VideoMamba: State Space Model for Efficient Video Understanding

Kunchang Li[1,2,3♠], Xinhao Li[4,3♠], Yi Wang[3♡], Yinan He[3]
Yali Wang[1,3♡], Limin Wang[4,3♡], and Yu Qiao[3,1♡]

[1] Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences
[3] OpenGVLab, Shanghai AI Laboratory
[4] State Key Laboratory for Novel Software Technology, Nanjing University
`Code & Models:` `https://github.com/OpenGVLab/VideoMamba`

Abstract. Addressing the dual challenges of local redundancy and global dependencies in video understanding, this work innovatively adapts the Mamba to the video domain. The proposed VideoMamba overcomes the limitations of existing 3D convolution neural networks (CNNs) and video transformers. Its linear-complexity operator enables efficient long-term modeling, which is crucial for high-resolution long video understanding. Extensive evaluations reveal VideoMamba's four core abilities: (1) *Scalability* in the visual domain without extensive dataset pretraining, thanks to a novel self-distillation technique; (2) *Sensitivity* for recognizing short-term actions even with fine-grained motion differences; (3) *Superiority* in long-term video understanding, showcasing significant advancements over traditional feature-based models; and (4) *Compatibility* with other modalities, demonstrating robustness in multi-modal contexts. Through these advantages, VideoMamba sets a new benchmark, offering a scalable and efficient solution for comprehensive video understanding.

Keywords: Mamba · Video Understanding · Multimodal Learning

1 Introduction

The core objective for video understanding lies in mastering spatiotemporal representations, which presents two formidable challenges: large spatiotemporal redundancy in short video clips and complex spatiotemporal dependencies in long contexts. Although the once-dominant 3D CNNs [9,20,77] and video transformers [2,4] effectively tackle one of these challenges by leveraging either local convolution or long-range attention, they fall short in addressing both simultaneously. UniFormer [44] attempts to integrate the advantages of both methods, but it struggles with modeling long videos, which has been the major trend in recent research on video understanding [48,73] and generation [5,92].

♠ Interns at Shanghai AI Laboratory. ♡ Corresponding authors.

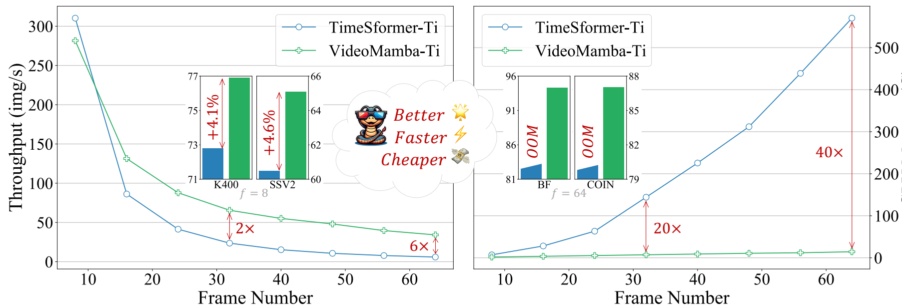


Fig. 1: Comparisons of throughput and memory. The TimeSformer-Ti [4] is built based on DeiT-Ti [76] with joint spatiotemporal attention. Our VideoMamba is ***better, faster and cheaper*** for both short-term and long-term video understanding.

The emergence of low-cost operators such as S4 [26], RWKV [74], and RetNet [71] in the NLP domain, has carved a novel pathway for the vision model. Mamba [25] stands out with its selective state space model (SSM), striking a balance between maintaining linear complexity and facilitating long-term dynamic modeling. This innovation has spurred its adoption in vision tasks, as evidenced by Vision Mamba [91] and VMamba [50], which leverage multi-directional SSMs for enhanced 2D image processing. These models rival attention-based architectures in performance while offering a significant reduction in memory usage. Given the inherently longer sequences produced by video, a natural question arises: ***Can Mamba work well for video understanding?***

Inspired by this, we introduce VideoMamba, a purely SSM-based model tailored for video understanding. VideoMamba harmoniously merges the strengths of convolution and attention in vanilla ViT [16] style. It offers a linear-complexity method for dynamic spatiotemporal context modeling, ideal for high-resolution long videos. The related evaluation focuses on VideoMamba's four key abilities:

(1) *Scalability in the Visual Domain*: We examine VideoMamba's scalability and find that, while the pure Mamba model tends to overfit as it scales, our introduction of a simple yet effective self-distillation strategy allows VideoMamba to achieve remarkable performance enhancements as the model and input sizes increase, without the need for large-scale dataset pretraining.

(2) *Sensitivity for Short-term Action Recognition*: Our analysis extends to assessing VideoMamba's capability to accurately distinguish short-term actions, especially those with fine-grained motion differences, *e.g.*, opening and closing. The findings reveal VideoMamba's superior performance over existing attention-based models [2,4,52]. More importantly, it is also suitable for masked modeling, which further enhances its temporal sensitivity.

(3) *Superiority in Long-term Video Understanding*: We then assess VideoMamba's prowess in interpreting long videos. It showcases remarkable superiority over conventional feature-based methods [36, 47] through end-to-end training. Notably, VideoMamba operates 6× faster than TimeSformer [4] and demands 40× less GPU memory for 64-frame videos (see Fig. 1).

(4) ***Compatibility with Other Modalities*****:** Lastly, we assess VideoMamba's adaptability with other modalities. Results in video-text retrievals show its improved performance than ViT, particularly in long videos with complex scenarios. This underscores its robustness and multi-modal integration capacity.

In conclusion, our experiments reveal VideoMamba's potential in understanding both short-term (K400 [37] and SthSthV2 [24]) and long-term (Breakfast [38], COIN [72], and LVU [86]) video contents. Given its efficiency and effectiveness, VideoMamba is poised to become a cornerstone in long-video comprehension.

2 Related Works

2.1 State Space Models

Recently, State Space Models (SSMs) have shown significant effectiveness in capturing the dynamics and dependencies of language sequences. [26] introduces a structured state-space sequence model (S4), designed to model long-range dependencies with linear complexity. Based on it, various models have been developed (*e.g.*, S5 [67], H3 [21], and GSS [57]). Mamba [25] distinguishes itself by introducing a data-dependent SSM layer and a selection mechanism using parallel scan (S6). Compared to transformers [6,54] with quadratic-complexity attention, Mamba excels at processing long sequences with linear complexity.

In the vision domain, [26] first applies SSM in pixel-level image classification, and [36] uses S4 to handle long-range temporal dependencies for movie clip classification. Mamba's potential has motivated a series of works [11,28,30,32,46,50,56,79,80,88,91], demonstrating better performance and higher GPU efficiency than Transformers on visual tasks like object detection and semantic segmentation. Unlike previous works, our VideoMamba is the first purely SSM-based video model, demonstrating exceptional efficiency and effectiveness in both short-term and long-term video understanding.

2.2 Video Understanding

Video understanding is a cornerstone of computer vision, amplified by the growth of short video platforms. To advance this field, numerous datasets with extensive data and meticulous human annotations have been developed to enhance action recognition. Notable examples include UCF101 [68] and Kinetics [7,8,37], which have been pivotal in benchmarking progress. Other datasets [22,27,31,35,49,63] provide annotated activity videos for action localization, fostering deeper research into human activities. Beyond action recognition, large-scale video-text datasets [10,13,58,84,87,89] extend video understanding into multi-modality tasks like video captioning, retrieval, and question answering.

