

Video Mamba Suite: State Space Model as a Versatile Alternative for Video Understanding

Guo Chen^{1†}, Yifei Huang^{2†}, Jilan Xu^{3†}, Baoqi Pei^{4,2†}, Zhe Chen¹,
Zhiqi Li¹, Jiahao Wang¹, Kunchang Li², Tong Lu^{1*}, and Limin Wang^{1,2*}

¹Nanjing University ²OpenGVLab, Shanghai AI Laboratory
³Fudan University ⁴Zhejiang University

Abstract. Understanding videos is one of the fundamental directions in computer vision research, with extensive efforts dedicated to exploring various architectures such as RNN, 3D CNN, and Transformers. The newly proposed architecture of state space model, *e.g.*, Mamba, shows promising traits to extend its success in long sequence modeling to video modeling. **To assess whether Mamba can be a viable alternative to Transformers in the video understanding domain**, in this work, we conduct a comprehensive set of studies, probing different roles Mamba can play in modeling videos, while investigating diverse tasks where Mamba could exhibit superiority. **We categorize Mamba into four roles for modeling videos, deriving a Video Mamba Suite composed of 14 models/modules**, and evaluating them on 12 video understanding tasks. Our extensive experiments reveal the strong potential of Mamba on both video-only and video-language tasks while showing promising efficiency-performance trade-offs. We hope this work could provide valuable data points and insights for future research on video understanding. Code is public: <https://github.com/OpenGVLab/video-mamba-suite>.

mamba作四个角色
测评了mamba组成的14个模块

Keywords: Video Understanding · State Space Model · Mamba

1 Introduction

Video understanding is a foundation problem in computer vision research, which requires capturing spatial-temporal dynamics from videos to localize activities or infer their evolution. Current explorations on architectures for video understanding can be divided into three categories. The first line of work employs frame-based feature encoding followed by temporal dependency modeling via recurrent networks such as GRU and LSTM. Since this type of divided space-time modeling cannot capture joint spatiotemporal information, another stripe of work uses 3D kernels in convolutional neural networks to simultaneously consider spatial and temporal correlations [25, 73].

Following the triumph of language [62, 63, 70, 76] and image [5, 18, 34, 61, 69] transformers, video transformers [2, 68, 80] have also made significant strides in

1. 特征编码+LSTM的时间建模
无法关注时间空间信息

[†] Equal key contributions by Guo Chen, Yifei Huang, Jilan Xu, and Baoqi Pei.

^{*} Tong Lu and Limin Wang are joint corresponding authors.

2. 使用3D卷积考虑时间空间信息

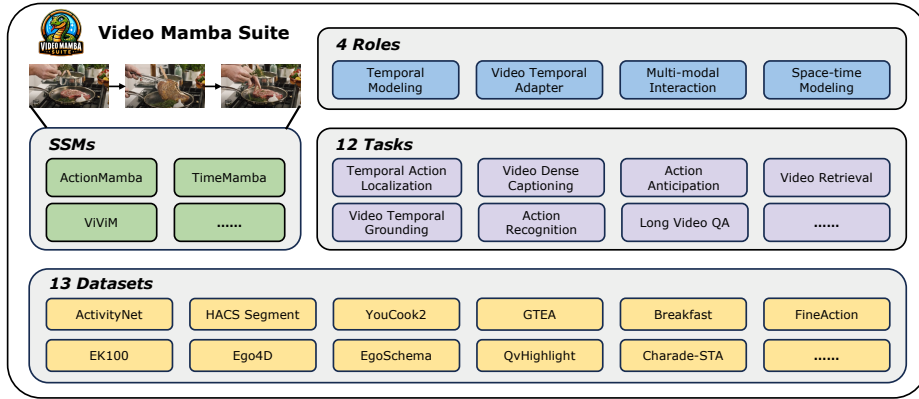



Fig. 1: We investigate SSMs exemplified by Mamba on video understanding. Our **Video Mamba Suite** comprises 14 SSM models/modules for 12 video understanding tasks. We explore 4 roles of SSM in video modeling and conduct extensive experiments on 13 major datasets.

video understanding, exhibiting stronger capabilities compared to RNNs and 3D-CNNs. Video transformers encapsulate videos within a sequence of tokens, and the attention mechanism can enable global context interaction and data-dependent dynamic computation. Consequently, the model is adept at handling temporal [10, 78, 89, 90] or spatial-temporal [6, 68] information within videos in a unified manner. Due to the limited computational efficiency of video transformers in long videos, several variants [2, 6, 20, 55] have emerged, balancing the speed-performance trade-off.

Recently, State Space Models (SSMs) have demonstrated their advantages in Natural Language Processing (NLP). Modern SSMs [32] exhibit strong representation ability in NLP especially in long sequence modeling, while maintaining linear-time complexity. This is because their selection mechanism can eliminate the need to store the complete context. Notably, Mamba [30], incorporates time-varying parameters into the SSM and proposes a hardware-aware algorithm to enable highly efficient training and inference. The impressive scaling performance of Mamba indicates that it is a promising alternative to transformers. Meanwhile, Mamba’s strong performance and efficiency make it extremely suitable for video understanding tasks. However, despite some initial attempts to explore how Mamba can be applied in image modeling [53, 96], its effectiveness in video understanding remains unclear. The absence of a comprehensive study on Mamba’s potential for video understanding hinders further exploration of its capabilities in the diverse range of video-related tasks.

In this paper, we do not propose a novel method. Instead, we conduct an extensive investigation into the potential of SSM exemplified by Mamba in the context of video understanding. Our objective is to assess whether Mamba can be a viable alternative to transformers in this domain. To achieve this, we explore different roles Mamba can potentially play in understanding videos, as well as investigate diverse tasks where Mamba may exhibit superiority. We catego-

size Mamba into four roles for modeling videos: 1) temporal model, 2) temporal module, 3) multi-modal interaction network, and 4) spatial-temporal model. For each role, we investigate its video modeling capabilities on different video understanding tasks. To ensure a fair comparison with transformers, we carefully select counterpart models based on either standard or improved transformer architectures. Our exploration derives a **Video Mamba Suite** , comprising 14 models/modules for 12 video understanding tasks. We hope our **Video Mamba Suite** can serve as a valuable resource for future explorations of SSM-based models in the field of video understanding.

2 Related Work


2.1 Video Modeling

Video modeling serves as the cornerstone for achieving deep video understanding and has undergone significant development over time. Initially, TSN [82] employed uniform video sampling and utilized 2D networks [35, 40] to establish multi-frame consensus for generating video representations. After that, substantial advances in video convolutional networks have been made. Approaches like [48, 75, 81] focused on integrating temporal modules into 2D networks to facilitate spatial-temporal modeling. Conversely, methods such as [9, 25, 60, 73, 74, 87] either augment the kernels of 2D convolutional networks [35, 40, 86] or train 3D convolutional networks from scratch to extract spatial-temporal features.

Nevertheless, convolutional networks are constrained by their reliance on static local operators, resulting in limited representation capacity. Inspired by the effectiveness of language transformers [8, 17, 62, 63, 70] and image transformers [11, 18, 20, 34, 54, 69, 84], various structures of video transformers have been investigated by researchers [2, 6, 45, 55]. Notably, the video transformer incorporating spatial-temporal joint attention demonstrates superior capabilities among these structures. Consequently, subsequent research efforts [24, 46, 68, 80, 85] delve into different pre-training methodologies based on the structure. However, the quadratic computational complexity associated with joint attention poses a significant obstacle, impeding the scalability of video transformers for handling longer contexts, which is similar to what issue LLMs [71, 72] face. Several studies have developed transformer-based variants [2, 6, 55] tailored for modeling long-form videos frequently consisting of more than 128 frames.

Alternatively, another approach is to design model architectures with linear complexity. For instance, RetNet [67] and RWKV [59] utilized exponential decay techniques to capture global information. State space models [31–33] also offer linear complexity, with Mamba [30] employing efficient implementations to facilitate data-dependent inference. In the vision field, some works [1, 19, 96] have explored vision encoders with linear complexity. XCiT [1] achieved global interaction with linear complexity by calculating the cross-variance between input tokens. Recently, [53, 96] has conducted an initial exploration of a Mamba-based vision application. In our study, we investigate the creation of a Mamba-based video model for video understanding.

2.2 State-Space Models (SSMs)

As a frontrunner in the era of State-Space Models (SSMs), [32] introduced a novel model called the Structured State-Space Sequence (S4), which offers an alternative to CNNs and transformers for capturing long-range dependencies. The S4 model exhibits a promising characteristic of linearly scaling with sequence length. Building on this, [65] proposed an advanced layer dubbed S5, which integrates MIMO SSM with efficient parallel scanning into the S4 architecture. This development seeks to overcome the constraints of SSMs, enhancing their efficacy. Furthermore, [26] contributed a novel SSM layer, H3, significantly closing the performance gap between SSMs and transformer-based attention in language modeling. [57] extended the S4 model by introducing additional gating units in the Gated State Space layer to enhance its expressivity. More recently, [30] introduced a data-dependent SSM layer and developed a versatile language model called Mamba. Mamba outperforms transformers in performance across various sizes of large-scale real data and demonstrates linear scaling with sequence length. In this study, we aim to extend the success of Mamba in the language domain to video understanding. Specifically, we construct a versatile framework, termed **Video Mamba Suite** , to develop, validate, and analyze the performance of Mamba for video understanding.

