VideoSEMA: a scalable and efficient Mamba-like attention for video understanding

Nhat Thanh Tran*
Department of Mathematics
University of California, Irvine
Irvine, USA
nhattt@uci.edu

Fanghui Xue
Qualcomm AI Research†
San Diego, USA
fangxue@qti.qualcomm.com

Shuai Zhang
Qualcomm AI Research
San Diego, USA
shuazhan@qti.qualcomm.com

Jiancheng Lyu
Qualcomm AI Research
San Diego, USA
jianlyu@qti.qualcomm.com

Yunling Zheng
Qualcomm AI Research
San Diego, USA
yunlzhen@qti.qualcomm.com

Yingyong Qi
Qualcomm AI Research
San Diego, USA
yingyong@qti.qualcomm.com

Jack Xin
Department of Mathematics
University of California, Irvine
Irvine, USA
jack.xin@uci.edu

Abstract

We present for video understanding (classification) a split space-time attention model, VideoSEMA, consisting of a scalable and efficient Mamba-like attention (SEMA) block in space and a softmax temporal attention in time. In each frame, SEMA attention applies a local window attention in parallel with a global averaging in a Mamba macro-architecture, which is called Mamba-like. Under certain rank conditions, we prove that the computationally cheaper split space-time attention is equivalent to full space-time attention. On benchmark K400 data sets, VideoSEMA out-performs heavier vision transformer and Mamba models. On benchmark SSv2 data, VideoSEMA leads in top-1 accuracy among models of similar parameter sizes. As image resolution scales up from standard 224^2 to 1024^2 on K400 and without fine-tuning, VideoSEMA degrades much more gracefully than VideoMamba in accuracy. It is promising to extend VideoSEMA to longer videos with a dilated/sparse temporal attention.

1 Introduction

Understanding complex spatiotemporal patterns in videos remains a fundamental challenge in computer vision, with application ranging from classification and segmentation to world modeling for

*Corresponding author
†Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc.

Preprint.

autonomous system such as robotics and self-driving vehicles. With the widespread of usage of multimodal models in combination with language models, recent advances have leveraged transformer based architecture and large foundation models to capture long range dependencies, achieving state of the art performance [2, 4, 15, 27, 25, 1]. Despite their success, these models predominately rely on Vision Transformer (ViT) backbones, which incur high computational costs for large video inputs. To alleviate this problem, some works explores Mamba [8] as backbone, which reduces the computation cost for long sequences, such as in VideoMamba [13]. Linear attention is another approach to reduce the cost of classical attention mechanism and WLiT demonstrates that it is an effective way to process video [21]. Lately, Mamba-like attention models have been developed as an efficient replacement of classical softmax attention in image application [10]. Mamba-like means to keep the macro-architecture of Mamba yet embed in it light weight non-Mamba attention blocks. Along this line, SEMA [24] is designed to mimic the exponential forgetting property of Mamba by an asymptotically guided global approximation of softmax attention. For a duality view and treatment of softmax and efficient attentions, see [18, 28]. An operator splitting and integro-differential equation perspective of transformer is in [22]. In this work, we explore the efficacy of SEMA backbone in video applications.

Our main contributions are:

- We develop a novel video model in the split space-time attention framework based on a Mamba-like scalable and efficient image backbone (SEMA,[24]).
- To our best knowledge, this is the first work to explore Mamba-like backbones for videos.
- We show theoretically that the split space-time attention is equivalent to the full space-time attention under specific rank conditions.
- We demonstrate on K400 (SSv2) classification tasks that VideoSEMA is an efficient model, outperforming much larger (comparable) parameter and flops size transformer and Mamba video models to date. Post-training and without fine-tuning on K400, it is much more robust than VideoMamba [13] as video frames scale up in size.

2 Related Works

Video representation learning has evolved from CNN to attention based models to hierarchical and efficient sequence architectures such as Mamba [8]. TimeSformer [2] demonstrated that factorized space time attention can replace 3D convolutions for video recognition, while MViT and MViTv2 [4, 15] introduced multiscale hierarchical Transformer that improves efficiency and scalability across image and video tasks. More recently, state space models have been explored for long range temporal modeling, with VideoMamba achieving competitive performance on video understanding tasks [13]. At scale, foundation models such as InternVideo2 unify self-supervised, constrastive, and generative objectives for multimodal video understanding [27]. Other efforts focus on efficiency and prediction such as adaptive token strategies improving training and inference flexibility [25], while V-JEPA 2 advanced self-supervised world modeling for motion prediction and long horizon reasoning in video [1]. However, many of these rely on ViT as a vision backbone, whereas our work explores a light weight Mamba-like model (SEMA [24]) as an alternative backbone to capture spatial and temporal features efficiently.