The architecture has evolved from CNNs to more advanced techniques. Initially, 3D CNNs [9,18,77,78] expanded traditional 2D CNNs to capture spatiotemporal information. Two-Stream [66], TSN [82], and SlowFast [20] further enhanced action recognition by combining spatial and temporal streams, proposing sparse sampling, and using parallel networks, respectively. Attention-based

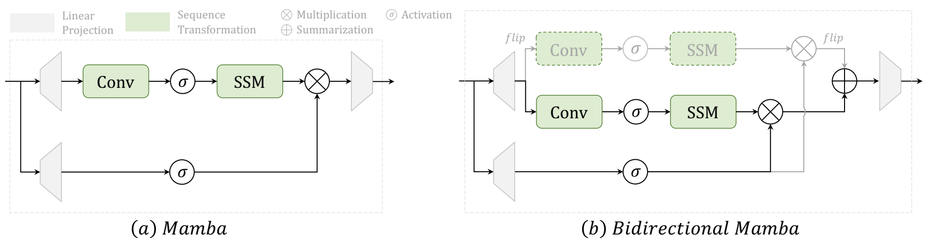


Fig. 2: Mamba blocks for 1D [25] and 2D [91] sequence. We omit the initial normalization and the final residual for simplification.

models [2,4,60,64,90] like TimeSformer [4] and ViViT [2] significantly advanced the field by capturing long-range dependencies, improving temporal relationship understanding. Recent models [42,44,52,85] have focused on efficient video transformers, with innovations like VideoSwin's window attention [52] and UniFormer's integration of convolution and self-attention [44], balancing computational efficiency with performance. Despite these advancements, high computational costs remain for long sequences. In contrast, our VideoMamba introduces a linear-complexity operator for efficient long-term modeling, outperforming existing methods with faster speed and lower GPU consumption.

3 Method

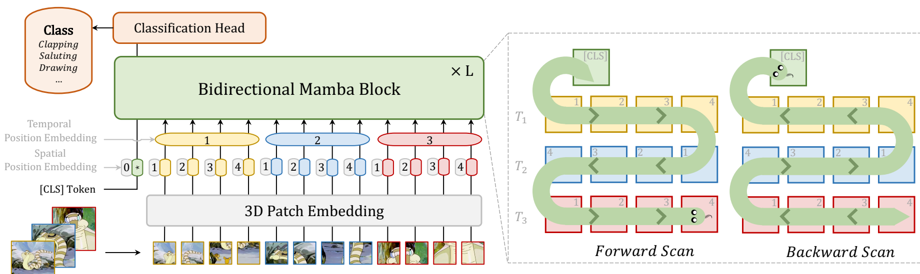
3.1 Preliminaries

SSM for 1D sequence. State Space Models (SSMs) are conceptualized based on continuous systems that map a 1D function or sequence, $x(t) \in \mathbb{R}^L \rightarrow y(t) \in \mathbb{R}^L$ through a hidden state $h(t) \in \mathbb{R}^N$. Formally, SSMs employ the following ordinary differential equation (ODE) to model the input data:

$$h'(t) = \mathbf{A}h(t) + \mathbf{B}x(t), \tag{1}$$
$$y(t) = \mathbf{C}h(t), \tag{2}$$

where $\mathbf{A} \in \mathbb{R}^{N \times N}$ represents the system's evolution matrix, and $\mathbf{B} \in \mathbb{R}^{N \times 1}, \mathbf{C} \in \mathbb{R}^{N \times 1}$ are the projection matrices. This continuous ODE is approximated through discretization in modern SSMs. Mamba [25] is one of the discrete versions of the continuous system, which includes a timescale parameter $\mathbf{\Delta}$ to transform the continuous parameters $\mathbf{A}, \mathbf{B}$ to their discrete counterparts $\overline{\mathbf{A}}, \overline{\mathbf{B}}$. The transformation typically employs the zero-order hold (ZOH) method, defined by:

$$\overline{\mathbf{A}} = \exp(\mathbf{\Delta A}), \tag{3}$$
$$\overline{\mathbf{B}} = (\mathbf{\Delta A})^{-1}(\exp(\mathbf{\Delta A}) - \mathbf{I}) \cdot \mathbf{\Delta B} \tag{4}$$
$$h_t = \overline{\mathbf{A}}h_{t-1} + \overline{\mathbf{B}}x_t, \tag{5}$$
$$y_t = \mathbf{C}h_t. \tag{6}$$



(a) *VideoMamba* (b) *Spatiotemporal Scan*

Fig. 3: Framework of VideoMamba. We strictly follow the architecture of vanilla ViT [16], and adapt the bidirectional mamba block [91] for 3D video sequences.

Contrary to traditional models that primarily rely on linear time-invariant SSMs, Mamba distinguishes itself by implementing a Selective Scan Mechanism (S6) as its core SSM operator. Within S6, the parameters $\mathbf{B} \in \mathbb{R}^{B\times L\times N}$, $\mathbf{C} \in \mathbb{R}^{B\times L\times N}$, and $\mathbf{\Delta} \in \mathbb{R}^{B\times L\times D}$ are directly derived from the input data $x \in \mathbb{R}^{B\times L\times D}$, indicating an intrinsic capacity for contextual sensitivity and adaptive weight modulation. Fig. 2a shows the details of the Mamba block.

Bidirectional SSM for Vision. The original Mamba block, designed for 1D sequences, falls short for visual tasks requiring spatial awareness. Building on this, Vision Mamba introduces a bidirectional Mamba (B-Mamba) block in Fig. 2b, which adapts bidirectional sequence modeling for vision-specific applications. This block processes flattened visual sequences through simultaneous forward and backward SSMs, enhancing its capacity for spatially-aware processing. In this work, we extend the B-Mamba block for 3D video understanding.

3.2 VideoMamba

Overview. Fig. 3 illustrates the overall framework of VideoMamba. Specifically, we first use 3D convolution (*i.e.*, 1×16×16) to project the input videos $\mathbf{X}^v \in \mathbb{R}^{3\times T\times H\times W}$ into L non-overlapping spatiotemporal patches $\mathbf{X}^p \in \mathbb{R}^{L\times C}$, where $L{=}t{\times}h{\times}w$ ($t{=}T$, $h{=}\frac{H}{16}$, and $w{=}\frac{W}{16}$). The sequence of tokens input to the following VideoMamba encoder is

$$\mathbf{X} = [\mathbf{X}_{cls}, \mathbf{X}] + \mathbf{p}_s + \mathbf{p}_t, \tag{7}$$

where $\mathbf{X}_{cls}$ is a learnable classification token that is prepended to the start of the sequence. Following previous works [2, 4, 16], we added a learnable spatial position embedding $\mathbf{p}_s \in \mathbb{R}^{(hw+1)\times C}$ and the extra temporal one $\mathbf{p}_t \in \mathbb{R}^{t\times C}$ to retain the spatiotemporal position information, since the SSM modeling is sensitive to token position. The tokens $\mathbf{X}$ are then passed through by L stacked B-Mamba blocks, and the representation of [CLS] token at the final layer is processed by normalization and linear layer for classification.

Spatiotemporal Scan. To apply the B-Mamba layer to spatiotemporal input, we extend the original 2D scan into different bidirectional 3D scans in Fig. 4: (a)

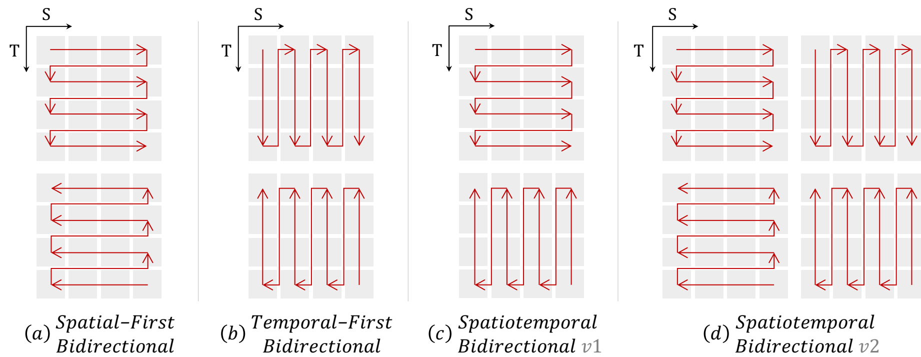


Fig. 4: Different scan methods. We omit the [CLS] token for simplification.