There are only limited works that use SSMs for video understanding, with the primary focus on long-term video classification [41, 79]. [41] expanded the S4 architecture and incorporated a multi-scale temporal S4 decoder into video sequence classification. [79] introduced a masked contrastive learning framework to augment state-space models. Our study extends the application scope of SSMs to a much broader range of video understanding tasks. We cover multiple usages of SSMs and more video-related tasks, using Mamba [30] as an example, moving beyond the sole focus on the classification of long videos.

3 Preliminaries

3.1 State Space Model

In this section, we introduce models based on Structured State-Space (SSM), specifically the S4 [32] and Mamba [30] models. These models draw inspiration from continuous systems that process sequences or functions. Imagine a system that takes a sequence of inputs over time, $x(t)$, and produces a sequence of outputs, $y(t)$, while transforming the input through a hidden state, $h(t)$.

This system uses matrices $\mathbf{A}^{N \times N}$ to define the evolution of the hidden state, and $\mathbf{B}^{N \times 1}$ and $\mathbf{C}^{1 \times N}$ to project the input and hidden state to the output, respectively. This process can be summarized as $h'(t) = \mathbf{A}h(t) + \mathbf{B}x(t)$, $y(t) = \mathbf{C}h(t)$.

S4 and Mamba are designed as discrete equivalents of these continuous systems. They incorporate a timescale parameter Δ , to convert the continuous parameters (\mathbf{A}, \mathbf{B}) into their discrete counterparts $(\bar{\mathbf{A}}, \bar{\mathbf{B}})$. The conversion uses a technique, zero-order hold, resulting in the discrete formulas, $\bar{\mathbf{A}} = \exp(\Delta\mathbf{A})$, $\bar{\mathbf{B}} = (\Delta\mathbf{A})^{-1}(\exp(\Delta\mathbf{A}) - \mathbf{I}) \cdot \Delta\mathbf{B}$. Once the parameters are discretized, the system's

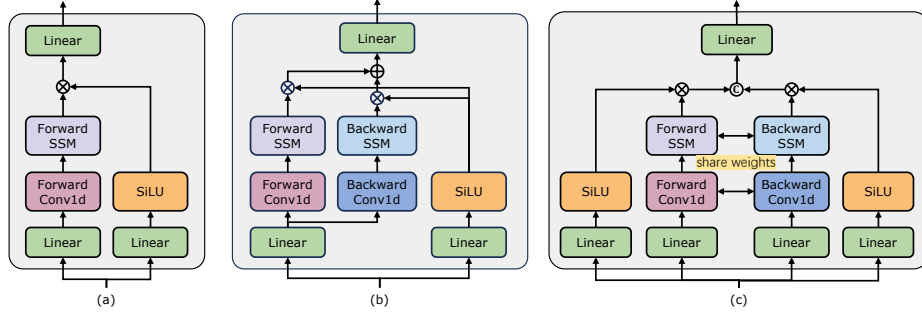


Fig. 2: Illustration of three SSMs blocks. (a) is the vanilla Mamba block [30]. (b) is the ViM block [96]. (c) is our proposed DBM block, which separates the input projector and shares the parameters of SSM in both scanning directions.

behavior at each discrete time step t is given by $h_t = \bar{\mathbf{A}}h_{t-1} + \bar{\mathbf{B}}x_t, y_t = \mathbf{C}h_t$, showcasing how the hidden state updates and produces output.

Finally, the output is computed through a global convolution process, involving a structured convolutional kernel, $\bar{\mathbf{K}} = (\mathbf{C}\bar{\mathbf{B}}, \mathbf{C}\bar{\mathbf{A}}\bar{\mathbf{B}}, \dots, \mathbf{C}\bar{\mathbf{A}}^{M-1}\bar{\mathbf{B}})$. This kernel is constructed from the transformed projection parameters and applied across the entire input sequence \mathbf{x} to produce the final output sequence \mathbf{y} , *i.e.* $\mathbf{y} = \mathbf{x} * \bar{\mathbf{K}}$. Here, M is the length of the input sequence \mathbf{x} . By transforming continuous dynamics into discrete steps, S4 and Mamba models allow for the processing of sequences with complex dependencies.

3.2 Mamba Block

Mamba block [30] combines a linear attention operator and an MLP block, inspired by the gated attention unit (GAU) [38], as illustrated in Figure 2 (a). This architecture involves expanding the model dimension D by a controllable expansion factor E . For each block, most of the parameters ($3ED^2$) are in the linear projections ($2ED^2$ for input projections, ED^2 for output projection) while the inner SSM contributes less. The number of SSM parameters (projections for Δ , \mathbf{B} , \mathbf{C} , and the matrix \mathbf{A}) are much smaller in comparison.

3.3 ViM Block

The ViM block [96], as shown in Figure 2 (b), adds a backward selective scanning branch with additional parameters, to the Mamba block. Based on this, features for both scanning directions share the same linear projection and gated layers. Given the input \mathbf{x} , for each direction, the ViM block first applies 1D convolution to the input \mathbf{x} and gets \mathbf{x}'_o . Then the block linearly projects \mathbf{x}'_o to $\mathbf{B}_o, \mathbf{C}_o, \Delta_o$, respectively. Δ_o is then used to transform the $\bar{\mathbf{A}}_o$ and $\bar{\mathbf{B}}_o$, respectively. Next, SSM computes forward-scanned features \mathbf{y}_f and backward-scanned features \mathbf{y}_b . Finally, both features are gated by the gated layer and averaged to get the output token sequence.

Method	Block	mAP@0.5	mAP@0.75	mAP@0.95	mAP@Avg
ActionFormer [90]	Window Attn [12]	62.62	44.61	12.73	43.34
ActionMamba [96]	ViM [96]	63.78	45.45	13.01	44.26
ActionMamba	DBM (ours)	64.02	45.71	13.34	44.56

Table 1: Results of temporal action localization on HACS Segment [91]. The metric is mean Average Precision (mAP) under multiple tIoU thresholds {0.5, 0.75, 0.95}.


Method	Block	Acc	Edit	F1@10	F1@25	F1@50
MS-TCN [21]	Dilated Conv, Enc	76.3	79.0	85.8	83.4	69.8
ASFormer [89]	Window Attn [12], Enc-Dec	79.7	84.6	90.1	88.8	79.2
ASFormer [†] [89]	Window Attn [12], Enc-Dec	77.1	81.0	86.2	84.8	77.0
ASFormer [†] [89]	Window Attn [12], Enc	75.4	78.1	82.7	80.5	68.4
ASMamba [96]	ViM [96], Enc	79.3	87.0	90.3	89.0	77.9
ASMamba	DBM (ours), Enc	78.4	87.5	91.1	89.8	79.7

Table 2: Results of temporal action segmentation on GTEA [22] dataset. The metrics are accuracy, edit distance [7], and instance-wise F1 under multiple tIoU thresholds {0.1, 0.25, 0.5}. † denotes our reproduced results with its official code.

3.4 DBM Block

We further investigate a Decomposed Bidirectionally Mamba block, denoted as DBM, as depicted in Figure 2 (c). It employs an inverse design compared to the ViM block. Given the input sequence \mathbf{x} , the DBM block initially employs distinct linear layers to segregate forward features \mathbf{x}_f and backward features \mathbf{x}_b . These features are then passed through the SSM module with shared parameters for bidirectional scanning, resulting in \mathbf{x}'_f and \mathbf{x}'_b . Subsequently, both features undergo gating by two distinct layers before being concatenated to generate the output token sequence. This block introduces directional bias and weakens specific dynamics. In our experiments detailed in Section 4.1, we observe that the DBM block demonstrates improved performance on small-scale datasets.

4 Video Mamba Suite

In this section, we present **Video Mamba Suite** , which encompasses various Mamba-based models for a diverse set of video understanding tasks. The suite leverages Mamba in four distinct roles: temporal models, temporal modules, multi-modal interaction models, and space-time sequence models. Each role will be discussed in a separate subsection, gradually revealing Mamba’s performance in video understanding and highlighting its key characteristics. Due to space limitations, more implementation details and experimental results are provided in the supplementary material.

4.1 Mamba for Video Temporal Modeling

Tasks and datasets. We assess Mamba’s performance across five video temporal tasks: temporal action localization (HACS Segment [91]), temporal action

Method	Block	ActivityNet				YouCook2			
		B-4	M	C	SODA	B-4	M	C	SODA
PDVC [83]	DeformAttn [97]	1.75	6.73	26.07	5.47	0.73	4.25	20.48	4.02
PDVC [83]	ViM [96]	1.68	6.92	26.26	5.33	0.71	4.32	20.59	4.09
PDVC	DBM (ours)	1.76	7.16	26.77	5.27	0.86	4.44	22.11	4.32

Table 3: Results of dense video captioning on ActivityNet [36] and YouCook2 [95]. The metrics include BLEU-4 [58], METEOR [4], CIDEr [77] and SODA_c [27].