3 Methodology

3.1 Preliminary

3.1.1 Attention and SEMA

Attention is a core component of Transformer which is the main driving force of deep learning in recent years. Formally, given an input $x \in \mathbb{R}^{n\times d}$, then softmax full attention is defined as:

$$A(Q, K, V) = \text{softmax}(QK^T)V, \tag{1}$$

where $Q = xW_Q + b_Q, K = xW_K + b_K, V = xW_V + b_V$, for $W_Q, W_K, W_V \in \mathbb{R}^{d\times d}$ and $b_Q, b_K, b_V \in \mathbb{R}^{n\times d}$. We observe that compute the attention matrix (softmax(QK^T)) requires $\mathcal{O}(n^2)$ operations. Also as $n \to \infty$, the attention matrix disperses, i.e. tending to zero uniformly [24]. Thus, it is unable to distinguish the variations across the keys. On the other hand, SEMA [24] is designed to mimic the recurrent relation of Mamba [8] in the large token number limit with an averaging operation to approximate the global aspect of the attention matrix. This alleviates the burden in calculating full attention, with an access to global information in an efficient manner. Concretely,

$$SEMA(Q, K, V) := A_w(Q, K, V) + \left[\frac{1}{n}\sum_{j=1}^{n} v_j\right], \tag{2}$$

where $[\cdot]$ broadcasts the row n times to permit matrix addition and A_w is window attention [16] defined as:

$$A_w(Q, K, V) := \begin{bmatrix} \frac{\sum_{j\in J(1)} \exp(q_1 k_j^T) v_j}{\sum_{i\in J(1)} \exp(q_1 k_i^T)} \\ \vdots \\ \frac{\sum_{j\in J(n)} \exp(q_n k_j^T) v_j}{\sum_{i\in J(n)} \exp(q_n k_i^T)} \end{bmatrix}, \tag{3}$$

for some index set $J(m)$. An example is $J(m) = \{Mw+1, \ldots, (M+1)w\}$, where $M = \lfloor \frac{m-1}{w} \rfloor$.

The latter term of Eq. (2) is proven to be a good approximation for softmax(QK^T) as $n \to \infty$ under realistic assumption of the input feature x, and also with high probability that such an approximation holds [24]. Thus SEMA is an effective mechanism to process large input sequence with $\mathcal{O}(n)$ computational complexity.

3.1.2 Mamba as a Recursive Attention

Mamba [8] is a state space models to handle long input sequence with computational complexity of $\mathcal{O}(n)$. For complete derivation of state space models, we refer to [8, 9]. Concretely, Mamba is a map from x to y through the dynamical system:

$$h_t = A_t \odot h_{t-1} + B_t(\Delta_t \odot x_t), \tag{4}$$
$$y_t = C_t h_t + D \odot x_t, \tag{5}$$

where $x_t, \Delta_t \in \mathbb{R}^{1\times d}, A_t, h_t \in \mathbb{R}^{d\times d}, B_t \in \mathbb{R}^{d\times 1}$ and $y_t \in \mathbb{R}^{1\times d}, C_t \in \mathbb{R}^{1\times d}, D \in \mathbb{R}^{1\times d}$, and $\odot$ denotes the Hadamard product. It follows in discrete form that

$$y_m = \sum_{i=1}^{m} q_m \tilde{k}_i^T \tilde{v}_i + D \odot x_m, \tag{6}$$

where $\tilde{k}_i^T = \left(\prod_{j=1}^{m-i} A_{m-(j-1)}\right) \odot B_i$, $\tilde{v}_i = \Delta_i \odot x_i$, $\prod$ is short for multiple matrix products in the elementwise (Hadamard) sense, with the convention that $\prod$ acts as identity if the upper index is zero. The first term in (6) reveals the (Q, K, V) structure implicit in Mamba, while the second term can be understood as a skip connection (a casual masked attention).

3.2 Proposed Method - VideoSEMA

Given an input video $x \in \mathbb{R}^{C\times T\times h\times w}$. VideoSEMA first tokenizes using a 3D convolution (e.g. kernel=(1,4,4)) to project the input video into $X \in \mathbb{R}^{C\times T\times H\times W}$ of non-overlapping spatiotemporal patches where $H = h/4, W = w/4$. Second, we append a learnable classifier token at the end of the sequence. The last step of pre-processing is to add learned positional embedding and then temporal embedding. Concretely,

$$X = \text{3DConv}(x) \tag{7a}$$
$$X = [X, X_{cls}] + \text{Emb}_{pos} + \text{Emb}_{temp}, \tag{7b}$$

where $X_{cls} \in \mathbb{R}^{H\times W\times C}$ is classifier token, $\text{Emb}_{pos} \in \mathbb{R}^{H\times W\times C}$ and $\text{Emb}_{temp} \in \mathbb{R}^{T\times C}$ are learned positional and temporal embedding respectively. Here the addition is broadcast along the appropriate dimension to enable matrix addition.

Next, we will process the spatial and temporal information of the input via the VideoSEMA block. The VideoSEMA block is constructed as

$$y = Spatial\_SEMA(X), \tag{8a}$$
$$y' = LayerNorm(y), \tag{8b}$$
$$z' = Temp\_Attn(y'), \tag{8c}$$
$$z = X + z', \tag{8d}$$

where spatial function operates over the $H \times W$ dimension of the input, while the temporal function operates over the T dimension. To be precise, for $Q, K, V \in \mathbb{R}^{T\times H\times W\times C}$, we have

$$\begin{aligned} Spatial\_SEMA(q_{t,h,w}) := \frac{1}{HW}\sum_{i=1}^{H}\sum_{j=1}^{W} v_{t,i,j} \\ + \sum_{l,p\in I(h,w)} \frac{\exp(q_{t,h,w}k_{t,l,p}^T)}{\sum_{i,j\in I(h,w)}\exp(q_{t,h,w}k_{t,i,j}^T)} v_{t,l,p}, \end{aligned} \tag{9}$$

where $I(h, w)$ is the index set for the window, e.g. window size of 7. And temporal attention

$$Temp\_Attn(q_{t,h,w}) := \sum_{i=1}^{T} \frac{\exp(q_{t,h,w}k_{i,h,w}^T)}{\sum_{j=1}^{T}\exp(q_{t,h,w}k_{j,h,w}^T)} v_{i,h,w}. \tag{10}$$