Spatial-First, organizing spatial tokens by location, then stacking them frame by frame; (b) *Temporal-First*, arranging temporal tokens by frame, then stacking them along the spatial dimension; (c) *Spatiotemporal*, a hybrid of *Spatial-First* and *Temporal-First*, with v1 conducting half and v2 conducting full (2× computation). Our experiments in Fig. 7a demonstrate that the Spatial-First bidirectional scan is the most effective and simple. Thanks to Mamba's linear complexity, our VideoMamba can efficiently handle long, high-resolution videos.

Comparison to Vim [91] and VMamba [50]. Our VideoMamba builds upon Vim, streamlining its architecture by omitting the middle [CLS] token and Rotary Position Embedding (RoPE [69]), resulting in superior performance on ImageNet-1K with gains of +**0.8%** and +**0.7%** for Vim-Ti and Vim-S, respectively. Unlike VMamba, which incorporates additional depthwise convolution, VideoMamba strictly follows the ViT design without downsampling layers. To counter overfitting issues observed in VMamba, we introduce an effective self-distillation technique outlined in Section 3.3, demonstrating VideoMamba's great scalability for image and video tasks.

Comparison to TimeSformer [4] and ViViT [2]. Traditional attention-based models like TimeSformer and ViViT address the self-attention mechanism's quadratic complexity by adopting divided spatiotemporal attention. Despite being more efficient, it introduces additional parameters and underperforms compared to joint attention, particularly in masked pretraining scenarios [43,75]. In contrast, VideoMamba processes spatiotemporal tokens with linear complexity, outperforming TimeSformer on Kinetics-400 by +**2.6%** and making significant strides on SthSthV2 with a +**5.9%** improvement (see Table 3 and 4). Furthermore, VideoMamba achieves a **6**× increase in processing speed and requires **40**× less GPU memory for long videos (see Fig. 1, demonstrating its efficiency and effectiveness in handling long-video tasks.

3.3 Architecture

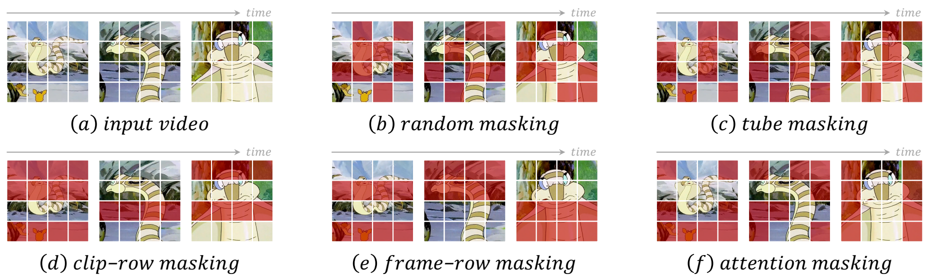


Fig. 5: Different masking strategies. Row masking, tailored for VideoMamba in light of the 1D convolution preceding SSM, enhances performance with continuous tokens. The difference between clip-row and frame-row masking is that the former masks the entire video clip, while the latter masks each frame individually.

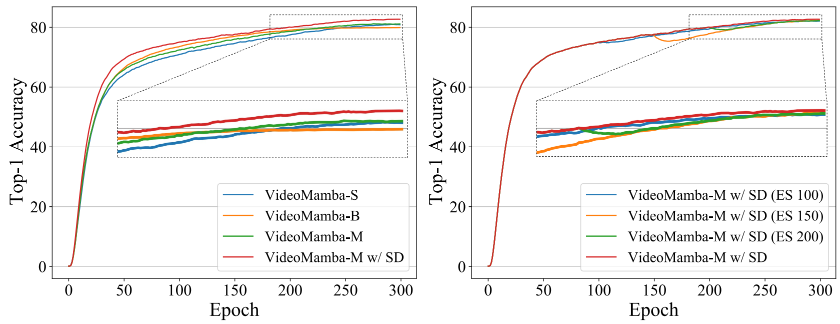
Table 1: Different model sizes. Base model is finally excluded due to its suboptimization.

Model	#Depth	#Dim	#Param.
Tiny	24	192	7M
Small	24	384	26M
Middle	32	576	74M
Base	24	768	98M

For SSM in the B-Mamba layer, we adopt the default hyperparameters as in Mamba [25]. setting the state dimension and expansion ratio to 16 and 2, respectively. Following ViT [16], we adjust the depth and embedding dimensions to create models of comparable sizes in Table 1, including VideoMamba-Ti, VideoMamba-S and VideoMamba-M. However, we observe that larger VideoMamba tends to overfit during our experiments, leading to suboptimal performance as illustrated in Fig. 6a. This overfitting issue is not unique to our models but is also found in VMamba [50], where the optimal performance of VMamba-B was achieved at three-quarters of the total training epochs. To counteract the overfitting in larger Mamba models, we introduce an effective Self-Distillation strategy, which uses a smaller and well-trained model as the "teacher" to guide the training of the larger "student" model. The results, depicted in Fig. 6a, show that this strategy leads to expected better convergence.

3.4 Masked Modeling

Recently, VideoMAE and ST-MAE [19,75] have showcased the significant benefits of masked modeling in enhancing a model's capability for FINE-GRAINED temporal understanding. UMT [43] takes this further by introducing an efficient masked alignment technique that yields robust results across single and multi-modal video tasks. To augment VideoMamba's temporal sensitivity and verify its adaptability with text modalities, we adopt a masked alignment approach inspired by UMT. Firstly, VideoMamba is trained from scratch on video data alone, aligning unmasked tokens with those from CLIP-ViT. Subsequently, it is integrated with a text encoder and a cross-modal decoder (*i.e.*, BERT [15]), for pretraining on both image-text and video-text datasets.



(a) *Self-Distillation avoids overfitting.* (b) *Early stopping does not help.*

Fig. 6: Ablation studies of Self-Distillation and Early Stopping.

Note that different from UMT, which employs multi-layer alignment between student and teacher models, we align only the final outputs due to VideoMamba's unique architecture (SSM *vs.* Transformer), Regarding our masking strategy, we propose different row masking techniques, depicted in Fig. 5, tailored to the B-Mamba block's preference for continuous tokens. Additionally, we explore attention masking to preserve meaningful adjacency among tokens, leveraging the 1D convolution within the B-Mamba block for improved performance.

4 Experiments

4.1 Scaling Up

Dataset and Settings. We first conduct experiments on ImageNet-1K [14], which includes 1.28M training images and 50K validation images across 1,000 categories. For fair comparisons, we follow most of the training strategies of DeiT [76], but adopt weaker data augmentation for the tiny variant. We adjust the stochastic depth ratio to 0/0.15/0.5 for VideoMamba-Ti/S/M. Our models are trained using the AdamW optimizer with a cosine learning rate schedule over 300 epochs, with the initial 5 epochs for linear warm-up. Default settings for the learning rate, weight decay, and batch size are 1e-3, 0.05, and 1024, respectively. We use BFloat16 precision during training to enhance stability without EMA. For the VideoMamba-M model, we employ a pretrained VideoMamba-S model as a "teacher" to guide the training process by aligning the final feature maps through L2 loss. For large resolution (>224) fine-tuning, we use a reduced learning rate (5e-6) and minimal weight decay (1e-8) for 30 epochs.
Effect of Self-Distillation. Fig. 6a reveals that when trained from scratch, VideoMamba-B tends to overfit more easily and underperforms compared to VideoMamba-S, whereas VideoMamba-M achieves similar performances. Fortunately, our self-distillation has shown to be effective in achieving the desired optimization with marginal additional computational cost. To mitigate teacher's potential overdirection, we experimented with early stopping [12] in Fig. 6b,

Table 2: Comparison with the state-of-the-art on ImageNet. "*iso.*" means isotropic architecture without downsampling layers.