Method	Block	ActivityNet				YouCook2			
		B-4	M	R	C	B-4	M	R	C
PDVC [83]	DeformAttn [97]	9.07	12.52	29.02	13.12	6.44	12.94	28.74	12.48
PDVC [83]	ViM [96]	9.33	13.52	29.83	14.25	6.50	12.96	28.59	13.08
PDVC	DBM (ours)	9.05	14.05	29.86	14.43	7.24	13.47	29.09	13.03

Table 4: Results of video paragraph captioning on ActivityNet [36] and YouCook2 [95]. The metrics include BLEU-4 [58], METEOR [4], ROUGE-L [47], CIDEr [77].

segmentation (GTEA [22]), dense video captioning (ActivityNet [36], YouCook [95]), video paragraph captioning (ActivityNet [36], YouCook [95]), and action anticipation (Epic-Kitchen-100 [13]).

Baseline and competitor. We select transformer-based counterparts as the baseline of each task. Specifically, the transformer baselines are ActionFormer [90], ASFormer [89], Testra [92], and PDVC [83]. To build a Mamba challenger, we replace transformer blocks of the baseline model with Mamba-based blocks, including vanilla Mamba [30], ViM [96], and our DBM. Note that in the context of action anticipation, which involves causal inference, we compare the performance of the baseline with the vanilla Mamba block [30].

Results and analysis. We present the comparison results of different models for four tasks in Table 1 to Table 5. Overall speaking, while some transformer-based models have incorporated attention variants to enhance performance, the tables demonstrate the superior performance of the Mamba series compared to existing methods in the transformer series.

Temporal action localization. As depicted in Table 1, our introduced ActionMamba, which is based on changing Transformer blocks of ActionFormer [90] to ViM blocks, achieves an average mAP of 44.26 on the HACS Segment dataset. Using our DBM blocks, ActionMamba can further improve the average mAP to 44.56. This significantly outperforms its transformer counterpart by 1.22 (44.56 *vs.* 43.34).

Temporal action segmentation. Table 2 reports the results of temporal action segmentation. Our proposed ASMamba demonstrates superior performance compared to our reproduced ASFormer [89]. Since ASMamba is a pure encoder while ASFormer adopts the encoder-decoder structure, we also experiment with an encoder version of ASFormer, where a drop in performance is observed. The results suggest the great potential of Mamba compared with the Transformer.

Method	Block	Overall			Unseen			Tail		
		Ver	Nou	Act	Ver	Nou	Act	Ver	Nou	Act
Testra ⁺ [92]	long short Attn [76]	30.8	35.8	17.6	29.6	26.0	12.8	23.2	29.2	14.2
MAT ⁺ [78]	long short Attn [76]	35.0	38.8	19.5	32.5	30.3	13.8	28.7	33.1	16.9
Testra [92]	short Attn [76]	25.1	30.8	14.1	24.3	24.5	10.7	17.4	23.0	10.9
Testra	short Mamba [30]	27.9	34.1	15.2	28.1	24.2	12.0	20.5	27.8	12.3

Table 5: Results of action anticipation on EK-100 [13]. Accuracy measured by class-mean recall@5(%) following the standard protocol. ⁺ denotes the model is trained with data augmentation. “long” and “short” denote using long- and short-term memory.

Dense video captioning. Table 3 shows the results on dense video captioning. The baseline PDVC model [83] adopts a Deformable Transformer to encode visual information. In comparison, our bidirectional Mamba with DBM block shows stronger performances on both temporal event localization and caption generation.

Video paragraph captioning. Table 4 demonstrates the result comparison on video paragraph captioning. We also adopt the PDVC as the baseline model [83], by training the model with captioning loss only. Different from the dense video captioning task where both captioning and localization are critical, video paragraph captioning merely focuses on extracting fine-grained visual information for generating captions. Results in Table 4 reveal our bidirectional Mamba brings stronger feature representation for captioning, compared with the temporal deformable encoder.

Action anticipation. We further assess the Mamba’s ability in causal modeling via the action anticipation task. By considering 5-second temporal features as input, we compare Testra [92] with causal self-attention blocks and causal Mamba [30] blocks. As presented in Table 5, the results demonstrate the superior causal reasoning capability of Mamba, which aligns with the conclusion in text generation of Table 4.

4.2 Mamba for Cross-Modal Interaction

Tasks and datasets. In addition to single-modal tasks, we assess the performance of Mamba for cross-modal interaction. We first employ the video temporal grounding (VTG) task for evaluation. The involved datasets contain QvHighlight [44] and Charade-STA [28].

Baseline and competitor. In this work, we use UniVTG [50] to create our mamba-based VTG model. UniVTG employs a transformer as the multi-modal interaction network. Given video feature $\mathbf{V} = \{\mathbf{v}_i\}_{i=1}^{L_v} \in \mathbb{R}^{L_v \times D}$ and text feature $\mathbf{Q} = \{\mathbf{q}_j\}_{j=1}^{L_q} \in \mathbb{R}^{L_q \times D}$, we first add learnable position embeddings \mathbf{E}^{pos} and modality-type embeddings \mathbf{E}^{type} to each modality to retain both positional and modality information:

Method	Block	Qvhighlight							Charade-STA		
		R1		mAP			HD (\geq VG)		R1		
		@0.5	@0.7	@0.5	@0.75	Avg.	mAP	HIT@1	@0.3	@0.5	@0.7
MDETR [44]	-	52.89	33.02	54.82	29.40	30.73	35.69	55.60	65.83	52.07	30.59
UMT [51]	-	56.23	41.18	53.83	37.01	36.12	38.18	59.99	-	-	-
UniVTG [50]	Trans	58.86	40.86	57.60	35.59	35.47	38.20	60.96	70.81	58.01	35.65
UniVTG†	Trans	59.87	44.32	59.09	40.25	38.48	39.22	64.71	68.20	57.26	34.68
UniVTG [50]	ViM	65.55	50.77	63.97	46.11	44.74	40.22	64.26	68.12	57.07	35.83
UniVTG [50]	DBM (ours)	66.65	52.19	64.37	46.68	45.18	40.18	64.77	68.06	57.18	36.05

Table 6: Results of video temporal grounding tasks on Qvhighlight [44] and Charade-STA [28]. UniVTG [50] default adopts a 4-layer transformer for cross-modal interaction. † denotes the reproduced results of a 6-layer transformer.

Method	Qvhighlight		Charade-STA	
	R@0.5	R@0.7	R@0.5	R@0.7
L	66.97	51.23	56.29	35.99
R	62.06	44.84	46.88	25.89
L+R	65.10	50.91	55.65	34.30
M	61.87	44.06	46.56	25.65

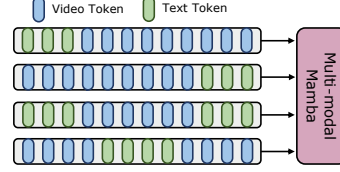


Table 7: Effect of the position of text tokens in the whole token sequence. **Fig. 3:** Illustration for different positions of video and text tokens in the input sequence.

$$\begin{aligned}\tilde{\mathbf{V}} &= \mathbf{V} + \mathbf{E}_V^{pos} + \mathbf{E}_V^{type}, \\ \tilde{\mathbf{Q}} &= \mathbf{Q} + \mathbf{E}_Q^{pos} + \mathbf{E}_Q^{type}.\end{aligned}\tag{1}$$

Then, the text and video tokens are concatenated to get a joint input $\mathbf{Z} = [\tilde{\mathbf{V}}; \tilde{\mathbf{Q}}] \in \mathbb{R}^{L \times D}$, where $L = L_v + L_q$. Further, \mathbf{Z} is fed into a multi-modal transformer encoder. Finally, the text-enhanced video features $\tilde{\mathbf{V}}^e$ are taken out and then fed into the prediction head. To create a cross-modal Mamba competitor, we stack bidirectional Mamba blocks to form a multi-modal Mamba encoder to replace the transformer baseline.