The alternating spatial and temporal attention treatment in (8a) and (8c) can be viewed as an efficient operator splitting approximation of the full SEMA obtained by directly extending SEMA formula (2) to space and time. For video frames of moderate lengths considered here, a softmax attention in (8c) is affordable and effective as supported by our ablation study, making linear complexity approximations unnecessary in time. As seen later, VideoSEMA outperforms much heavier networks (Tab.1). As a future direction for handling longer videos, a sparse or dilated attention [3] in the temporal attention step is a promising alternative to leverage continuity of information in time at reduced computational costs.

VideoSEMA's macro-structure is shown in Fig.1, and is repeated in the network. Between two adjacent repeats, we use a convolutional layer to down-sample the spatial dimension of the video signal to produce a hierarchical network (similar to [10, 24, 16]). Lastly in Algorithm 1, the representation of the $[CLS]$ token is normalized before passing through a linear classification head.

3.3 Complexity

Given an input video $x \in \mathbb{R}^{T\times H\times W\times C}$, the joint space-time attention treats the video as a sequence of length $N = THW$. Therefore, the attention matrix has size $N \times N$, yielding computational complexity $\mathcal{O}((THW)^2C)$. In practice, this formulation is prohibitively expensive for videos due to the quadratic memory and compute cost of the attention matrix. Consequently, fully joint space time attention is rarely used in large scale video models [2].

The split space time attention approach works with a factorized (or separate spatial and temporal) attention, thereby lower the computational costs similar to operator splitting [20, 6]. In particular, TimeSformer [2] applies a split 2-step procedure: (1) spatial attention independently in each frame, and (2) temporal attention independently across frames at each spatial location. The spatial attention complexity in (1) is $\mathcal{O}(T(HW)^2C)$ as each of the T frames performs full attention over the HW spatial tokens. The temporal attention complexity in (2) is $\mathcal{O}(T^2HWC)$ as each spatial location performs full attention over T tokens. So the total complexity of the split attention design becomes $\mathcal{O}((T^2HW + T(HW)^2)C)$.

In contrast, VideoSEMA replaces full spatial attention with SEMA, whose complexity scales linearly with spatial dimension $\mathcal{O}(THWwC)$ where w denotes the window size. The temporal attention is

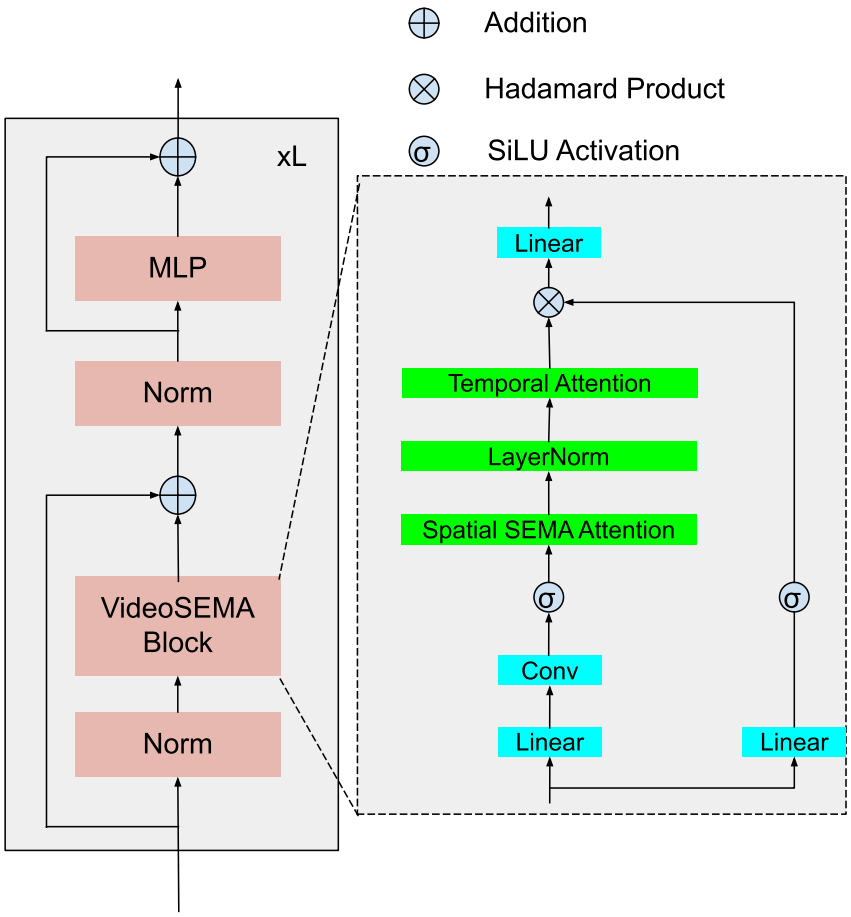


Figure 1: Overview of VideoSEMA macro-structure.