Arch.	Model	*iso.*	Input Size	#Param (M)	FLOPs (G)	IN-1K Top-1
CNN	ConvNeXt-T [53]	✗	224^2	29	4.5	82.1
	ConvNeXt-S [53]	✗	224^2	50	8.7	83.1
	ConvNeXt-B [53]	✗	224^2	89	15.4	**83.8**
Trans.	SwinT-T [51]	✗	224^2	28	4.5	81.3
	Swin-S [51]	✗	224^2	50	8.7	83.0
	Swin-B [51]	✗	224^2	88	15.4	83.5
CNN+ SSM	VMamba-T [50]	✗	224^2	22	5.6	82.2
	VMamba-S [50]	✗	224^2	44	11.2	83.5
	VMamba-B [50]	✗	224^2	75	18.0	<u>83.7</u>
CNN	ConvNeXt-S [53]	✓	224^2	22	4.3	79.7
	ConvNeXt-B [53]	✓	224^2	87	16.9	82.0
Trans.	DeiT-Ti [76]	✓	224^2	6	1.3	72.2
	DeiT-S [76]	✓	224^2	22	4.6	79.8
	DeiT-B [76]	✓	224^2	87	17.6	81.8
	DeiT-B [76]	✓	384^2	87	55.5	<u>83.1</u>
SSM	S4ND-ViT-B [59]	✓	224^2	89	-	80.4
	Vim-Ti [91]	✓	224^2	7	1.1	76.1
	Vim-S [91]	✓	224^2	26	4.3	80.5
	VideoMamba-Ti	✓	224^2	7	1.1	76.9
	VideoMamba-Ti	✓	448^2	7	4.3	79.3
	VideoMamba-S	✓	224^2	26	4.3	81.2
	VideoMamba-S	✓	448^2	26	16.9	83.2
	VideoMamba-M	✓	224^2	74	12.7	82.8
	VideoMamba-M	✓	448^2	75	50.4	83.8
	VideoMamba-M	✓	576^2	75	83.1	**84.0**

although it did not yield beneficial outcomes. These findings indicate that self-distillation offers a viable strategy for enhancing the scalability of the Mamba architecture without significant computational overhead.

Results. Table 2 showcases the results on the ImageNet-1K dataset. Notably, VideoMamba-M outperforms other isotropic architectures by significant margins, achieving a $+\mathbf{0.8\%}$ improvement over ConvNeXt-B [53] and a $+\mathbf{2.0\%}$ increase compared to DeiT-B [76], while utilizing fewer parameters. Additionally, VideoMamba-M holds its ground against non-isotropic backbones that leverage hierarchical features for enhanced performance. Given Mamba's efficiency in processing long sequences, we further enhance performance by increasing the resolution, achieving a top-1 accuracy of **84.0%** with only 74M parameters. This remarkable improvement extends to video tasks, as detailed in Section 4.2, underscoring VideoMamba's effectiveness and scalability.

4.2 Short-term Video Understanding

Datasets and Settings. We evaluate VideoMamba on the scene-related Kinetics-400 [37] and temporal-related Something-Something V2 [24], with average video lengths of 10s and 4s, respectively. For supervised pretraining, we fine-tune models pretrained on ImageNet-1K using the same strategy as VideoMAE [75].

Table 3: Comparison with the state-of-the-art on scene-related Kinetics-400. "*iso.*" means isotropic architecture without downsampling layers. Masked modeling [43] also works for Mamba, but the inconsistent architecture leads to inferior alignment.

Arch.	Model	*iso.*	Extra Data	Input Size	#Param (M)	FLOPs (G)	K400 Top-1	K400 Top-5
Supervised: *Those models with extra data are under supervised training.*								
CNN	$\text{SlowFast}_{R101+NL}$ [20]	✗		80×224^2	60	234×3×10	79.8	93.9
	X3D-XL [18]	✗		16×312^2	20	194×3×10	80.4	94.6
Trans.	Swin-T [52]	✗	IN-1K	32×224^2	28	88×3×4	78.8	93.6
	Swin-B [52]	✗	IN-1K	32×224^2	88	88×3×4	80.6	94.5
	Swin-B [52]	✗	IN-21K	32×224^2	88	282×3×4	<u>82.7</u>	**95.5**
CNN+ Trans.	MViTv1-B [17]	✗		32×224^2	37	70×1×5	80.2	94.4
	MViTv2-S [45]	✗		16×224^2	35	64×1×5	81.0	94.6
	UniFormer-S [44]	✗	IN-1K	16×224^2	21	42×1×4	80.8	94.7
	UniFormer-B [44]	✗	IN-1K	32×224^2	50	259×3×4	**83.0**	<u>95.4</u>
Trans.	STAM [64]	✓	IN-21K	64×224^2	121	1040×1×1	79.2	-
	TimeSformer-L [4]	✓	IN-21K	96×224^2	121	2380×3×1	80.7	<u>94.7</u>
	ViViT-L [2]	✓	IN-21K	16×224^2	311	3992×3×4	<u>81.3</u>	<u>94.7</u>
SSM	VideoMamba-Ti	✓	IN-1K	16×224^2	7	17×3×4	78.1	93.5
	VideoMamba-Ti	✓	IN-1K	32×224^2	7	34×3×4	78.8	93.9
	VideoMamba-S	✓	IN-1K	16×224^2	26	68×3×4	80.8	94.8
	VideoMamba-S	✓	IN-1K	32×224^2	26	135×3×4	81.5	95.2
	VideoMamba-M	✓	IN-1K	16×224^2	74	202×3×4	81.9	95.4
	VideoMamba-M	✓	IN-1K	32×224^2	74	403×3×4	82.4	95.7
	VideoMamba-M	✓	IN-1K	64×384^2	74	2368×3×4	**83.3**	**96.1**
Self-supervised: *For UMT, the CLIP-400M is used in pretrained teacher.*								
Trans.	BEVT-B_{800e} [83]	✗	IN-1K	32×224^2	88	282×3×4	81.1	-
	$\text{VideoMAE-S}_{2400e}$ [75]	✓		16×224^2	22	57×3×5	79.0	93.8
	$\text{VideoMAE-B}_{1600e}$ [75]	✓		16×224^2	87	180×3×5	81.5	95.1
	UMT-B_{800e} [43]	✓	CLIP-400M	8×224^2	87	180×3×5	**85.7**	**97.0**
SSM	$\text{VideoMamba-M}_{800e}$	✓	CLIP-400M	8×224^2	74	101×3×4	82.0	95.4
	$\text{VideoMamba-M}_{800e}$	✓	CLIP-400M	16×224^2	74	202×3×4	83.4	95.9
	$\text{VideoMamba-M}_{800e}$	✓	CLIP-400M	32×224^2	74	403×3×4	83.9	96.2
	$\text{VideoMamba-M}_{800e}$	✓	CLIP-400M	64×384^2	74	2368×3×4	<u>85.0</u>	<u>96.9</u>

Specifically, for VideoMamba-M, the warmup epoch, total epoch, stochastic depth rate, and weight decay are set to 5, 50, 0.8, and 0.05 for K400, and 5, 30, 0.8, and 0.05 for SthSth. For smaller models, all hyperparameters are the same except for a decreased stochastic depth rate and increased training epochs. We linearly scale the base learning rates according to batch size: $2e^{-4} \cdot \frac{batchsize}{256}$ for K400 and $4e^{-4} \cdot \frac{batchsize}{256}$ for SthSth. For self-supervised pretraining, we adopt the UMT [43] recipe, using CLIP-ViT-B [61] to distill VideoMamba-M over 800 epochs. During fine-tuning, we use similar hyperparameters but opt for a smaller stochastic depth rate and learning rate for both datasets.

Results. Table 3 and 4 list the results on short-term video datasets. **(a)** ***Supervised*****:** Compared with the purely attention-based methods [2,4], our SSM-based VideoMamba-M secures a notable advantage, outperforming ViViT-L [2] by **+2.0%** and **+3.0%** on the scene-related K400 and the temporally-related SthSthV2 datasets, respectively. This improvement comes with significantly reduced computational demands and less pretraining data. Furthermore, VideoMamba-M delivers results that are on par with the SOTA UniFormer [44], which skillfully

Table 4: Comparison with the state-of-the-art on temporal-related SthSth V2. "*iso.*" means isotropic architecture without downsampling layers. Masked modeling [43] also works for Mamba, and it performs better than VideoMAE.