Results and analysis. We present the performance of multiple models on Qvhighlight [44] in Table 6. Mamba achieves an average mAP of 44.74, representing a significant improvement compared to the transformer (44.74 *vs.* 38.48). For Charade-STA [28], the Mamba-based method also achieves comparable performance. This suggests that Mamba has the potential to integrate multiple modalities effectively. Given that Mamba [30] is a model based on linear scanning, while the transformer is based on global token interaction, intuitively, we believe that the position of text in the token sequence may influence the effectiveness of multi-modal aggregation. To investigate this, we include different text-visual fusion methods in Table 7, while Figure 3 illustrates four different arrangements of tokens. We observe that the best results are obtained when textual conditions are fused on the left side of the visual features. Qvhighlight [44]

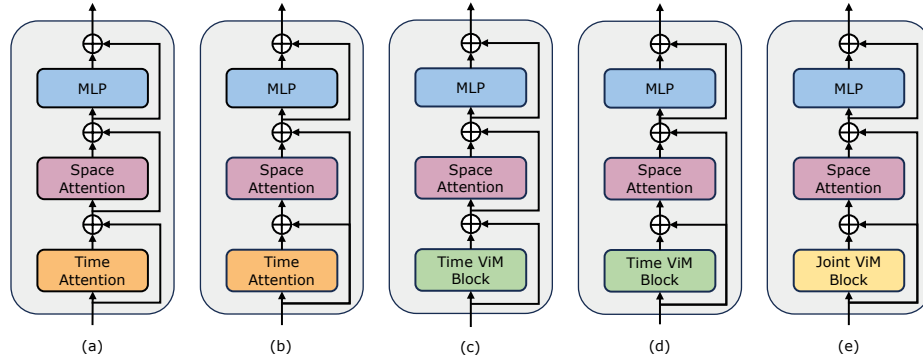


Fig. 4: Illustration of our explored structures. (a) and (b) shows vanilla-style [6] and frozen-style [3] residual connection forms for TimeSformer [6]. (c) and (d) presents our created TimeMamba which uses ViM block as a temporal module in both styles. (e) provides the replacement of the temporal ViM block with a space-time ViM block.

is less affected by this fusion, whereas Charade-STA [28] shows particular sensitivity to the position of the text, which may be attributed to the characteristics of the dataset.

Results and analysis. The performance comparison on TimeMamba and TimeSformer [6] is shown in Table 8, Table 9, Table 10 and Figure 5.

4.3 Mamba as Video Temporal Adapter

Tasks and datasets. In addition to evaluating Mamba’s performance in post-temporal modeling, we also assess its effectiveness as a video-temporal adapter. We pre-train a dual-tower model by performing video-text contrastive learning on the egocentric data [29, 49], which contains 4M video clips with fine-grained narrations. For evaluation, we consider zero-shot/fine-tuned multi-instance retrieval and fine-tuned action recognition on the Epic-Kitchens-100 dataset [13], and zero-shot long-form Question Answering on the EgoSchema dataset [56].

Baseline and competitor. TimeSformer [6] adopts the divided space-time attention block to model the spatial and temporal relationship in the video separately. Following TimeSformer, we introduce bidirectional Mamba blocks as temporal adapters to replace the vanilla temporal self-attention for improved divided space-time interaction. The space attention layers in the TimeSformer remain unchanged for fair comparison.

Here, we use ViM [96] block as the temporal module, and term the resulting model as TimeMamba. For consistency, we re-implement the transformer baseline and employ a consistent adaptation method, which involves adding a \tanh -gating mechanism [37] with an initial value of zero. This ensures that the output of the new model matches that of the original model. Notably, a standard ViM block has more parameters (slightly more than $6.25C^2$) than a self-attention

ID	Model	Adaptation	mAP			nDCG		
			V2T	T2V	Avg	V2T	T2V	Avg
1	EgoVLP [49]	Frozen [3], Attn in Time	19.4	13.9	16.6	24.1	22.0	23.1
2	EgoVLP [93]	Frozen [3], Attn in Time	26.0	20.6	23.3	28.8	27.0	27.9
3	LaViLa [94]	Frozen [3], Attn in Time	-	-	26.0	-	-	28.8
4	TimeSformer	Vanilla [6], Attn in Time	29.2	21.8	25.5	30.1	27.1	28.6
5	TimeSformer	Frozen [3], Attn in Time	29.8	22.2	26.0	30.6	27.5	29.0
6	TimeMamba (ours)	Vanilla [6], Mamba in Time	30.3	22.1	26.2	30.9	27.5	29.2
7	TimeMamba (ours)	Frozen [3], Mamba in Time	30.7	22.8	26.8	31.3	27.8	29.5
8	TimeMamba (ours)	Frozen [3], Mamba in Space-Time	30.1	21.9	26.0	30.7	27.1	28.9

Table 8: Results of zero-shot multi-instance retrieval on EK100 [13]. We compare different models with divided space-time interaction. The additional configuration “Mamba in space-time” is used for further comparison. We uniformly sample 4 frames for training and inference.

Model	Multi-instance Retrieval						Action Recognition			
	mAP			nDCG			Verb	Noun	Action	
	V2T	T2V	Avg	V2T	T2V	Avg	Top1	Top1	Top1	Top5
EgoVLP [49]	49.9	40.5	45.0	60.9	57.9	59.4	-	-	-	-
TimeSformer [6]	49.1	39.3	44.2	60.0	57.6	58.8	63.8	52.4	41.3	60.4
TimeMamba (ours)	50.3	40.3	45.3	62.4	59.2	60.9	66.6	53.3	42.8	63.2

Table 9: Results of fine-tuned multi-instance retrieval and action recognition on EK100 [13]. We uniformly sample 16 frames for training and inference.

block ($4C^2$), where C is the feature dimension. Therefore, we set the expanding ratio E of ViM block to 1, reducing its parameter amount to $3.25C^2$ for a fair comparison. In addition to the vanilla residual connection form used by TimeSformer [6], we additionally explored the Frozen-style [3] adaption fashion. We list blocks with different divided space-time interactions in Figure 4. We train the model with 4-frame input using AVION [93] codebase, with the remaining settings unchanged as [94] and [93]. The model is initialized with CLIP-B16 [61] pre-trained via image-text contrastive learning.

Zero-shot multi-instance retrieval. We first evaluate different models with divided space-time interaction operations in Table 8. Our reproduced Frozen style residual connection achieves consistent results to LaviLa [94]. When comparing vanilla and Frozen [3] styles, we observe that the Frozen style consistently yields better results (ID4 vs ID5, ID6 vs ID7). Furthermore, under the same adaptation method, the ViM-based temporal module consistently outperforms the attention-based temporal module (ID4 vs ID6, ID5 vs ID7). Notably, the ViM temporal block we used has fewer parameters compared to the temporal self-attention block, highlighting the exceptional parameter utilization and information extraction capabilities of Mamba’s selective scanning [30]. Additionally, We go beyond the temporal modeling ability of ViM and validate the space-time ViM block. Space-time ViM block replaces the temporal ViM block with joint spatiotemporal modeling over the entire video sequence. Surprisingly, we observed that the space-time ViM block, despite introducing global modeling at the spatiotemporal level, actually leads to a decline in performance (ID7 vs

Model	#Frame	Accuracy
FrozenBiLM [88]	10	26.4
FrozenBiLM [88]	90	26.9
InternVideo [85]	15	31.6
InternVideo [85]	90	32.0
TimeSformer [6]	16	33.6
TimeSformer [6]	128	32.9
TimeSformer [6]	8192	38.1
TimeMamba (ours)	16	31.9
TimeMamba (ours)	128	33.4
TimeMamba (ours)	8192	38.7

Table 10: Results of zero-shot long-form video QA on EgoSchema [56]. The model is trained on Ego4D [29] with 4 frames and tested on different frames.

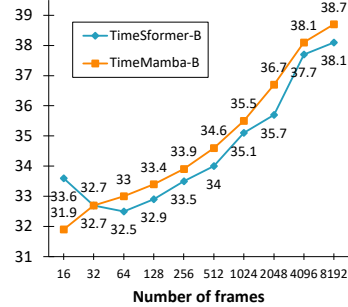


Fig. 5: The results of using different numbers of testing frames for zero-shot QA on EgoSchema [56]. The model is trained on Ego4D [29] with 4 frames.

ID8). We suppose the scanning-based spatial-temporal may damage the spatial feature distribution produced by the pre-trained space attention blocks.

Fine-tuned multi-instance retrieval and action recognition. We continue to finetune the pre-trained models with 16 frames on Epic-Kitchen-100 [13] dataset for multi-instance retrieval and action recognition. In Table 9, we observe that TimeMamba outperforms TimeSformer by a significant margin. In particular, TimeMamba surpasses TimeSformer in the context of verb recognition by 2.8 points, demonstrating its effectiveness in temporal modeling.

Zero-shot long-form video QA. We conduct further evaluation of the model’s long-form video Question Answering performance on EgoSchema [56]. As shown in Table 10, both TimeSformer and TimeMamba, when pre-trained on Ego4D [29], outperform the performance of large-scale pre-trained models [85, 88]. Additionally, we increase the number of testing frames to explore the effect of ViM block’s long-form temporal modeling ability. As shown in Figure 5, despite both models being pre-trained on 4 frames, the performance of both TimeMamba and TimeSformer steadily improves with increased frames. Meanwhile, significant improvement can be observed when using 8192 frames. When the input frame exceeds 32, TimeMamba generally benefits from more frames compared to TimeSformer, indicating the superiority of the temporal ViM block against temporal self-attention.

4.4 Mamba for Spatial-Temporal Modeling

Tasks and datasets. Finally, we assess the spatial-temporal modeling capability of Mamba. Similar to the previous subsection, we evaluate the model’s performance in zero-shot multi-instance retrieval on Epic-Kitchens-100 dataset [13].