still performed globally, thus the total computational complexity of VideoSEMA is $\mathcal{O}((T^2HW + THWw)C)$, linear in spatial resolution HW. In the datasets of this paper, the number of video frames is at most 32 (i.e. $T \leq 32$ in Tab. 1, and $T \leq 16$ in Tab. 2 and Tab. 3). The main contribution to the complexity comes from spatial resolution $HW = (224)^2$. With a linear complexity in HW, VideoSema made considerable computational savings from TimeSformer [2], as seen in Table 1 on 16×224^2 resolution of K400 dataset where VideoSema's parameter size is a fraction 31/121 of TimeSformer's while achieving better accuracies. We present visualization of each space-time attention in Fig. 2.

4 Theoretical Explanation

4.1 A Simplified Setting

In this section, we show that in an ideal setting, the split space-time attention can be equivalent to full attention. Given an input $x \in \mathbb{R}^{N \times T \times d}$, where N, T, d represent spatial, temporal and channel dimensions respectively. For simplicity, let $V = x$, then the full attention for a fixed query q becomes:

$$A_{ST} = \sum_{i=1}^{N} \sum_{j=1}^{T} \eta_{ij} x_{ij}, \quad \eta_{ij} = \frac{\phi(q\, k_{ij}^T)}{\sum_{ij} \phi(q\, k_{ij}^T)}, \tag{11}$$

where k_{ij} and x_{ij} are vectors in $\mathbb{R}^d$ at space-time location (i, j), $1 \leq i \leq N$, $1 \leq j \leq T$. Here we define $\phi : \mathbb{R} \to \mathbb{R}^+$ to be any continuous function.

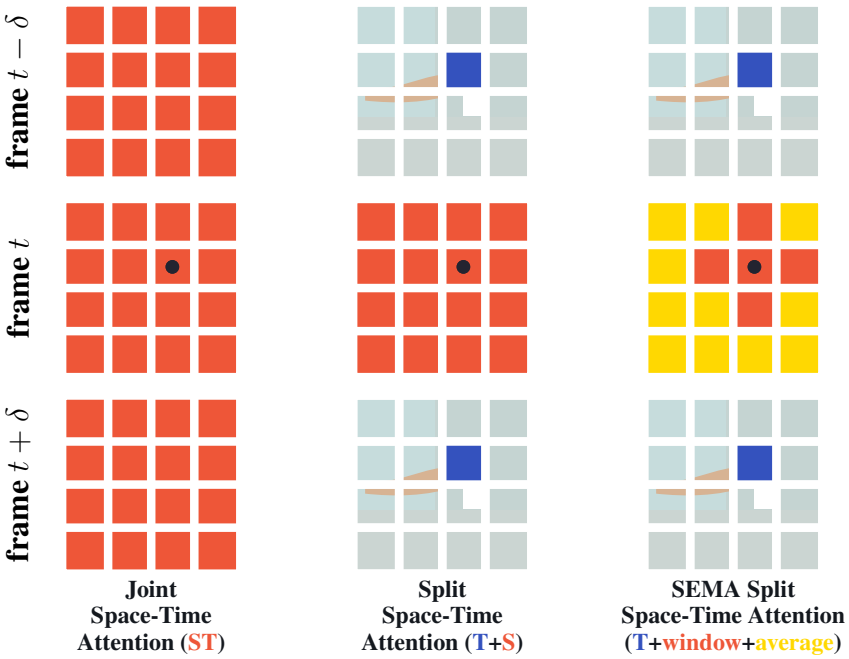


Figure 2: Visualization of space time attention types. For illustration, the dark dot denotes the query patch and colored patches show its self-attention space-time neighborhood under each scheme. Patches without color are not used for the self-attention computation of the query patch. Multiple colors within a scheme denote attentions separately applied along different dimensions, e.g., space and time for $(T + S)$. Note that self-attention is computed for every single patch in the video clip, i.e., every patch serves as a query. Although the attention pattern is shown for only two adjacent frames, it extends in the same fashion to all frames of the clip.

Now we compute split space-time attention A_{T+S} for the fixed query q. Without loss of generality, we have the split space-time function as the composite function of space and then time. The order does not matter, as it is up to a swap in the first and second dimensions of the input x. We have

Algorithm 1 Proposed VideoSEMA algorithm. Here cls, temp, pos denote classifier tokens, temporal embedding, and positional embedding respectively. When there are dimensions mismatch in binary operations, there are implicit reshaping/broadcasting to match the dimensions. On the right hand side, we denote the current running shape of x.

