantic segmentation task, it is limited to 1 due to the dynamic input size present in the validation set. All experiments were conducted on a single NVIDIA RTX A6000 GPU with 48GB of memory. The profiling was performed using NVTX API, Nvidia Nsight Systems, and Nvidia Nsight Compute tools.

E More Experiment results

Table 2: Speedup analysis of the *VMeanba* method compared to the original inner dimension size kernel. All VMeanba times are approximately 0.02 ms.

Backbone	Inner dimension	Sequence length	Original time (ms)	Speedup
Tiny & Small	384	3136	5.46	273x
	768	784	2.23	112x
	1536	196	1.10	55x
	3072	49	0.71	36x
Base	512	3136	5.86	293x
	1024	784	2.94	147x
	2048	196	1.47	74x
	4096	49	0.93	47x

Table 3: GPU kernel memory usage with and without the *VMeanba* method. † indicates that the original kernel memory usage is too small to be measured.

Inner dimension	Sequence length	Original memory R/W (Bytes)	Optimized memory R/W (Bytes)
512	3136	3.3G / 823.5M	6.4M / 1.3M
1024	784	1.6G / 411.9M	1.6M / 14.5K
2048	196	822.1M / 207.5M	412.4K / 5.8K
4096	49	411.1M / 107.5M	108.5K / 0†

Kernel Analysis We analyzed GPU kernel speedup and memory usage when applying *VMeanba* across varying scan sequence lengths and inner dimensions, as shown in Tables 2 and 3. Optimized kernel times, consistently around 0.02 ms, are excluded from Table 2. The *VMeanba* method achieves up to 293x speedup, particularly for longer scan sequences, aligning with the $O(DL)$ complexity discussed in B. Additionally, memory transfer between global and shared memory is significantly reduced, enabling longer scan sequences and larger batch sizes for improved throughput.

Table 4: Batch inference time comparison for the VMamba models with and without the *VMeanba* method on the image classification task.

Backbone	K	Accuracy (Acc@1 / aAcc)	Batch Inference Time (ms)	Speedup
Tiny	0	82.5%	283	1x
	2	82.3%	261	1.08x
	4	80.7%	252	1.12x
	8	72.3%	240	1.18x
Small	0	83.9%	415	1x
	2	83.8%	393	1.06x
	4	83.3%	391	1.06x
	8	80.1%	383	1.08x
Base	0	83.7%	527	1x
	2	83.7%	519	1.02x
	4	83.3%	515	1.02x
	8	82.6%	508	1.04x

Batch Inference Time Analysis. We compared batch inference times of VMamba models with and without the proposed *VMeanba* method across three backbone models on an image classification task (Table 4). The application of *VMeanba* reduced inference times, increasingly so as the value of K increased due to time savings from applying the mean operation to more layers. Notably, the `base` model exhibited less speedup compared to the `small` and `tiny` models, likely due to its larger inner dimension size incurring greater time consumption during discretization and the mean operation.

References

[1] I. Beltagy, M. E. Peters, and A. Cohan. Longformer: The long-document transformer. arXiv preprint arXiv:2004.05150, 2020.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255. Ieee, 2009.

[3] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In ICLR, 2021.

[4] D. Y. Fu, T. Dao, K. K. Saab, A. W. Thomas, A. Rudra, and C. Re. Hungry hungry hippos: Towards language modeling with state space models. In ICLR, 2022.

[5] R. Girshick. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587, 2014.

[7] A. Gu and T. Dao. Mamba: Linear-Time Sequence Modeling with Selective State Spaces, May 2024. arXiv:2312.00752 [cs].

[8] A. Gu, K. Goel, and C. Re. Efficiently modeling long sequences with structured state spaces. In ICLR, 2021.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[10] R. E. Kalman. A new approach to linear filtering and prediction problems. 1960.

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. NeurIPS, pages 1106–1114, 2012.

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

[13] H. Lin, G. Han, J. Ma, S. Huang, X. Lin, and S.-F. Chang. Supervised masked knowledge distillation for few-shot transformers. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 19649–19659, 2023.

