Snakes and Ladders: Two Steps Up for VideoMamba Supplementary Material

Hui Lu
Utrecht University
h.lu1@uu.nl

Albert A. Salah
Utrecht University
a.a.salah@uu.nl

Ronald Poppe
Utrecht University
r.w.poppe@uu.nl

We provide the architectures for VideoMambaPro models in Section 1. A comparison between the architectures of VideoMamba and VideoMambaPro appears in Section 2. Training details are presented in Section 3. Finally, we report ImageNet-1K image classification results in Section 4, to illustrate the potental of the proposed solutions for image classification tasks.

1. VideoMambaPro architectures

We present the architecture details of VideoMambaPro-Tiny (Ti), -Small (S), and -Middle (M) in Tables 1–3. The differences are in the embedding dimension (192, 384, 576) and the number of SSM blocks (24, 24, 32).

Stage	Tiny
Patch Embedding	nn.Conv3d (kernel size = $16 \times 16 \times 1$, **embedding dimension = 192**)
SSM	$\begin{bmatrix} \text{MLP}(768) \\ \text{MLP}(3072) \\ \text{MHA (head = 12)} \end{bmatrix} \times$ **24**
Projection	Layer Normalization Dropout (ratio) Linear layer (1000) Softmax

Table 1. Architecture details of VideoMambaPro-Ti.

Stage	Small
Patch Embedding	nn.Conv3d (kernel size = $16 \times 16 \times 1$, **embedding dimension = 384**)
SSM	$\begin{bmatrix} \text{MLP}(768) \\ \text{MLP}(3072) \\ \text{MHA (head = 12)} \end{bmatrix} \times$ **24**
Projection	Layer Normalization Dropout (ratio) Linear layer (1000) Softmax

Table 2. Architecture details of VideoMambaPro-S.

2. Architecture comparison with VideoMamba

We compare the architectures of VideoMambaPro and VideoMamba [1] in Figure 1. VideoMambaPro does not have the linear layer to generate parameters z. Additionally, our residual SSM and mask scheme do not introduce additional parameters or computational overhead, so our method has slightly fewer parameters and FLOPs.

Stage	Middle
Patch Embedding	nn.Conv3d (kernel size = $16 \times 16 \times 1$, **embedding dimension = 576**)
SSM	$\begin{bmatrix} \text{MLP}(768) \\ \text{MLP}(3072) \\ \text{MHA (head = 12)} \end{bmatrix} \times$ **32**
Projection	Layer Normalization Dropout (ratio) Linear layer (1000) Softmax

Table 3. Architecture details of VideoMambaPro-M.

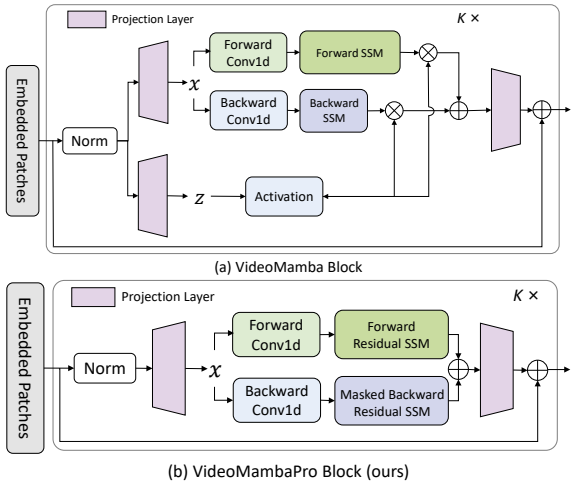

Figure 1. Comparison between the bi-directional VideoMamba (top) and VideoMambaPro (bottom) blocks.

3. Implementation details

We conduct pre-training on ImageNet-1K and fine-tuning on the Something-Something V2 and Kinetics-400 datasets with 16 NVIDIA A100-80G GPUs. Models for UCF101

and HMDB51 are trained with 8 A100-80G GPUs. The experiments on AVA V2.2 are conducted with 32 A100-80G GPUs. The values of the hyperparameters are largely similar to those used in VideoMamba [1]. We linearly scale the base learning rate with respect to the overall batch size, $lr = lr_{base} \times batchsize/256$. The pre-training details are shown in Table 4, and the fine-tuning details on the other datasets are listed in Tables 5–8.

config	image size: 224 × 224
optimizer	AdamW
base learning rate	1.5e-4
weight decay	0.1 (Tiny), 0.05 (Small, Middle)
minimal learning rate	1.0e-6
optimizer momentum	β_1, β_2 = 0.9, 0.95
batch size	512
learning rate schedule	cosine decay
warmup epochs	5 (Tiny), 10 (Small), 40 (Middle)
dropout ratio	0 (Tiny), 0.15 (Small), 0.5 (Middle)
augmentation	MultiScaleCrop
label smoothing	0.1