Input: Input x $(C \times T \times H \times W)$
Learnable Params: cls $\in \mathbb{R}^{H \times W \times C}$, temp $\in \mathbb{R}^{T \times C}$, pos $\in \mathbb{R}^{HW \times C}$

x = reshape(x) $(T \times H \times W \times C)$
x = concat((x, cls), dim=0) $((T+1) \times H \times W \times C)$
x = reshape(x) $((T+1) \times HW \times C)$
x = x+pos $((T+1) \times HW \times C)$
x = x+temp $((T+1) \times HW \times C)$
x = reshape(x) $((T+1) \times HW \times C)$
x = VideoSEMA(x) $((T+1) \times HW \times C)$
    Process spatial information via SEMA
    Process temporal information via classical attention
x = AvgPooling(x) $((T+1) \times C)$
Return $x[-1, :]$ $(1 \times C)$

$$A_{T+S} = \sum_{j=1}^{T} \alpha_j \left(\sum_{i=1}^{N} \beta_{N(j-1)+i}\, x_{ij} \right), \tag{12}$$

where α_j's represent attention score in the time dimension, and $\beta_\cdot$'s represent the frame-wise attention score. The equivalence $A_{ST} = A_{T+S}$ is realized if the following system of equations hold:

$$\eta_{ij} = \alpha_j \beta_{N(j-1)+i}, \tag{13}$$

$$\sum_{i,j=1}^{N,T} \eta_{ij} = 1, \tag{14}$$

$$\sum_{i=1}^{N} \beta_{N(j-1)+i} = 1, \;\; \forall j \in [1, \ldots, T], \tag{15}$$

$$\sum_{j=1}^{T} \alpha_j = 1. \tag{16}$$

An ideal solution for this system exists as follows. Suppose the standard full attention weights η_{ij} are given as in (11) and so (14) is satisfied. To match these attention scores by the split space-time attention (12), we let

$$\beta_{N(j-1)+i} = \frac{\eta_{ij}}{\sum_{l=1}^{N} \eta_{lj}}, \tag{17}$$

and

$$\alpha_j = \sum_{l=1}^{N} \eta_{lj}, \tag{18}$$

then (15) and (16) hold automatically. In practice however, one computes the weights $(\alpha_\cdot, \beta_\cdot)$ of the split attention (12) without knowledge of full attention weights. The corresponding factorization (13) may not satisfy normalization condition (14), and may not be equal to the (η_{ij}) in (11). The ideal solution (17)-(18) would only be an indirect theoretical target for the training of A_{T+S}.

4.2 Attention Factorization and Rank Conditions

Now we turn to standard attention. The previous section only addresses a single query, however in practice one works with multiple queries. Assume we have M queries, and let $m \in \{1, \ldots, M\}$ index the query row, and let η_{mij} be the full softmax attention coefficient from query m to spatial location i in frame j. The single-query construction above applies row-wise, so define

$$\alpha_{mj} = \sum_{l=1}^{N} \eta_{mlj}, \qquad \beta_{mi|j} = \frac{\eta_{mij}}{\alpha_{mj}}. \tag{19}$$

Since softmax coefficients are strictly positive, $\alpha_{mj} > 0$ and the definition is well-defined. Moreover,

$$\sum_{j=1}^{T} \alpha_{mj} = 1, \qquad \sum_{i=1}^{N} \beta_{mi|j} = 1, \qquad \alpha_{mj}\beta_{mi|j} = \eta_{mij}. \tag{20}$$

Thus the split output equals the full output for every query:

$$A_{ST}^{(m)} = \sum_{j=1}^{T} \sum_{i=1}^{N} \eta_{mij} x_{ij} = \sum_{j=1}^{T} \alpha_{mj} \left(\sum_{i=1}^{N} \beta_{mi|j} x_{ij} \right) = A_{T+S}^{(m)}. \tag{21}$$

Therefore split space time attention is an exact marginal-conditional decomposition of each row of the full attention matrix. The only remaining question is whether a chosen dot-product parameterization can realize the required temporal and spatial logits.

For positive normalized attention, $\alpha_{mj} > 0$, and these coefficients again satisfy $\alpha_{mj}\beta_{mi|j} = \eta_{mij}$. A particular split attention module realizes this decomposition exactly when its temporal branch can produce the distribution $\alpha_{m,:}$ and its spatial branch can produce the conditional distributions $\beta_{m:|j}$ under the same normalized ϕ operation. If ϕ is invertible on its positive range, one possible choice of temporal and spatial scores is any r_{mj} and s_{mij} satisfying

$$\phi(r_{mj}) = c_m\alpha_{mj}, \qquad \phi(s_{mij}) = c_{mj}\beta_{mi|j}, \tag{22}$$

for arbitrary positive constants c_m and c_{mj}.

Theorem 4.1 (Exact rank condition for normalized ϕ)**.** *Assume $\phi : \mathbb{R} \to \mathbb{R}^+$ is invertible on its positive range. Choose positive constants c_m and c_{mj} so that $c_m\alpha_{mj}$ and $c_{mj}\beta_{mi|j}$ lie in the range of ϕ, and define the target temporal score matrix $R^\phi \in \mathbb{R}^{M\times T}$ and target spatial score matrix $S^\phi \in \mathbb{R}^{M\times NT}$ by*

$$R^\phi_{mj} = \phi^{-1}(c_m\alpha_{mj}), \qquad S^\phi_{m,(j,i)} = \phi^{-1}(c_{mj}\beta_{mi|j}). \tag{23}$$

Let $d_T, d_S \in \mathbb{N}$ be the temporal and spatial head dimensions. If

$$d_T \geq \operatorname{rank}(R^\phi), \qquad d_S \geq \operatorname{rank}(S^\phi), \tag{24}$$

then there exist learned query and key matrices $Q^\tau \in \mathbb{R}^{M\times d_T}, K^\tau \in \mathbb{R}^{T\times d_T}$ and $Q^S \in \mathbb{R}^{M\times d_S}, K^S \in \mathbb{R}^{NT\times d_S}$ such that

$$Q^\tau(K^\tau)^T = R^\phi, \qquad Q^S(K^S)^T = S^\phi. \tag{25}$$

Consequently, normalized ϕ attention over the temporal scores produces $\alpha_{m,:}$, normalized ϕ attention over the spatial scores inside each frame produces $\beta_{m:|j}$, and split space time attention exactly equals full attention for every query row m.