Baseline and competitor. ViViT [2] and TimeSformer [6] investigated the transformation of ViT with spatial attention into models with spatial-temporal joint attention. In accordance with these works, we further extend the spatial

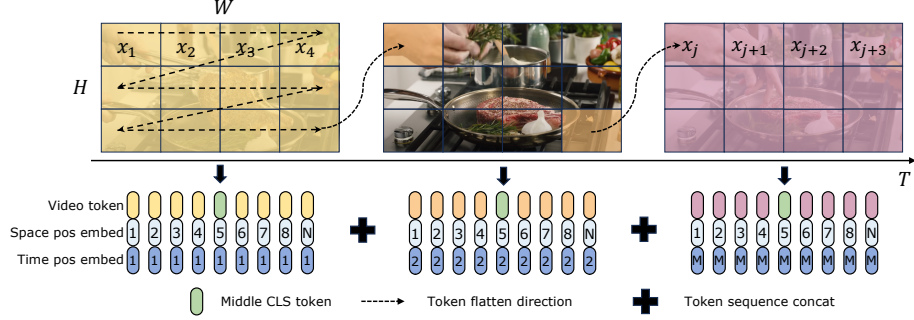


Fig. 6: Illustration of the modeling process of our ViViM. We highlight the direction of flattening video tokens through dashed arrows, which determine the process of space-time selective scanning.

Model	Param	Pretrain	#F	mAP			nDCG		
				V2T	T2V	Avg	V2T	T2V	Avg
ViT-B	86M	CLIP [61], WIT	4	30.95	23.15	27.05	30.95	27.65	29.30
ViT-T	6M	DeiT [69], IN1K [16]	4	15.50	11.10	13.30	22.48	19.66	21.07
ViT-B	86M	DeiT [69], IN1K [16]	4	25.08	18.49	21.79	27.80	24.87	26.34
ViT-T	6M	DeiT [69], IN1K [16]	16	20.47	15.29	17.88	25.74	22.89	24.31
ViT-S	22M	DeiT [69], IN1K [16]	16	23.80	17.60	20.70	27.40	24.40	25.90
ViViM-T (ours)	7M	DeiT [69], IN1K [16]	16	23.31	17.21	20.26	27.40	24.30	25.80
ViViM-S (ours)	26M	DeiT [69], IN1K [16]	16	26.00	19.60	22.80	28.20	25.30	26.70

Table 11: Results of zero-shot multi-instance retrieval on EK100 [13]. We compare ViT with space-time joint attention and our ViViM with space-time joint selective scanning. #F denotes the frame number for training and inference.

selective scanning of ViM model [96] to incorporate space-time selective scanning. We refer to this extended model as ViViM. We utilize the ViM model that has been pre-trained on ImageNet-1K [16] for initialization. The ViM model incorporates a **cls** token, which is inserted in the middle of the flattened token sequence. To convert the ViM model into ViViM, we adopt a straightforward approach illustrated in Figure 6. For a given input consisting of M frames, we insert the **cls** token in the middle of the token sequence corresponding to each frame. Additionally, we add temporal positional embeddings that are initialized to zero for each frame. The flattened video sequence is then input into the ViViM model. The output of the model is taken by computing the average of **cls** token from each frame.

Results and analysis. We further analyze the results of our ViViM on zero-shot multi-instance retrieval. Table 11 presents the performance of various spatiotemporal models on zero-shot multi-instance retrieval. When comparing ViT and ViViM, both of which are pre-trained on ImageNet-1K [16], we observe that our ViViM outperforms ViT. Interestingly, although the performance gap between ViT-S [69] and ViM-S [96] on ImageNet-1K is slight (79.8 *vs.* 80.5),

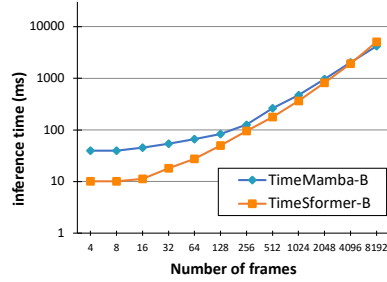


Fig. 7: Inference speed of TimeSformer-B and TimeMamba-B with different numbers of frames.

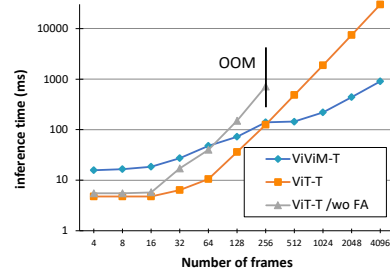


Fig. 8: Inference speed of ViT-T and ViViM-T with different numbers of frames.


ViViM-S demonstrates a significant improvement (+2.1 mAP@Avg) over ViT-S on zero-shot multi-instance retrieval. This finding suggests that our ViViM is highly effective in modeling long sequences, leading to improved performance.

5 Efficiency Analysis

We compare the inference speed of different spatiotemporal models. This test fixes 196 tokens in spatial dimensions, and continuously increases the number of frames. All tests are performed on a single A100 GPU at half precision. For a fair comparison, all attention blocks are equipped with Flash-attention [14, 15].

We test the inference speed from 4 frames to 8192 frames and list the test results in Figure 7 and Figure 8. Both tables show that Mamba can offer speed advantages over the transformer series models particularly when the number of frames is substantial. In Figure 8, for fair and comprehensive comparison, we compare ViViM-T with ViT with and without the use of Flash-attention [14, 15]. The comparison of ViViM-T with ViT+Flash-attention is fair because both methods are optimized considering hardware I/O speed. Our ViViM-T becomes more efficient than ViT-T with flash-attention when the input frame number is greater than 256. Without Flash-Attention, ViViM-T is relatively more efficient, surpassing ViT when the frame number is greater than 64. For TimeMamba-B in Figure 7, when the input is over 8192 frames, the efficiency begins to exceed that of timesformer-B. Since the form of token interaction only differs in time interactions, the efficiency difference is not as significant as the comparison between ViViM and ViT.

6 Conclusion

Our comprehensive evaluation of Mamba within the video understanding domain showcases its potential as a viable alternative to traditional transformers. Through the Video Mamba Suite , comprising 14 models/modules across

12 video understanding tasks, we demonstrate Mamba’s capability to efficiently handle complex spatial-temporal dynamics, exhibiting both superior performance and promising efficiency-performance trade-offs. These findings not only underline Mamba’s suitability for video analysis tasks but also open new avenues for its application in computer vision. Future work could further explore Mamba’s adaptability and extend its utility to more complex, multi-modal video understanding challenges.

This supplementary material shows additional details about the models and more experimental results.

A7 More Experimental Results

In this work, we present a comprehensive study on State Space Models exemplified by Mamba as a versatile alternative in video understanding. We extensively experiment with 4 roles of Mamba in 12 tasks on 13 datasets. Due to the page limitation, in this section, we show experimental results not listed in the main submission.

A7.1 Temporal Action Localization

Table A12, Table A13, and Table A14 present the results of ActionMamba equipped with the ViM [96] block and our DBM alternative on THUMOS-14 [39], ActivityNet [36] and FineAction [52] datasets. The superior performance compared to ActionFormer [90] on all datasets highlights the advantages of bidirectional Mamba in temporal action localization. Furthermore, our DBM block demonstrates superior performance compared to the ViM block. From the results of temporal action localization on all the datasets, we conclude Mamba exhibits stronger capabilities than the Transformer counterpart in the temporal action localization task.

Method	Block	mAP@0.3	mAP@0.4	mAP@0.5	mAP@0.6	mAP@0.7	mAP@Avg
ActionFormer [90]	Window Attn [12]	82.26	81.85	75.05	65.82	50.27	71.86
ActionMamba	ViM [96]	86.22	82.87	76.42	66.43	50.25	72.44
ActionMamba	DBM (ours)	86.89	83.09	76.90	65.91	50.82	72.72

Table A12: Results of temporal action localization on THUMOS-14 [39]. The metric is mean Average Precision (mAP) under multiple tIoU thresholds {0.3, 0.4, 0.5, 0.6, 0.7}.

Method	Block	mAP@0.5	mAP@0.75	mAP@0.95	mAP@Avg
ActionFormer [90]	Window Attn [12]	61.47	44.61	12.73	41.19
ActionMamba	ViM [96]	62.31	43.17	9.65	41.77
ActionMamba	DBM (ours)	62.43	43.49	10.23	42.02

Table A13: Results of temporal action localization on ActivityNet [36]. The metric is mean Average Precision (mAP) under multiple tIoU thresholds {0.5, 0.75, 0.95}.

A7.2 Temporal Action Segmentation

Table A15 and Table A16 show the performances of the temporal action segmentation task on the Breakfast and 50Salads datasets. On the Breakfast dataset (Table A15), Mamba-based methods achieve significantly stronger performances. On the 50Salads dataset (Table A16), Transformer-based method with encoder-decoder structure, ASFormer [89], gets better results in terms of Accuracy, Edit

Method	Block	mAP@0.5	mAP@0.75	mAP@0.95	mAP@Avg
ActionFormer [90]	Window Attn [12]	43.11	27.09	5.32	27.22
ActionMamba	ViM [96]	44.15	28.30	6.14	28.36
ActionMamba	DBM (ours)	45.44	28.82	6.79	29.04

Table A14: Results of temporal action localization on FineAction [52]. The metric is mean Average Precision (mAP) under multiple tIoU thresholds {0.5, 0.75, 0.95}.