[14] Y. Lin, T. Zhang, P. Sun, Z. Li, and S. Zhou. Fq-vit: Post-training quantization for fully quantized vision transformer. arXiv preprint arXiv:2111.13824, 2021.

[15] X. Liu, H. Peng, N. Zheng, Y. Yang, H. Hu, and Y. Yuan. Efficientvit: Memory efficient vision transformer with cascaded group attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 14420–14430, 2023.

[16] Y. Liu, Y. Tian, Y. Zhao, H. Yu, L. Xie, Y. Wang, Q. Ye, and Y. Liu. VMamba: Visual State Space Model, Apr. 2024. arXiv:2401.10166 [cs].

[17] Z. Liu, H. Hu, Y. Lin, Z. Yao, Z. Xie, Y. Wei, J. Ning, Y. Cao, Z. Zhang, L. Dong, et al. Swin transformer v2: Scaling up capacity and resolution. In CVPR, pages 12009–12019, 2022.

[18] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In ICCV, pages 10012–10022, 2021.

[19] Z. Liu, Y. Wang, K. Han, W. Zhang, S. Ma, and W. Gao. Post-training quantization for vision transformer. Advances in Neural Information Processing Systems, 34:28092–28103, 2021.

[20] B. Peng, E. Alcaide, Q. Anthony, A. Albalak, S. Arcadinho, H. Cao, X. Cheng, M. Chung, M. Grella, K. K. GV, et al. RWKV: reinventing rnns for the transformer era. In EMNLP, pages 14048–14077, 2023.

[21] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[23] Y. Teng, Y. Wu, H. Shi, X. Ning, G. Dai, Y. Wang, Z. Li, and X. Liu. Dim: Diffusion mamba for efficient high-resolution image synthesis. arXiv preprint arXiv:2405.14224, 2024.

[24] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou. Training data-efficient image transformers & distillation through attention. In ICML, pages 10347–10357, 2021.

[25] S. Wang, B. Z. Li, M. Khabsa, H. Fang, and H. Ma. Linformer: Self-attention with linear complexity. arXiv preprint arXiv:2006.04768, 2020.

[26] T. Xiao, Y. Liu, B. Zhou, Y. Jiang, and J. Sun. Unified perceptual parsing for scene understanding. In Proceedings of the European conference on computer vision (ECCV), pages 418–434, 2018.

[27] H. Yang, H. Yin, P. Molchanov, H. Li, and J. Kautz. Nvit: Vision transformer compression and parameter redistribution. 2021.

[28] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Scene parsing through ade20k dataset. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 633–641, 2017.

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

[30] M. Zhu, Y. Tang, and K. Han. Vision transformer pruning. arXiv preprint arXiv:2104.08500, 2021.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The main claims accurately reflected the paper's contribitions and scope.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [No]

   Justification: We conducted experiments on a limited set of tasks and did not extend the scope to all computer vision tasks using visual state space models. We did not discuss this limitation in our work because we currently lack evidence to suggest that our method cannot be expanded to other tasks.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We will open source our method once the paper is accepted. Our implementation can be easily reproduced using the main repository of the visual state space model, ensuring that all necessary details to replicate the main experimental results are accessible.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [NA]

Justification: We did not include our code in the submission. However, the experimental settings provided, along with the base visual state space model we used, offer sufficient information for reproducing the experimental results.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The details of experiments settings are attached in the appendix.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [NA]

Justification: The nature of our experiments does not require the use of error bars, as the results are consistent and deterministic. Therefore, reporting error bars or statistical significance measures would not provide additional insight into the findings.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We provided detailed information about the necessary computing resources in the appendix, focusing on the GPU workload, which is the primary concern of this work. CPU specifications and memory details were not included, as they are not relevant to the experiments conducted.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: This work conformed with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: This work won't have direct societal impacts as far as we know.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: This paper work poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We have properly credited the original creators by citing the relevant papers and explicitly mentioning the licenses and terms of use where applicable.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: No new assets were introduced in this paper. This work focuses solely on presenting the algorithm and experimental results.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.