Proof. The rank assumptions imply matrix factorizations $R^\phi = Q^\tau(K^\tau)^T$ and $S^\phi = Q^S(K^S)^T$, for example by the compact singular value decomposition. For each query row m, normalizing $\phi(R^\phi_{m,:})$ over the temporal index gives

$$\frac{\phi(R^\phi_{mj})}{\sum_{n=1}^T \phi(R^\phi_{mn})} = \frac{c_m\alpha_{mj}}{\sum_{n=1}^T c_m\alpha_{mn}} = \alpha_{mj}. \tag{26}$$

Likewise, for each fixed (m, j), normalizing $\phi(S^\phi_{m,(j,:)})$ over the spatial index gives

$$\frac{\phi(S^\phi_{m,(j,i)})}{\sum_{l=1}^N \phi(S^\phi_{m,(j,l)})} = \frac{c_{mj}\beta_{mi|j}}{\sum_{l=1}^N c_{mj}\beta_{ml|j}} = \beta_{mi|j}. \tag{27}$$

Therefore the split coefficient is $\alpha_{mj}\beta_{mi|j} = \eta_{mij}$ for all m, i, j. □

Remark 4.2 (Softmax as a special case). If $\phi(x) = \exp(x)$, then the above construction recovers the usual softmax factorization. In this case

$$\beta_{mi|j} = \frac{\exp(e_{mij})}{\sum_{l=1}^N \exp(e_{mlj})}, \qquad \alpha_{mj} = \frac{\exp(\tau_{mj})}{\sum_{n=1}^T \exp(\tau_{mn})}, \tag{28}$$

where the temporal score can be chosen as the frame-level log-sum-exp

$$\tau_{mj} = \log\left(\sum_{l=1}^N \exp(e_{mlj})\right), e_{mij} = q_m k_{ij}^T. \tag{29}$$

More generally, if the desired normalized coefficients η_{mij} are already known, softmax scores can be chosen as

$$r_{mj} = \log(\alpha_{mj}) + c_m, \qquad s_{mij} = \log(\beta_{mi|j}) + c_{mj}, \tag{30}$$

since adding a constant to all scores in a normalized softmax does not change the resulting distribution. This is the special case of Theorem 4.1 with $\phi^{-1}(x) = \log(x)$.

Now suppose that we are given a full rank condition as in Theorem 4.1, then we can realize the matrices Q^τ, K^τ, Q^S, K^S under a typical condition, for example, given $y \in \mathbb{R}^{M\times d_T}$, we want $Q^\tau = yW_Q$, for some $W_Q \in \mathbb{R}^{d_T\times d_T}$. Thus we solve the optimization problem

$$\min_W \|Q^\tau - yW\|_2, \tag{31}$$

and this has an exact solution if $col(Q^\tau) \subset col(y)$.

Theorem 4.3 (Low-rank approximation for normalized ϕ)**.** *Assume ϕ is invertible on its positive range, and let R^ϕ and S^ϕ be the target score matrices from Theorem 4.1. Let $\widehat{R}$ and $\widehat{S}$ be rank-constrained approximations with*

$$\operatorname{rank}(\widehat{R}) \le d_T, \qquad \operatorname{rank}(\widehat{S}) \le d_S. \tag{32}$$

Define the approximate normalized coefficients

$$\widehat{\alpha}_{mj} = \frac{\phi(\widehat{R}_{mj})}{\sum_{n=1}^{T}\phi(\widehat{R}_{mn})}, \qquad \widehat{\beta}_{mi|j} = \frac{\phi(\widehat{S}_{m,(j,i)})}{\sum_{l=1}^{N}\phi(\widehat{S}_{m,(j,l)})}, \tag{33}$$

and let $\widehat{\eta}_{mij} = \widehat{\alpha}_{mj}\widehat{\beta}_{mi|j}$. If

$$\epsilon_T = \max_m \|\widehat{\alpha}_{m,:} - \alpha_{m,:}\|_1, \qquad \epsilon_S = \max_{m,j} \|\widehat{\beta}_{m:|j} - \beta_{m:|j}\|_1, \tag{34}$$

then for every query row m,

$$\sum_{j=1}^{T}\sum_{i=1}^{N} |\widehat{\eta}_{mij} - \eta_{mij}| \le \epsilon_T + \epsilon_S. \tag{35}$$

If additionally $\|x_{ij}\|_2 \le B$ for all i, j, then

$$\|\widehat{A}^{(m)}_{T+S} - A^{(m)}_{ST}\|_2 \le B(\epsilon_T + \epsilon_S). \tag{36}$$