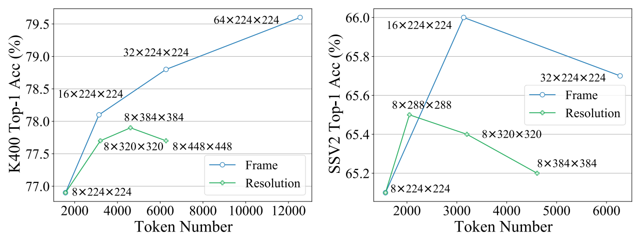
Arch.	Model	*iso.*	Extra Data	Input Size	#Param (M)	FLOPs (G)	SSV2 Top-1	SSV2 Top-5
Supervised: *Those models with extra data are under supervised training.*								
CNN	SlowFast$_{R101}$ [20]	✗	K400	32×224^2	53	106×3×1	63.1	87.6
	CT-Net$_{R50}$ [41]	✗	IN-1K	16×224^2	21	75×1×1	64.5	89.3
	TDN$_{R50}$ [81]	✗	IN-1K	16×224^2	26	75×1×1	65.3	91.6
Trans.	Swin-B [52]	✗	K400	32×224^2	89	88×3×1	69.6	92.7
CNN+ Trans.	MViTv1-B [17]	✗	K400	32×224^2	37	170×3×1	67.1	90.8
	MViTv2-B [45]	✗	K400	32×224^2	51	225×3×1	**70.5**	<u>92.7</u>
	UniFormer-S [44]	✗	IN-1K+K400	16×224^2	21	42×3×1	67.7	91.4
	UniFormer-B [44]	✗	IN-1K+K400	16×224^2	50	97×3×1	<u>70.4</u>	**92.8**
Trans.	TimeSformer-HR [4]	✓	IN-21K	16×224^2	121	1703×3×1	62.5	-
	ViViT-L [2]	✓	IN-21K+K400	16×224^2	311	3992×3×4	<u>65.4</u>	<u>89.8</u>
SSM	VideoMamba-Ti	✓	IN-1K	8×224^2	7	9×3×2	65.1	89.1
	VideoMamba-Ti	✓	IN-1K	16×224^2	7	17×3×2	66.0	89.6
	VideoMamba-Ti	✓	IN-1K	16×288^2	7	28×3×2	66.2	90.0
	VideoMamba-S	✓	IN-1K	8×224^2	26	34×3×2	66.6	90.4
	VideoMamba-S	✓	IN-1K	16×224^2	26	68×3×2	67.6	90.9
	VideoMamba-S	✓	IN-1K	16×288^2	26	112×3×2	68.1	91.2
	VideoMamba-M	✓	IN-1K	8×224^2	74	101×3×4	67.3	91.0
	VideoMamba-M	✓	IN-1K	16×224^2	74	202×3×4	68.3	91.4
	VideoMamba-M	✓	IN-1K	16×288^2	74	333×3×4	**68.4**	**91.6**
Self-supervised: *For UMT, the CLIP-400M is used in pretrained teacher.*								
Trans.	BEVT-B$_{800e}$ [83]	✗	IN-1K+K400	32×224^2	88	321×3×1	70.6	-
	VideoMAE-S$_{2400e}$ [75]	✓		16×224^2	22	57×3×2	66.8	90.3
	VideoMAE-B$_{2400e}$ [75]	✓		16×224^2	87	180×3×2	<u>70.8</u>	92.4
	UMT-B$_{800e}$ [43]	✓	CLIP-400M	8×224^2	87	180×3×2	<u>70.8</u>	<u>92.6</u>
SSM	VideoMamba-M$_{800e}$	✓	CLIP-400M	8×224^2	74	101×3×2	70.2	92.6
	VideoMamba-M$_{800e}$	✓	CLIP-400M	16×224^2	74	202×3×2	71.0	92.7
	VideoMamba-M$_{800e}$	✓	CLIP-400M	16×288^2	74	333×3×2	**71.4**	**92.9**

integrates convolution with attention in a non-isotropic structure. **(b)** ***Self-supervised*****:** The performance of VideoMamba under masked pretraining surpasses that of the VideoMAE [75], known for its proficiency in fine-grained action. This achievement underscores the potential of our purely SSM-based model in efficiently and effectively understanding short-term videos, highlighting its suitability for both supervised and self-supervised learning paradigms.

Ablation Studies. Through comprehensive ablation studies detailed in Fig. 7 and Table 5, we explore various aspects of our model. **(a)** ***Scan Type*****:** Among all methods, the spatial-first approach is the most effective, while the temporal-first strategy is the worst. The superiority of the spatial-first method is due to its ability to leverage 2D pretrained knowledge by scanning frame by frame. **(b)** ***Frame and Resolution*****:** Contrary to findings from ImageNet (see Table 2), higher resolution does not uniformly lead to better performance. Increasing the number of frames consistently enhances results on the K400 dataset. However, this is not the case with SthSthV2, possibly due to the brief duration of its videos, which may not accommodate longer inputs effectively. **(c)** ***Masked Pretraining*****:** Our findings reveal that row masking, being particularly compatible

Type	SSV2
SF-Bidirectional	**65.1**
TF-Bidirectional	62.4
ST-Bidirectional v1	63.9
ST-Bidirectional v2	64.2
Half-SF + Half-TF	64.0
Half-TF + Half-SF	64.1
Alternative SF&TF	**65.1**

(a) Scan Type. Spatial-First scan is simple yet effective.



(b) Frame & Resolution for K400 and SSV2.

Fig. 7: Ablation studies of scan type, frame and resolution. All the models are fine-tuned from VideoMamba-Ti pretrained on ImageNet.

Table 5: Ablation studies of masked pretraining. We adopt CLIP-ViT-B [61] as a teacher to distill VideoMamba-M for 200 epochs.

Type	SSV2
Random	67.4
Tube	66.3
Clip-Row	68.2
Frame-Row	67.8
Attention	**68.5**

(a) Mask Type.

Layer	SSV2
Last 1	**68.5**
Last 2	68.4
Last 6	68.2
Last 6×2	67.7

(b) Alignment Layer.

Ratio	SSV2
50%	68.1
65%	68.4
80%	**68.5**
90%	68.2

(c) Mask Ratio.

DP	SSV2
0.1	68.0
0.2	68.2
0.3	68.4
0.4	**68.5**

(d) Droppath.

with 1D convolution, outperforms random and tube masking. Clip-row masking excels due to its higher degree of randomness. Attention masking stands out as the most efficient by preserving adjacent meaningful content. Aligning only the model's final output is most effective, likely due to architectural differences. Lastly, an optimal masking ratio (80%) combined with stronger regularization significantly benefits VideoMamba during masked pretraining.

4.3 Long-term Video Understanding

Datasets and Settings. We rigorously assess VideoMamba's proficiency in processing long-term videos using three comprehensive datasets: Breakfast [38], COIN [72], and Long-form Video Understanding (LVU [86]). Breakfast comprises 1,712 videos of 10 intricate cooking activities over 77 hours. COIN features 11,827 videos across 180 procedural tasks, averaging 2.36 minutes. The LVU benchmark includes approximately 30K movie clips lasting 1 to 3 minutes, encompassing nine tasks across three categories: content understanding, metadata prediction, and user engagement. For regression tasks, we evaluate using mean-squared error; for classification tasks, accuracy is the metric of choice. Unlike prior studies [36, 47] that rely on features from pretrained video models like Swin-B [51] trained on Kinetics-600, our method uses end-to-end training as detailed in Section 4.2. For fair comparisons, we fine-tune our models pretrained on K400.