Table 4. Pre-training setting on ImageNet-1K

config	image size: 224 × 224
optimizer	AdamW
base learning rate	1.5e-4
weight decay	0.1 (Tiny), 0.05 (Small, Middle)
minimal learning rate	1.0e-6
optimizer momentum	β_1, β_2 = 0.9, 0.99
batch size	256
learning rate schedule	cosine decay
warmup epochs	5 (Tiny), 5 (Small) 10 (Middle)
dropout ratio	0.1 (Tiny), 0.35 (Small), 0.6 (Middle)
augmentation	RandAug (7, 0.25) (Tiny), RandAug (9, 0.5) (Small, Middle)
label smoothing	0.1
flip augmentation	yes

Table 5. Fine-tuning setting for Kinetics-400

config	image size: 224 × 224
optimizer	AdamW
base learning rate	4e-4
weight decay	0.1 (Tiny), 0.05 (Small, Middle)
minimal learning rate	1.0e-6
optimizer momentum	β_1, β_2 = 0.9, 0.999
batch size	256
learning rate schedule	cosine decay
warmup epochs	5 (Tiny), 5 (Small) 10 (Middle)
dropout ratio	0.1 (Tiny), 0.35 (Small), 0.6 (Middle)
augmentation	RandAug (7, 0.25) (Tiny), RandAug (9, 0.5) (Small, Middle)
label smoothing	0.1
flip augmentation	no

Table 6. Fine-tuning setting for Something-Something V2

4. Results on ImageNet-1K

We argue that the proposed solutions in handling the feature extraction capabilities of Mamba models are most effective when relevant tokens are more sparsely distributed in the input. While our main focus is on the video domain, we also summary our experiments on image classification. We pre-train VideoMambaPro on ImageNet-1K, which contains 1.28M training images and 50K validation images across 1,000 categories. All models are trained on the training set, and top-1 accuracy on the validation set is reported. For fair comparison, we adopt the same method as VideoMamba, and our training settings primarily follows DeiT [2]. When training on 224^2 input images, we use AdamW with a momentum of 0.9 and a total batch size of 512. Training is performed on 8 A800 GPUs, with more details provided in Table 4. The results are summarized in Table 9. VideoMambaPro achieves accuracy gains of 0.9-2.0% over VideoMamba.

config	image size: 224 × 224
optimizer	AdamW
base learning rate	4e-4
weight decay	0.1 (Tiny), 0.05 (Small, Middle)
minimal learning rate	1.0e-6
optimizer momentum	β_1, β_2 = 0.9, 0.99
batch size	128
learning rate schedule	cosine decay
warmup epochs	5 (Tiny), 5 (Small) 10 (Middle)
dropout ratio	0.1 (Tiny), 0.35 (Small), 0.6 (Middle)
augmentation	RandAug (7, 0.25) (Tiny), RandAug (9, 0.5) (Small, Middle)
label smoothing	0.1
flip augmentation	yes

Table 7. Fine-tuning setting for UCF101/HMDB51

config	image size: 224 × 224
optimizer	AdamW
base learning rate	1.5e-3 (Tiny), 2.5e-4 (Small, Middle)
weight decay	0.051 (Tiny, Small, Middle)
minimal learning rate	1.0e-6
optimizer momentum	β_1, β_2 = 0.9, 0.999
batch size	128
learning rate schedule	cosine decay
warmup epochs	5 (Tiny), 5 (Small) 10 (Middle)
dropout ratio	0.1 (Tiny), 0.35 (Small) 0.6 (Middle)
augmentation	RandAug (7, 0.25) (Tiny), RandAug (9, 0.5) (Small, Middle)
label smoothing	0.1
flip augmentation	yes

Table 8. Fine-tuning setting for AVA 2.2

	Input	Param	FLOPs	Top-1
VideoMamba (Ti)	224^2	7M	1.1G	76.9
VideoMambaPro (Ti)	224^2	7M	1.1G	78.9
VideoMamba (S)	224^2	26M	4.3G	81.2
VideoMambaPro (S)	224^2	25M	4.2G	82.4
VideoMamba (M)	224^2	74M	12.7G	82.8
VideoMambaPro (M)	224^2	72M	12.4G	83.8
VideoMamba (M)	448^2	75M	50.4G	83.8
VideoMambaPro (M)	448^2	73M	49.6G	84.7

Table 9. ImageNet-1K pre-training results for VideoMamba and VideoMambaPro.

References

[1] Kunchang Li, Xinhao Li, Yi Wang, Yinan He, Yali Wang, Limin Wang, and Yu Qiao. VideoMamba: State space model for efficient video understanding. *arXiv preprint arXiv:2403.06977*, 2024. 1, 2

[2] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, pages 10347–10357. PMLR, 2021. 2