Proof. Add and subtract $\widehat{\alpha}_{mj}\beta_{mi|j}$:

$$|\widehat{\eta}_{mij} - \eta_{mij}| = |\widehat{\alpha}_{mj}\widehat{\beta}_{mi|j} - \alpha_{mj}\beta_{mi|j}| \tag{37}$$

$$\le \widehat{\alpha}_{mj}|\widehat{\beta}_{mi|j} - \beta_{mi|j}| + |\widehat{\alpha}_{mj} - \alpha_{mj}|\beta_{mi|j}. \tag{38}$$

Summing over i, j, and using that $\widehat{\alpha}$ and β are probability distributions, gives

$$\sum_{j=1}^{T}\sum_{i=1}^{N} |\widehat{\eta}_{mij} - \eta_{mij}| \le \|\widehat{\alpha}_{m,:} - \alpha_{m,:}\|_1 + \sum_{j=1}^{T}\widehat{\alpha}_{mj}\|\widehat{\beta}_{m:|j} - \beta_{m:|j}\|_1 \le \epsilon_T + \epsilon_S. \tag{39}$$

The output bound follows from

$$\|\widehat{A}^{(m)}_{T+S} - A^{(m)}_{ST}\|_2 \le \sum_{j=1}^{T}\sum_{i=1}^{N} |\widehat{\eta}_{mij} - \eta_{mij}|\|x_{ij}\|_2. \tag{40}$$

□

Remark 4.4. In particular, one may choose $\widehat{R}$ and $\widehat{S}$ as best rank-d_T and rank-d_S approximations of R^ϕ and S^ϕ in Frobenius norm. By the Eckart-Young theorem,

$$\|R^\phi - \widehat{R}\|_F^2 = \sum_{r>d_T}\sigma_r(R^\phi)^2, \qquad \|S^\phi - \widehat{S}\|_F^2 = \sum_{r>d_S}\sigma_r(S^\phi)^2, \tag{41}$$

where σ_r is the r singular value. Thus the approximation quality is controlled by the singular value tails of the target temporal and spatial score matrices, together with how sensitively the normalized ϕ map converts score errors into coefficient errors.

Remark 4.5. We are mostly dealing with long sequence, thus the number of query M or spatial N is large. Therefore the rank condition is quite strict, that is we are most likely to be in the scenario of Theorem 4.3. However when we work with short video clips as in SSv2 dataset, i.e. T is small, Q^τ and K^τ may be found exactly during training.

Table 1: Performance reported on standard K400 dataset. Here $F \times m \times n$ denotes by F the total FLOPs, m the number of testing segments, n the number of testing crops. Res: resolution, T: transformer. P(M): parameter size in millions. T1: top-1, T5: top-5. C: CNN, CT: CNN +T, M: Mamba, Ml: Mamba-like.

Method	Type	P(M)	FLOPs (G)	Res	T1(%)	T5(%)
X3D-XL [5]	C	20	$194 \times 3 \times 10$	16×224^2	80.4	94.6
Swin-T [17]	T	28	$88 \times 3 \times 4$	32×224^2	78.8	93.6
MViTv1-B [4]	CT	37	$70 \times 1 \times 5$	32×224^2	80.2	94.4
MViTv2-S [15]	CT	35	$64 \times 1 \times 5$	16×224^2	81.0	94.6
Uniformer-S [14]	CT	21	$42 \times 1 \times 4$	16×224^2	80.8	94.7
WLiT [21]	T	22	$21 \times 1 \times 4$	8×224^2	74.6	92.0
TimeSformer-L [2]	T	121	$2380 \times 3 \times 1$	16×224^2	80.7	94.7
Uniformer-S [14]	CT	311	$3992 \times 3 \times 4$	16×224^2	81.3	94.7
Mformer-HR [19]	T	311	$959 \times 3 \times 10$	16×336^2	81.1	95.2
VideoMamba-M [13]	M	74	$202 \times 3 \times 4$	16×224^2	81.9	95.4
VideoMamba-S [13]	M	26	$34 \times 3 \times 4$	8×224^2	79.3	94.2
VideoMamba-S [13]	M	26	$68 \times 3 \times 4$	16×224^2	80.8	94.8
VideoMamba-S [13]	M	26	$135 \times 3 \times 4$	32×224^2	81.5	95.2
VideoSEMA (Ours)	Ml	31	$46 \times 3 \times 4$	8×224^2	**81.3**	**94.9**
VideoSEMA (Ours)	Ml	31	$87 \times 3 \times 4$	16×224^2	**82.4**	**95.4**
VideoSEMA (Ours)	Ml	31	$172 \times 3 \times 4$	32×224^2	**82.6**	**95.2**

Table 2: Performance reported on standard SSv2 dataset. Here $F \times m \times n$ denotes by F the total FLOPs, m the number of testing segments, and n the number of testing crops. Res: resolution, T: transformer. P(M): parameter size in millions. T1: top-1, T5: top-5. C: CNN, CT: CNN +T, M: Mamba, Ml: Mamba-like.