Table 6: Comparison with the state-of-the-art on Breakfast and COIN. "*e2e*" means end-to-end methods without exhausting feature extraction. "†" marks the backbone with masked pretraining.

Method	*e2e*	Backbone	Neck Type	Pretraining Dataset	BF Top-1	COIN Top-1
Timeception [33]	✗	3D-ResNet	Conv.	IN-1K+K400	71.3	-
VideoGraph [34]	✗	I3D	Conv.+Atten.	IN-1K+K400	69.5	-
Distant Supervision [47]	✗	TimeSformer	Atten. w/ KB	IN-21K+HTM	**89.9**	**90.0**
ViS4mer [36]	✗	Swin-B	SSM	IN-21K+K600	<u>88.2</u>	<u>88.4</u>
Turbo$_{f32}$ [29]	✓	VideoMAE-B		K400	86.8	82.3
Turbo$_{f32}$ [29]	✓	VideoMAE-B		K400+HTM-AA	<u>91.3</u>	<u>87.5</u>
VideoMamba$_{f32}$	✓	VideoMamba-Ti		K400	94.3	86.2
VideoMamba$_{f64}$	✓	VideoMamba-Ti		K400	94.3	87.0
VideoMamba$_{f32}$	✓	VideoMamba-S		K400	95.3	88.4
VideoMamba$_{f64}$	✓	VideoMamba-S		K400	97.4	88.7
VideoMamba$_{f32}$	✓	VideoMamba-M		K400	94.8	88.3
VideoMamba$_{f64}$	✓	VideoMamba-M		K400	95.8	89.5
VideoMamba$_{f32}$	✓	VideoMamba-M†		K400	**97.9**	89.6
VideoMamba$_{f64}$	✓	VideoMamba-M†		K400	96.9	**90.4**

Table 7: Comparison with the state-of-the-art on LVU. "*e2e*" means end-to-end methods without exhausting feature extraction. "Rel.", "Dir." and "Wtr." refers to "Relation", "Director" and "Writer", respectively.

Method	*e2e*	***Backbone***	Content(↑)			Metadata(↑)				User(↓)	
			Rel.	Speak	Scene	Dir.	Genre	Wtr.	Year	Like	View
VideoBERT [70]	✗	S3D	52.80	37.90	54.90	47.30	51.90	38.50	36.10	0.32	4.46
Object Trans. [86]	✗	ResNet	53.10	39.40	56.90	51.20	54.60	34.50	39.10	**0.23**	<u>3.55</u>
Orthoformer [36]	✗	ViT-L	50.00	39.30	66.27	55.14	<u>55.79</u>	47.02	43.35	0.29	3.86
ViS4mer [36]	✗	ViT-L	<u>57.14</u>	**40.79**	<u>67.44</u>	<u>62.61</u>	54.71	<u>48.80</u>	<u>44.75</u>	<u>0.26</u>	3.63
VideoMamba$_{f32}$	✓	VM-Ti	**62.50**	<u>40.43</u>	**70.37**	**67.29**	**65.24**	**52.98**	**48.23**	<u>0.26</u>	**2.90**

Results. As illustrated in Figure 1, the linear complexity of VideoMamba makes it well-suited for end-to-end training with long-duration videos. The comparisons in Tables 6 and 7 highlight VideoMamba's simplicity and effectiveness against traditional feature-based methods [36,47] on these tasks. It yields significant performance improvements, achieving SOTA results even with smaller model sizes. For example, VideoMamba-Ti shows a notable increase of +**6.1%** over ViS4mer using Swin-B features and a +**3.0%** uplift against Turbo's multi-modality alignment approach [29]. Notably, the results underscore the positive impact of the scaling model and frame numbers for long-term tasks. In the diverse and challenging set of nine tasks presented by LVU, our VideoMamba-Ti, fine-tuned in an end-to-end manner, delivers outstanding or comparable results to current SOTA methods. These outcomes not only highlight VideoMamba's effectiveness but also its great potential for future long-video comprehension.

4.4 Multi-modality Video Understanding

Datasets and Settings. Following UMT [43], we utilize WebVid-2M [3] video-text pairs and CC3M [65] image-text pairs for joint pretraining with four ob-

Table 8: Zero-shot text-to-video retrieval on MSRVTT, DiDeMo, AcitivityNet, LSMDC, and MSVD. "BB" means the visual backbone. "#P" refers to the number of pretraining pairs. Models pretrained with large-scale pairs are noted in gray.

Method	BB	#P	MSRVTT			DiDeMo			ANet			LSMDC			MSVD		
			@1	@5	@10	@1	@5	@10	@1	@5	@10	@1	@5	@10	@1	@5	@10
Singularity [39]	Swin	5M	28.4	50.2	59.5	**36.9**	61.1	**69.3**	30.8	55.9	66.3	-	-	-	-	-	-
BridgeFormer [23]	ViT	5M	26.0	46.4	56.4	25.6	50.6	61.1	-	-	-	12.2	25.9	32.2	**43.6**	**74.9**	**84.9**
UMT [43]	ViT	5M	29.6	52.8	61.9	33.4	58.3	67.0	28.3	53.0	64.2	16.8	30.5	37.6	36.2	65.7	76.1
VideoMamba	VM	5M	**32.0**	**53.0**	**63.8**	36.6	**61.7**	**70.3**	**35.9**	**61.1**	**72.3**	**18.0**	**36.1**	**43.4**	38.0	68.6	79.0
Singularity [39]	Swin	17M	34.0	56.7	66.7	37.1	61.7	69.9	30.6	55.6	66.9	-	-	-	-	-	-
OmniVL [39]	ViT	17M	34.6	58.4	66.6	33.3	58.7	68.5	-	-	-	-	-	-	-	-	-
UMT [43]	ViT	17M	35.5	**59.3**	68.6	41.9	66.7	75.0	33.8	59.1	70.4	18.1	33.1	42.2	41.4	70.6	80.1
UMT [43]	ViT	25M	35.2	57.8	66.0	41.2	65.4	74.9	35.5	60.6	71.8	19.1	33.4	42.2	42.3	**71.7**	80.8
CLIP4Clip [55]	ViT	400M	30.6	54.4	64.3	-	-	-	-	-	-	13.6	27.9	35.5	36.2	63.8	73.5
InternVideo [85]	ViT	640M	40.0	65.3	74.1	31.5	57.6	68.2	30.7	57.4	70.2	17.6	32.4	40.2	43.4	69.9	79.1
VideoMamba	VM	17M	34.7	58.9	68.0	42.0	67.3	76.8	40.1	65.7	76.1	18.4	35.3	43.0	40.3	70.0	79.7
VideoMamba	VM	25M	**35.6**	58.1	**69.5**	**43.1**	**68.1**	**77.7**	**41.0**	**67.5**	**77.8**	**20.4**	**37.1**	**45.7**	**42.6**	71.6	**81.2**

jectives: vision-text contrastive learning [3], vision-text matching [40], masked language modeling [15] and unmasked token alignment [43]. Initially, we mask 50% image tokens and 80% video tokens, conducting pretraining across 8 frames for 10 epochs. Given Mamba's sensitivity to positional information, an additional unmasked tuning phase is carried out for one epoch to refine its comprehension further. For evaluation, we undertake zero-shot video-text retrieval tasks across five prominent benchmarks, including MSRVTT [87], DiDeMo [1], ActivityNet [31], LSMDC [62], and MSVD [10].

Results. As indicated in Table 8, under the same pretraining corpus and similar training strategies, our VideoMamba achieves superior zero-shot video retrieval performances to UMT [43] based on ViT [16]. It underscores Mamba's comparable efficiency and scalability to the ViT in handling multi-modal video tasks. Notably, for datasets featuring longer video lengths (*e.g.*, ANet and DiDeMo) and more complex scenarios (*e.g.*, LSMDC), VideoMamba demonstrates a significant improvement. This demonstrates Mamba's aptitude for the demands of cross-modality alignment even in challenging multimodal contexts.