Method	Block	Acc	Edit	F1@10	F1@25	F1@50
MS-TCN [21]	Dilated Conv, Enc	66.3	61.7	52.6	48.1	37.9
ASFormer [89]	Window Attn [12], Enc-Dec	73.5	75.0	76.0	70.6	57.4
ASMamba	ViM [96], Enc	75.8	76.9	78.7	73.8	61.6
ASMamba	DBM (ours), Enc	73.5	75.8	77.4	72.0	59.3

Table A15: Results of temporal action segmentation on Breakfast [43] dataset. The metrics are accuracy, edit distance [7], and instance-wise F1 under multiple tIoU thresholds {0.1, 0.25, 0.5}. † denotes our reproduced results with its official code.

score, and F1@10, while Mamba-based methods, ASMamba, perform better on F1@25. We also observe that ASMamba outperforms encoder-only ASFormer, highlighting the strong temporal encoding capability of the Mamba-based model. These results further reveal that the encoder-decoder structure may be a better choice compared with the encode-only structure in some scenarios.

A7.3 Dense Video Captioning

We also present the localization metrics (Recall and Precision) of dense video captioning in Table A17 and Table A18. The results demonstrate that the Mamba-based temporal encoder produces improved position representations and enhances the localization capability, which aligns with its superior performance on the temporal action localization task.

A7.4 Action Recognition

Furthermore, we present the results of action recognition on Kinetics-400 [42], as demonstrated in Table A19. ViViM-T and ViViM-S achieve 77.4% and 80.1% Top-1 accuracy, respectively. Compared with Transformer-based models with tiny size (*e.g.* 28M VideoSwin-T [55]), ViViM-T achieves a competitive performance with significantly fewer parameters (7M). One can still observe a performance gap between ViViM and models with dedicated spatial/temporal/spatio-temporal module designs (*e.g.* UniFormer [45]), and we leave the design of SSM-based modules for video understanding as future work.

A8 ViM and DBM

In this section, we provide a detailed description of our explored DBM block and its differences from the ViM block.

Method	Block	Acc	Edit	F1@10	F1@25	F1@50
MS-TCN [21]	Dilated Conv, Enc	80.7	67.9	76.3	74.0	64.5
ASFormer [89]	Window Attn [12], Enc-Enc	77.0	64.4	73.0	70.5	60.3
ASFormer [89]	Window Attn [12], Enc-Dec	85.6	79.6	85.1	73.4	76.0
ASMamba	ViM [96], Enc	85.4	77.7	84.5	83.4	74.7
ASMamba	DBM (ours), Enc	83.7	75.2	82.5	80.5	72.5

Table A16: Results of temporal action segmentation on 50salads [66] dataset. The metrics are accuracy, edit distance [7], and instance-wise F1 under multiple tIoU thresholds {0.1, 0.25, 0.5}. † denotes our reproduced results with its official code.

Method	Block	B-4	METEOR	ROUGE-L	CIDER	SODA	Recall	Precision
PDVC [83]	DeformAttn [97]	1.75	6.73	14.65	26.07	5.47	51.7	56.1
PDVC [83]	ViM [96]	1.68	6.92	14.72	26.26	5.33	53.1	56.3
PDVC	DBM (ours)	1.76	7.16	14.83	26.77	5.27	52.4	56.3

Table A17: Results of dense video captioning on ActivityNet [36]. The metrics include BLEU-4 [58], METEOR [4], CIDEr [77] SODA_c [27], Recall and Precision.

ViM block. Given an input sequence $\mathbf{x} \in \mathbb{R}^{N \times d}$, where N represents the sequence length and d represents the feature dimension, the ViM block initially expands the dimension to $d \cdot E$, resulting in two hidden state sequences, namely \mathbf{x}_s and \mathbf{x}_g . The first sequence, \mathbf{x}_s , is intended for scanning, while the second sequence, \mathbf{x}_g , is used for gating. The selective scanning layer scans \mathbf{x}_s in both directions using two distinct parameters. Finally, \mathbf{x}_g is employed to gate the bidirectionally scanned features, and the output feature is obtained through the average of the two gated features.

DBM block. In the DBM block, given an input sequence $\mathbf{x} \in \mathbb{R}^{N \times d}$, where N represents the sequence length and d represents the feature dimension, the process is similar to the ViM block. Initially, the DBM block expands the dimension to $d \cdot E$, resulting in two hidden state sequences, namely \mathbf{x}_s and \mathbf{x}_g . However, in the DBM block, the forward and backward features are separated along the channel dimension, resulting in four hidden state sequences: $\mathbf{x}_s^{forward}, \mathbf{x}_g^{forward}, \mathbf{x}_s^{backward}, \mathbf{x}_g^{backward} \in \mathbb{R}^{N \times d \cdot \frac{E}{2}}$. The selective scanning layer scans $\mathbf{x}_s^{forward}$ and $\mathbf{x}_g^{forward}$ using shared parameters. Finally, $\mathbf{x}_g^{forward}$ and $\mathbf{x}_g^{backward}$ are used to gate the both bidirectionally scanned features, respectively. The output feature is obtained by concatenating and projecting the two gated features.

Analysis. Considering $E = 2$, the ViM block and DBM block exhibit differences in terms of the number of parameters utilized and the scanning context for static and dynamic modeling. With Mamba [30] serving as the baseline, Table A20 presents the disparity in parameters, context, and time cost between ViM [96] and DBM. Compared to the ViM block, the DBM block provides static direction separation and reduces the capacity for dynamic modeling, making it more compatible with certain downstream datasets and tasks, such as temporal action localization. Additionally, halving the scanning context results in a cor-

Method	Block	B-4	METEOR	ROUGE-L	CIDER	SODA	Recall	Precision
PDVC [83]	DeformAttn [97]	0.73	4.25	9.31	20.48	4.02	23.0	31.1
PDVC [83]	ViM [96]	0.71	4.32	9.57	20.59	4.09	24.1	33.0
PDVC	DBM (ours)	0.86	4.44	9.64	22.11	4.32	25.2	32.4

Table A18: Results of dense video captioning on YouCook2 [95]. The metrics include BLEU-4 [58], METEOR [4], CIDEr [77], SODA_c [27], Recall and Precision.

Method	Arch	Pretrain	Input Size	Param	GFLOPs	Top-1	Top-5
SlowFast _{R101} +NL [25]	CNN	-	80×224^2	60	$234 \times 3 \times 10$	79.8	93.9
X3D-XL [23]	CNN	-	16×312^2	20	$194 \times 3 \times 10$	80.4	94.6
VideoSwin-T [55]	Trans	IN1K [16]	32×224^2	28	$88 \times 3 \times 4$	78.8	93.6
VideoSwin-B [55]	Trans	IN1K [16]	32×224^2	88	$88 \times 3 \times 4$	80.6	94.5
MViTv1-B [20]	Trans+CNN	-	32×224^2	37	$70 \times 1 \times 5$	80.2	94.4
UniFormer-S [45]	Trans+CNN	IN1K [16]	16×224^2	21	$42 \times 1 \times 4$	80.8	94.7
STAM [64]	Trans	IN22K [16]	64×224^2	121	$1040 \times 1 \times 1$	79.2	-
TimeSformer-L [6]	Trans	IN22K [16]	96×224^2	311	$2380 \times 3 \times 1$	80.7	94.7
ViViT-L [2]	Trans	IN22K [16]	16×224^2	311	$3992 \times 3 \times 4$	81.3	94.7
ViViM-T (ours)	SSM	IN1K [16]	16×224^2	7	$17 \times 3 \times 4$	77.4	92.8
ViViM-S (ours)	SSM	IN1K [16]	16×224^2	26	$68 \times 3 \times 4$	80.1	94.1

Table A19: Comparison with the previous methods on Kinetics-400 [42].

Block	Param (sta)	Param (dyn)	Memory (sta)	Memory (dyn)	Time cost
Mamba [30]	100%	100%	100%	100%	100%
ViM [96]	100%	200%	100%	200%	200%
DBM (ours)	100%	100%	100%	100%	100%

Table A20: Parameters, context and cost time comparison between Mamba [30], ViM [96] and DBM. “sta” denotes the static projection and “dyn” is dynamic scanning.

responding halving of the (training and inference) time cost, consistent with the vanilla Mamba [30] block.

A9 Hyperparameter Sensitivity

Additionally, we conducted an analysis of the hyperparameter sensitivity of the Mamba series models. In most experiments, the training hyperparameters were found to be insensitive. We simply replaced the transformer block with the Mamba-based block for most tasks. However, for video temporal grounding, we observed that a larger learning rate yielded better optimization results. Furthermore, increasing the loss weight for video-text alignment loss facilitated the model’s convergence. We postulate that these adjustments are related to the distinction between the scanning mechanism and global interaction, especially for mutli-modal aggregation.