5 Conclusion

In this paper, we propose VideoMamba, a purely SSM-based model for efficient video understanding. Our extensive experiments demonstrate its scalability in the visual domain, sensitivity for short-term action recognition, superiority in long-term video understanding and compatibility with other modalities. We hope it can pave the way for future model design for long-video comprehension.

Limitations. Due to resource constraints, we have not yet fully validated the scalability of VideoMamba, such as extending VideoMamba to larger sizes (*e.g.*, VideoMamba-g), incorporating additional modalities (*e.g.*, audio), and integrating with large language models for hour-level video understanding. Despite these limitations, our findings confirm VideoMamba's promising potential and we plan to conduct thorough explorations of its capabilities in the future.

Acknowledgements

This work was supported in part by the National Key R&D Program of China (No. 2022ZD0160505), and the National Natural Science Foundation of China under Grant (62272450, 62076119).

References

1. Anne Hendricks, L., Wang, O., Shechtman, E., Sivic, J., Darrell, T., Russell, B.: Localizing moments in video with natural language. In: ICCV (2017)
2. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C.: Vivit: A video vision transformer. In: ICCV (2021)
3. Bain, M., Nagrani, A., Varol, G., Zisserman, A.: Frozen in time: A joint video and image encoder for end-to-end retrieval. In: ICCV (2021)
4. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? In: ICML (2021)
5. Brooks, T., Peebles, B., Holmes, C., DePue, W., Guo, Y., Jing, L., Schnurr, D., Taylor, J., Luhman, T., Luhman, E., Ng, C., Wang, R., Ramesh, A.: Video generation models as world simulators (2024), https://openai.com/research/video-generation-models-as-world-simulators
6. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. In: NeurIPS (2020)
7. Carreira, J., Noland, E., Banki-Horvath, A., Hillier, C., Zisserman, A.: A short note about kinetics-600. ArXiv **abs/1808.01340** (2018)
8. Carreira, J., Noland, E., Hillier, C., Zisserman, A.: A short note on the kinetics-700 human action dataset. ArXiv **abs/1907.06987** (2019)
9. Carreira, J., Zisserman, A.: Quo vadis, action recognition? a new model and the kinetics dataset. In: CVPR (2017)
10. Chen, D.L., Dolan, W.B.: Collecting highly parallel data for paraphrase evaluation. In: ACL (2011)
11. Chen, G., Huang, Y., Xu, J., Pei, B., Chen, Z., Li, Z., Wang, J., Li, K., Lu, T., Wang, L.: Video mamba suite: State space model as a versatile alternative for video understanding. ArXiv **abs/2403.09626** (2024)
12. Cho, J.H., Hariharan, B.: On the efficacy of knowledge distillation. In: ICCV (2019)
13. Das, P., Xu, C., Doell, R.F., Corso, J.J.: A thousand frames in just a few words: Lingual description of videos through latent topics and sparse object stitching. In: CVPR (2013)
14. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR (2009)
15. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. ArXiv **abs/1810.04805** (2018)
16. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: ICLR (2021)
17. Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., Feichtenhofer, C.: Multiscale vision transformers. In: ICCV (2021)
18. Feichtenhofer, C.: X3d: Expanding architectures for efficient video recognition. In: CVPR (2020)

19. Feichtenhofer, C., Fan, H., Li, Y., He, K.: Masked autoencoders as spatiotemporal learners. NeurIPS (2022)
20. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: ICCV (2019)
21. Fu, D.Y., Dao, T., Saab, K.K., Thomas, A.W., Rudra, A., Ré, C.: Hungry hungry hippos: Towards language modeling with state space models. In: ICLR (2023)
22. Gao, J., Sun, C., Yang, Z., Nevatia, R.: Tall: Temporal activity localization via language query. In: ICCV (2017)
23. Ge, Y., Ge, Y., Liu, X., Li, D., Shan, Y., Qie, X., Luo, P.: Bridging video-text retrieval with multiple choice questions. In: CVP (2022)
24. Goyal, R., Kahou, S.E., Michalski, V., Materzynska, J., Westphal, S., Kim, H., Haenel, V., Fründ, I., Yianilos, P., Mueller-Freitag, M., Hoppe, F., Thurau, C., Bax, I., Memisevic, R.: The "something something" video database for learning and evaluating visual common sense. In: ICCV (2017)
25. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. ArXiv **abs/2312.00752** (2023)
26. Gu, A., Goel, K., Ré, C.: Efficiently modeling long sequences with structured state spaces. In: ICLR (2022)
27. Gu, C., Sun, C., Vijayanarasimhan, S., Pantofaru, C., Ross, D.A., Toderici, G., Li, Y., Ricco, S., Sukthankar, R., Schmid, C., Malik, J.: Ava: A video dataset of spatio-temporally localized atomic visual actions. CVPR (2017)
28. Guo, H., Li, J., Dai, T., Ouyang, Z., Ren, X., Xia, S.T.: Mambair: A simple baseline for image restoration with state-space model. ArXiv **abs/2402.15648** (2024)
29. Han, T., Xie, W., Zisserman, A.: Turbo training with token dropout. In: BMVC (2022)
30. He, X., Cao, K., Yan, K., Li, R., Xie, C., Zhang, J., Zhou, M.: Pan-mamba: Effective pan-sharpening with state space model. ArXiv **abs/2402.12192** (2024)
31. Heilbron, F.C., Escorcia, V., Ghanem, B., Niebles, J.C.: Activitynet: A large-scale video benchmark for human activity understanding. In: CVPR (2015)
32. Hu, V.T., Baumann, S.A., Gui, M., Grebenkova, O., Ma, P., Fischer, J.S., Ommer, B.: Zigma: A dit-style zigzag mamba diffusion model. In: ECCV (2024)
33. Hussein, N., Gavves, E., Smeulders, A.W.M.: Timeception for complex action recognition. In: CVPR (2019)
34. Hussein, N., Gavves, E., Smeulders, A.W.M.: Videograph: Recognizing minutes-long human activities in videos. ArXiv **abs/1905.05143** (2019)
35. Idrees, H., Zamir, A.R., Jiang, Y.G., Gorban, A., Laptev, I., Sukthankar, R., Shah, M.: The thumos challenge on action recognition for videos "in the wild". Computer Vision and Image Understanding (2017)
36. Islam, M.M., Bertasius, G.: Long movie clip classification with state-space video models. In: ECCV (2022)
37. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, A., Suleyman, M., Zisserman, A.: The kinetics human action video dataset. ArXiv **abs/1705.06950** (2017)
38. Kuehne, H., Arslan, A., Serre, T.: The language of actions: Recovering the syntax and semantics of goal-directed human activities. In: CVPR (2014)
39. Lei, J., Berg, T.L., Bansal, M.: Revealing single frame bias for video-and-language learning. ArXiv **abs/2206.03428** (2022)
40. Li, J., Selvaraju, R., Gotmare, A., Joty, S., Xiong, C., Hoi, S.C.H.: Align before fuse: Vision and language representation learning with momentum distillation. In: NeurIPS (2021)