References

1. Ali, A., Touvron, H., Caron, M., Bojanowski, P., Douze, M., Joulin, A., Laptev, I., Neverova, N., Synnaeve, G., Verbeek, J., Jégou, H.: Xcit: Cross-covariance image transformers. pp. 20014–20027 (2021) [3](#)
2. Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lucic, M., Schmid, C.: Vivit: A video vision transformer. In: ICCV. pp. 6816–6826 (2021) [1](#), [2](#), [3](#), [12](#), [19](#)
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. pp. 1708–1718 (2021) [10](#), [11](#)
4. Banerjee, S., Lavie, A.: Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In: Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization. pp. 65–72 (2005) [7](#), [18](#), [19](#)
5. Bao, H., Dong, L., Piao, S., Wei, F.: Beit: BERT pre-training of image transformers. In: ICLR (2022) [1](#)
6. Bertasius, G., Wang, H., Torresani, L.: Is space-time attention all you need for video understanding? In: ICML. vol. 139, pp. 813–824 (2021) [2](#), [3](#), [10](#), [11](#), [12](#), [19](#)
7. Brill, E., Moore, R.C.: An improved error model for noisy channel spelling correction. In: 38th Annual Meeting of the Association for Computational Linguistics, Hong Kong, China, October 1-8, 2000. pp. 286–293. ACL (2000) [6](#), [17](#), [18](#)
8. Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners (2020) [3](#)
9. Carreira, J., Zisserman, A.: Quo vadis, action recognition? A new model and the kinetics dataset. In: CVPR. pp. 4724–4733 (2017) [3](#)
10. Chen, G., Zheng, Y.D., Wang, J., Xu, J., Huang, Y., Pan, J., Wang, Y., Wang, Y., Qiao, Y., Lu, T., et al.: Videollm: Modeling video sequence with large language models. arXiv preprint arXiv:2305.13292 (2023) [2](#)
11. Chen, Z., Wu, J., Wang, W., Su, W., Chen, G., Xing, S., Muyan, Z., Zhang, Q., Zhu, X., Lu, L., et al.: Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. arXiv preprint arXiv:2312.14238 (2023) [3](#)
12. Choromanski, K., Likhoshesterov, V., Dohan, D., Song, X., Gane, A., Sarlos, T., Hawkins, P., Davis, J., Mohiuddin, A., Kaiser, L., et al.: Rethinking attention with performers. arXiv preprint arXiv:2009.14794 (2020) [6](#), [16](#), [17](#), [18](#)
13. Damen, D., Doughty, H., Farinella, G.M., Furnari, A., Kazakos, E., Ma, J., Moltisanti, D., Munro, J., Perrett, T., Price, W., Wray, M.: Rescaling egocentric vision: Collection, pipeline and challenges for EPIC-KITCHENS-100. IJCV **130**(1), 33–55 (2022) [7](#), [8](#), [10](#), [11](#), [12](#), [13](#)
14. Dao, T.: FlashAttention-2: Faster attention with better parallelism and work partitioning. In: ICLR (2023) [14](#)
15. Dao, T., Fu, D.Y., Ermon, S., Rudra, A., Ré, C.: FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In: NeurIPS (2022) [14](#)
16. Deng, J., Dong, W., Socher, R., Li, L., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: CVPR. pp. 248–255 (2009) [13](#), [19](#)
17. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. In: NAACL. pp. 4171–4186 (2019) [3](#)

18. 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) [1](#), [3](#)
19. Duan, Y., Wang, W., Chen, Z., Zhu, X., Lu, L., Lu, T., Qiao, Y., Li, H., Dai, J., Wang, W.: Vision-rwkv: Efficient and scalable visual perception with rwkv-like architectures. arXiv preprint arXiv:TODO (2024) [3](#)
20. Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., Feichtenhofer, C.: Multiscale vision transformers. In: ICCV. pp. 6804–6815 (2021) [2](#), [3](#), [19](#)
21. Farha, Y.A., Gall, J.: Ms-tcn: Multi-stage temporal convolutional network for action segmentation. In: CVPR. pp. 3575–3584 (2019) [6](#), [17](#), [18](#)
22. Fathi, A., Ren, X., Rehg, J.M.: Learning to recognize objects in egocentric activities. In: CVPR. pp. 3281–3288 (2011) [6](#), [7](#)
23. Feichtenhofer, C.: X3d: Expanding architectures for efficient video recognition. In: CVPR. pp. 203–213 (2020) [19](#)
24. Feichtenhofer, C., Fan, H., Li, Y., He, K.: Masked autoencoders as spatiotemporal learners (2022) [3](#)
25. Feichtenhofer, C., Fan, H., Malik, J., He, K.: Slowfast networks for video recognition. In: ICCV. pp. 6201–6210 (2019) [1](#), [3](#), [19](#)
26. 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. arXiv preprint arXiv:2212.14052 (2022) [4](#)
27. Fujita, S., Hirao, T., Kamigaito, H., Okumura, M., Nagata, M.: Soda: Story oriented dense video captioning evaluation framework. In: ECCV. pp. 517–531 (2020) [7](#), [18](#), [19](#)
28. Gao, J., Sun, C., Yang, Z., Nevatia, R.: Tall: Temporal activity localization via language query. In: ICCV. pp. 5267–5275 (2017) [8](#), [9](#), [10](#)
29. Grauman, K., Westbury, A., Byrne, E., Chavis, Z., Furnari, A., Girdhar, R., Hamburger, J., Jiang, H., Liu, M., Liu, X., et al.: Ego4d: Around the world in 3,000 hours of egocentric video. In: CVPR. pp. 18995–19012 (2022) [10](#), [12](#)
30. Gu, A., Dao, T.: Mamba: Linear-time sequence modeling with selective state spaces. arXiv preprint arXiv:2312.00752 (2023) [2](#), [3](#), [4](#), [5](#), [7](#), [8](#), [9](#), [11](#), [18](#), [19](#)
31. Gu, A., Goel, K., Gupta, A., Ré, C.: On the parameterization and initialization of diagonal state space models. NeurIPS **35**, 35971–35983 (2022) [3](#)
32. Gu, A., Goel, K., Ré, C.: Efficiently modeling long sequences with structured state spaces. arXiv preprint arXiv:2111.00396 (2021) [2](#), [3](#), [4](#)
33. Gu, A., Johnson, I., Goel, K., Saab, K., Dao, T., Rudra, A., Ré, C.: Combining recurrent, convolutional, and continuous-time models with linear state space layers. NeurIPS **34**, 572–585 (2021) [3](#)
34. He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R.B.: Masked autoencoders are scalable vision learners. In: CVPR. pp. 15979–15988 (2022) [1](#), [3](#)
35. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016) [3](#)
36. Heilbron, F.C., Escorcia, V., Ghanem, B., Niebles, J.C.: Activitynet: A large-scale video benchmark for human activity understanding. In: CVPR. pp. 961–970 (2015) [7](#), [16](#), [18](#)
37. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997) [10](#)
38. Hua, W., Dai, Z., Liu, H., Le, Q.V.: Transformer quality in linear time. vol. 162, pp. 9099–9117 (2022) [5](#)