41. Li, K., Li, X., Wang, Y., Wang, J., Qiao, Y.: Ct-net: Channel tensorization network for video classification. In: ICLR (2020)
42. Li, K., Wang, Y., He, Y., Li, Y., Wang, Y., Wang, L., Qiao, Y.: Uniformerv2: Spatiotemporal learning by arming image vits with video uniformer. In: ICCV (2023)
43. Li, K., Wang, Y., Li, Y., Wang, Y., He, Y., Wang, L., Qiao, Y.: Unmasked teacher: Towards training-efficient video foundation models. In: ICCV (2023)
44. Li, K., Wang, Y., Peng, G., Song, G., Liu, Y., Li, H., Qiao, Y.: Uniformer: Unified transformer for efficient spatial-temporal representation learning. In: ICLR (2022)
45. Li, Y., Wu, C., Fan, H., Mangalam, K., Xiong, B., Malik, J., Feichtenhofer, C.: Improved multiscale vision transformers for classification and detection. ArXiv **abs/2112.01526** (2021)
46. Liang, D., Zhou, X., Wang, X., Zhu, X., Xu, W., Zou, Z., Ye, X., Bai, X.: Pointmamba: A simple state space model for point cloud analysis. ArXiv **abs/2402.10739** (2024)
47. Lin, X., Petroni, F., Bertasius, G., Rohrbach, M., Chang, S.F., Torresani, L.: Learning to recognize procedural activities with distant supervision. CVPR (2022)
48. Liu, H., Yan, W., Zaharia, M., Abbeel, P.: World model on million-length video and language with ringattention. ArXiv **abs/2402.08268** (2024)
49. Liu, Y., Wang, L., Wang, Y., Ma, X., Qiao, Y.: Fineaction: A fine-grained video dataset for temporal action localization. TIP (2022)
50. Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., Ye, Q., Liu, Y.: Vmamba: Visual state space model. ArXiv **abs/2401.10166** (2024)
51. Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., Guo, B.: Swin transformer: Hierarchical vision transformer using shifted windows. In: ICCV (2021)
52. Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H.: Video swin transformer. In: CVPR (2022)
53. Liu, Z., Mao, H., Wu, C., Feichtenhofer, C., Darrell, T., Xie, S.: A convnet for the 2020s. In: CVPR (2022)
54. Lu, J., Batra, D., Parikh, D., Lee, S.: Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. NeurIPS (2019)
55. Luo, H., Ji, L., Zhong, M., Chen, Y., Lei, W., Duan, N., Li, T.: Clip4clip: An empirical study of clip for end to end video clip retrieval and captioning. Neurocomputing (2022)
56. Ma, J., Li, F., Wang, B.: U-mamba: Enhancing long-range dependency for biomedical image segmentation. ArXiv **abs/2401.04722** (2024)
57. Mehta, H., Gupta, A., Cutkosky, A., Neyshabur, B.: Long range language modeling via gated state spaces. ArXiv **abs/2206.13947** (2022)
58. Miech, A., Zhukov, D., Alayrac, J.B., Tapaswi, M., Laptev, I., Sivic, J.: Howto100m: Learning a text-video embedding by watching hundred million narrated video clips. In: ICCV (2019)
59. Nguyen, E., Goel, K., Gu, A., Downs, G.W., Shah, P., Dao, T., Baccus, S.A., Ré, C.: S4nd: Modeling images and videos as multidimensional signals with state spaces. In: NeurIPS (2022)
60. Patrick, M., Campbell, D., Asano, Y., Misra, I., Metze, F., Feichtenhofer, C., Vedaldi, A., Henriques, J.F.: Keeping your eye on the ball: Trajectory attention in video transformers. In: NeurIPS (2021)
61. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: ICML (2021)

62. Rohrbach, A., Torabi, A., Rohrbach, M., Tandon, N., Pal, C.J., Larochelle, H., Courville, A.C., Schiele, B.: Movie description. International Journal of Computer Vision (2016)
63. Shao, D., Zhao, Y., Dai, B., Lin, D.: Finegym: A hierarchical video dataset for fine-grained action understanding. CVPR (2020)
64. Sharir, G., Noy, A., Zelnik-Manor, L.: An image is worth 16x16 words, what is a video worth? ArXiv **abs/2103.13915** (2021)
65. Sharma, P., Ding, N., Goodman, S., Soricut, R.: Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In: ACL (2018)
66. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. NeurIPS (2014)
67. Smith, J.T., Warrington, A., Linderman, S.W.: Simplified state space layers for sequence modeling. In: ICLR (2023)
68. Soomro, K., Zamir, A.R., Shah, M.: Ucf101: A dataset of 101 human actions classes from videos in the wild. arXiv preprint arXiv:1212.0402 (2012)
69. Su, J., Lu, Y., Pan, S., Wen, B., Liu, Y.: Roformer: Enhanced transformer with rotary position embedding. ArXiv **abs/2104.09864** (2021)
70. Sun, C., Myers, A., Vondrick, C., Murphy, K., Schmid, C.: Videobert: A joint model for video and language representation learning. In: ICCV (2019)
71. Sun, Y., Dong, L., Huang, S., Ma, S., Xia, Y., Xue, J., Wang, J., Wei, F.: Retentive network: A successor to transformer for large language models. ArXiv **abs/2307.08621** (2023)
72. Tang, Y., Ding, D., Rao, Y., Zheng, Y., Zhang, D., Zhao, L., Lu, J., Zhou, J.: Coin: A large-scale dataset for comprehensive instructional video analysis. In: CVPR (2019)
73. Team, G.: Gemini: A family of highly capable multimodal models. ArXiv **abs/2312.11805** (2023)
74. Team, R.: Rwkv: Reinventing rnns for the transformer era. In: EMNLP (2023)
75. Tong, Z., Song, Y., Wang, J., Wang, L.: VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In: NeurIPS (2022)
76. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., J'egou, H.: Training data-efficient image transformers & distillation through attention. In: ICML (2021)
77. Tran, D., Bourdev, L.D., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: IEEE International Conference on Computer Vision (2015)
78. Tran, D., xiu Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: CVPR (2018)
79. Wang, C., Tsepa, O., Ma, J., Wang, B.: Graph-mamba: Towards long-range graph sequence modeling with selective state spaces. ArXiv **abs/2402.00789** (2024)
80. Wang, J., Yan, J.N., Gu, A., Rush, A.M.: Pretraining without attention. ArXiv **abs/2212.10544** (2022)
81. Wang, L., Tong, Z., Ji, B., Wu, G.: TDN: Temporal difference networks for efficient action recognition. In: CVPR (2021)
82. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Gool, L.V.: Temporal segment networks: Towards good practices for deep action recognition. In: ECCV (2016)
83. Wang, R., Chen, D., Wu, Z., Chen, Y., Dai, X., Liu, M., Jiang, Y.G., Zhou, L., Yuan, L.: Bevt: Bert pretraining of video transformers. CVPR (2022)

84. Wang, Y., He, Y., Li, Y., Li, K., Yu, J., Ma, X.J., Chen, X., Wang, Y., Luo, P., Liu, Z., Wang, Y., Wang, L., Qiao, Y.: Internvid: A large-scale video-text dataset for multimodal understanding and generation. In: ICLR (2024)
85. Wang, Y., Li, K., Li, Y., He, Y., Huang, B., Zhao, Z., Zhang, H., Xu, J., Liu, Y., Wang, Z., Xing, S., Chen, G., Pan, J., Yu, J., Wang, Y., Wang, L., Qiao, Y.: Internvideo: General video foundation models via generative and discriminative learning. ArXiv **abs/2212.03191** (2022)
86. Wu, C.Y., Krahenbuhl, P.: Towards long-form video understanding. In: CVPR (2021)
87. Xu, J., Mei, T., Yao, T., Rui, Y.: Msr-vtt: A large video description dataset for bridging video and language. In: CVPR (2016)
88. Yang, Y., Xing, Z., Zhu, L.: Vivim: a video vision mamba for medical video object segmentation. ArXiv **abs/2401.14168** (2024)
89. Yu, Z., Xu, D., Yu, J., Yu, T., Zhao, Z., Zhuang, Y., Tao, D.: Activitynet-qa: A dataset for understanding complex web videos via question answering. In: AAAI (2019)
90. Zhang, D.J., Li, K., Wang, Y., Chen, Y., Chandra, S., Qiao, Y., Liu, L., Shou, M.Z.: Morphmlp: An efficient mlp-like backbone for spatial-temporal representation learning. In: ECCV (2022)
91. Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., Wang, X.: Vision mamba: Efficient visual representation learning with bidirectional state space model. ArXiv **abs/2401.09417** (2024)
92. Zhuang, S., Li, K., Chen, X., Wang, Y., Liu, Z., Qiao, Y., Wang, Y.: Vlogger: Make your dream a vlog. ArXiv **abs/2401.09414** (2024)