39. Idrees, H., Zamir, A.R., Jiang, Y., Gorban, A., Laptev, I., Sukthankar, R., Shah, M.: The THUMOS challenge on action recognition for videos "in the wild" **155**, 1–23 (2017) [16](#)
40. Ioffe, S., Szegedy, C.: Batch normalization: Accelerating deep network training by reducing internal covariate shift. vol. 37, pp. 448–456 (2015) [3](#)
41. Islam, M.M., Bertasius, G.: Long movie clip classification with state-space video models. In: ECCV. pp. 87–104 (2022) [4](#)
42. Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., Viola, F., Green, T., Back, T., Natsev, P., Suleyman, M., Zisserman, A.: The kinetics human action video dataset. CoRR **abs/1705.06950** (2017) [17](#), [19](#)
43. Kuehne, H., Arslan, A., Serre, T.: The language of actions: Recovering the syntax and semantics of goal-directed human activities. In: CVPR. pp. 780–787 (2014) [17](#)
44. Lei, J., Berg, T.L., Bansal, M.: Detecting moments and highlights in videos via natural language queries. pp. 11846–11858 (2021) [8](#), [9](#)
45. Li, K., Wang, Y., Gao, P., Song, G., Liu, Y., Li, H., Qiao, Y.: Uniformer: Unified transformer for efficient spatiotemporal representation learning. IEEE TPAMI (2023) [3](#), [17](#), [19](#)
46. Li, K., Wang, Y., Li, Y., Wang, Y., He, Y., Wang, L., Qiao, Y.: Unmasked teacher: Towards training-efficient video foundation models. In: ICCV. pp. 19948–19960 (2023) [3](#)
47. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text summarization branches out. pp. 74–81 (2004) [7](#)
48. Lin, J., Gan, C., Han, S.: TSM: temporal shift module for efficient video understanding. In: ICCV. pp. 7082–7092 (2019) [3](#)
49. Lin, K.Q., Wang, J., Soldan, M., Wray, M., Yan, R., Xu, E.Z., Gao, D., Tu, R., Zhao, W., Kong, W., Cai, C., Wang, H., Damen, D., Ghanem, B., Liu, W., Shou, M.Z.: Egocentric video-language pretraining (2022) [10](#), [11](#)
50. Lin, K.Q., Zhang, P., Chen, J., Pramanick, S., Gao, D., Wang, A.J., Yan, R., Shou, M.Z.: Univtg: Towards unified video-language temporal grounding. In: ICCV. pp. 2794–2804 (2023) [8](#), [9](#)
51. Liu, Y., Li, S., Wu, Y., Chen, C.W., Shan, Y., Qie, X.: Umt: Unified multi-modal transformers for joint video moment retrieval and highlight detection. In: CVPR. pp. 3042–3051 (2022) [9](#)
52. Liu, Y., Wang, L., Wang, Y., Ma, X., Qiao, Y.: Fineaction: A fine-grained video dataset for temporal action localization. IEEE TIP **31**, 6937–6950 (2022) [16](#), [17](#)
53. Liu, Y., Tian, Y., Zhao, Y., Yu, H., Xie, L., Wang, Y., Ye, Q., Liu, Y.: Vmamba: Visual state space model. arXiv preprint arXiv:2401.10166 (2024) [2](#), [3](#)
54. 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. pp. 9992–10002 (2021) [3](#)
55. Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H.: Video swin transformer. In: CVPR. pp. 3192–3201 (2022) [2](#), [3](#), [17](#), [19](#)
56. Mangalam, K., Akshulakov, R., Malik, J.: Egoschema: A diagnostic benchmark for very long-form video language understanding. NeurIPS **36** (2024) [10](#), [12](#)
57. Mehta, H., Gupta, A., Cutkosky, A., Neyshabur, B.: Long range language modeling via gated state spaces. arXiv preprint arXiv:2206.13947 (2022) [4](#)
58. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: ACL. pp. 311–318 (2002) [7](#), [18](#), [19](#)
59. Peng, B., Alcaide, E., Anthony, Q., Albalak, A., Arcadinho, S., Cao, H., Cheng, X., Chung, M., Grella, M., GV, K.K., et al.: Rwkv: Reinventing rnnns for the transformer era. arXiv preprint arXiv:2305.13048 (2023) [3](#)

60. Qiu, Z., Yao, T., Mei, T.: Learning spatio-temporal representation with pseudo-3d residual networks. In: ICCV. pp. 5534–5542 (2017) [3](#)
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. vol. 139, pp. 8748–8763 (2021) [1](#), [11](#), [13](#)
62. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019) [1](#), [3](#)
63. Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., Liu, P.J.: Exploring the limits of transfer learning with a unified text-to-text transformer. J. Mach. Learn. Res. **21**, 140:1–140:67 (2020) [1](#), [3](#)
64. Sharir, G., Noy, A., Zelnik-Manor, L.: An image is worth 16x16 words, what is a video worth? arXiv preprint arXiv:2103.13915 (2021) [19](#)
65. Smith, J.T., Warrington, A., Linderman, S.W.: Simplified state space layers for sequence modeling. arXiv preprint arXiv:2208.04933 (2022) [4](#)
66. Stein, S., McKenna, S.J.: Combining embedded accelerometers with computer vision for recognizing food preparation activities. pp. 729–738 (2013) [18](#)
67. 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 preprint arXiv:2307.08621 (2023) [3](#)
68. Tong, Z., Song, Y., Wang, J., Wang, L.: Videomae: Masked autoencoders are data-efficient learners for self-supervised video pre-training (2022) [1](#), [2](#), [3](#)
69. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. vol. 139, pp. 10347–10357 (2021) [1](#), [3](#), [13](#)
70. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023) [1](#), [3](#)
71. Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al.: Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971 (2023) [3](#)
72. Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al.: Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288 (2023) [3](#)
73. Tran, D., Ray, J., Shou, Z., Chang, S., Paluri, M.: Convnet architecture search for spatiotemporal feature learning. CoRR **abs/1708.05038** (2017) [1](#), [3](#)
74. Tran, D., Wang, H., Feiszli, M., Torresani, L.: Video classification with channel-separated convolutional networks. In: ICCV. pp. 5551–5560 (2019) [3](#)
75. Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M.: A closer look at spatiotemporal convolutions for action recognition. In: CVPR. pp. 6450–6459 (2018) [3](#)
76. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. In: NeurIPS. pp. 5998–6008 (2017) [1](#), [8](#)
77. Vedantam, R., Lawrence Zitnick, C., Parikh, D.: Cider: Consensus-based image description evaluation. In: CVPR. pp. 4566–4575 (2015) [7](#), [18](#), [19](#)
78. Wang, J., Chen, G., Huang, Y., Wang, L., Lu, T.: Memory-and-anticipation transformer for online action understanding. In: ICCV. pp. 13824–13835 (2023) [2](#), [8](#)
79. Wang, J., Zhu, W., Wang, P., Yu, X., Liu, L., Omar, M., Hamid, R.: Selective structured state-spaces for long-form video understanding. In: CVPR. pp. 6387–6397 (2023) [4](#)

80. Wang, L., Huang, B., Zhao, Z., Tong, Z., He, Y., Wang, Y., Wang, Y., Qiao, Y.: VideoMAE V2: scaling video masked autoencoders with dual masking. In: CVPR. pp. 14549–14560 (2023) [1](#), [3](#)
81. Wang, L., Tong, Z., Ji, B., Wu, G.: TDN: temporal difference networks for efficient action recognition. In: CVPR. pp. 1895–1904 (2021) [3](#)
82. Wang, L., Xiong, Y., Wang, Z., Qiao, Y., Lin, D., Tang, X., Gool, L.V.: Temporal segment networks for action recognition in videos. IEEE TPAMI **41**(11), 2740–2755 (2019) [3](#)
83. Wang, T., Zhang, R., Lu, Z., Zheng, F., Cheng, R., Luo, P.: End-to-end dense video captioning with parallel decoding. In: ICCV. pp. 6847–6857 (2021) [7](#), [8](#), [18](#), [19](#)
84. Wang, W., Xie, E., Li, X., Fan, D.P., Song, K., Liang, D., Lu, T., Luo, P., Shao, L.: Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In: ICCV. pp. 568–578 (2021) [3](#)
85. Wang, Y., Li, K., Li, Y., He, Y., Huang, B., Zhao, Z., Zhang, H., Xu, J., Liu, Y., Wang, Z., et al.: Internvideo: General video foundation models via generative and discriminative learning. arXiv preprint arXiv:2212.03191 (2022) [3](#), [12](#)
86. Xie, S., Girshick, R.B., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: CVPR. pp. 5987–5995 (2017) [3](#)
87. Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K.: Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In: ECCV. vol. 11219, pp. 318–335 (2018) [3](#)
88. Yang, A., Miech, A., Sivic, J., Laptev, I., Schmid, C.: Zero-shot video question answering via frozen bidirectional language models. NeurIPS **35**, 124–141 (2022) [12](#)
89. Yi, F., Wen, H., Jiang, T.: Asformer: Transformer for action segmentation. In: BMVC. p. 236 (2021) [2](#), [6](#), [7](#), [16](#), [17](#), [18](#)
90. Zhang, C., Wu, J., Li, Y.: Actionformer: Localizing moments of actions with transformers. In: Avidan, S., Brostow, G.J., Cissé, M., Farinella, G.M., Hassner, T. (eds.) ECCV. vol. 13664, pp. 492–510 (2022) [2](#), [6](#), [7](#), [16](#), [17](#)
91. Zhao, H., Torralba, A., Torresani, L., Yan, Z.: HACS: human action clips and segments dataset for recognition and temporal localization. In: ICCV. pp. 8667–8677 (2019) [6](#)
92. Zhao, Y., Krähenbühl, P.: Real-time online video detection with temporal smoothing transformers. In: ECCV. pp. 485–502 (2022) [7](#), [8](#)
93. Zhao, Y., Krähenbühl, P.: Training a large video model on a single machine in a day. arXiv preprint arXiv:2309.16669 (2023) [11](#)
94. Zhao, Y., Misra, I., Krähenbühl, P., Girdhar, R.: Learning video representations from large language models. In: CVPR. pp. 6586–6597 (2023) [11](#)
95. Zhou, L., Xu, C., Corso, J.: Towards automatic learning of procedures from web instructional videos. In: AAAI. vol. 32 (2018) [7](#), [19](#)
96. Zhu, L., Liao, B., Zhang, Q., Wang, X., Liu, W., Wang, X.: Vision mamba: Efficient visual representation learning with bidirectional state space model. arXiv preprint arXiv:2401.09417 (2024) [2](#), [3](#), [5](#), [6](#), [7](#), [10](#), [13](#), [16](#), [17](#), [18](#), [19](#)
97. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable DETR: deformable transformers for end-to-end object detection. In: ICLR (2021) [7](#), [18](#), [19](#)