Method	Type	P(M)	FLOPs (G)	Res	T1(%)	T5(%)
CT-Net$_{R50}$ [12]	C	21	$75 \times 1 \times 1$	16×224^2	64.5	89.3
TDN$_{R50}$ [26]	C	26	$75 \times 1 \times 1$	16×224^2	65.3	**91.6**
WLiT [21]	T	22	$50 \times 3 \times 1$	16×224^2	66.3	91.5
VideoMAE [23]	T	22	$57 \times 2 \times 3$	16×224^2	66.8	90.3
MViTv1-B [4]	CT	37	$71 \times 3 \times 1$	16×224^2	64.7	89.2
VideoMamba-S [13]	M	26	$34 \times 3 \times 2$	8×224^2	65.2	89.6
VideoMamba-S [13]	M	26	$68 \times 3 \times 2$	16×224^2	66.0	90.2
VideoSEMA (Ours)	Ml	31	$46 \times 3 \times 2$	8×224^2	**67.2**	91.0
VideoSEMA (Ours)	Ml	31	$87 \times 3 \times 2$	16×224^2	**67.3**	90.4

5 Experiments

5.1 Dataset

We evaluate our approach on two widely used large-scale video action recoginition benchmarks: Kinetic-400 (K400) [11] and Something-Something v2 (SSv2) [7]. K400 contains around 240000 training, 20000 validation, and 40000 testing videos spanning 400 human action categories, with clips source from YouTube and trimmed to focus on a single action that lasted around 10 seconds. The dataset is focused on human action from diverse scenes, actors, and viewpoints, making it a standard benchmark for learning high-level semantic. In contrast, SSv2 consists of around 220000 videos, with approximately 170000 training, 25000 validation and 27000 testing videos that last around 2-6 seconds. SSv2 places a strong temporal reasoning and fine-grained actions. Together, these datasets provide complementary evaluation settings.

5.2 Setting

A common strategy to train a video model is to pretrain a model with an image-based architecture [2, 13, 17, 23] on ImageNet-1K or ImageNet-21K, and then inflate the image model into a video

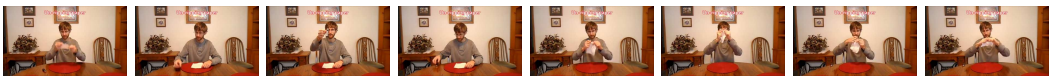
label: peeling potatoes
VideoSEMA prediction: peeling potatoes, P=0.9017
VideoMamba prediction: peeling apples, P=0.5198

(a) example depicts a person peeling potatoes, where VideoSEMA correctly predicts the action with high ($P = 0.9$), while VideoMamba incorrectly classifies it as peeling apples.

label: drinking shots
VideoSEMA prediction: drinking shots, P=0.5002
VideoMamba prediction: tasting beer, P=0.4820

(b) example shows a person drinking shots, which VideoSEMA correctly recognizes with moderate confidence $P = 0.5$.

label: ripping paper
VideoSEMA prediction: ripping paper, P=0.2000
VideoMamba prediction: folding napkins, P=0.4225



(c) example contains a video of a person ripping paper played in reverse time; VideoSEMA correctly identifies the action with low confidence, whereas VideoMamba fails to classify it correctly despite assigning moderate confidence to its prediction.

Figure 3: Qualitative comparison of VideoSEMA and VideoMamba on the K400 test set. Shown are representative test samples along with the predicted labels and corresponding highest confidence scores.

model. For fair comparison with VideoMamba, we pretrain the SEMA model on ImageNet-1K, then incorporate the temporal component and train the model on K400 and SSv2 to obtain the VideoSEMA results. For SEMA, we use the same training setup as described in [24]. We use the setting of the T model variant, i.e. 4 stages with hidden dimension of 64, 128, 256, and 512 respectively. To train VideoSEMA, we use a similar setup to VideoMamba. In particular, for K400 we use a set of 5 warmup epochs, 50 total epochs, a 0.35 stochastic depth rate, and 0.05 weight decay, with an initial learning rate of 2×10^{-4} using the AdamW optimizer. For SSv2, we use a set of 5 warmup epochs, 30 total epochs, a 0.35 stochastic depth rate, and 0.05 weight decay, with initial learning rate of 4×10^{-4}. We train and test VideoSEMA and VideoMamba on 8 NVIDIA RTX A6000 GPUs, each with 46G of memory. Code will be available upon publication.

5.3 Experimental Results

5.3.1 K400

We present the overall performance of VideoSEMA on K400 dataset in Table 1. Compared to VideoMamba under a similar computational budget, VideoSEMA's top-1 accuracies are 1-2% better across different input resolutions. Compared to other state of the art methods under similar training methodologies and computational budgets shown in the first row group of Table 1, VideoSEMA performs significantly better in both top-1 and top-5 metrics. For example, at an input resolution of 16×224^2, it is 1.4% better than MViTv2-S in top-1 and 0.8% in top-5.

Notably, the second row group of Table 1 includes models with substantially larger parameter counts and FLOPs. Despite operating under much smaller computational budget, VideoSEMA consistently achieves superior performance. This demonstrates that VideoSEMA is a

In addition, we present a few visual example comparisons between VideoSEMA and VideoMamba on somewhat challenging predictions in Fig. 3. Example (A) depicts a video of a person peeling potatoes. VideoSEMA correctly predicts the action with a high probability of 90%, while VideoMamba incorrectly classifies it as the similar label peeling apples. The global action of peeling is correct for both models; however, this particular video is challenging due to the local distinction between apples and potatoes, which occupy only a small region of the image. This could be explained by the window attention of SEMA, which keeps the fine details. Example (B) shows a person drinking shots. VideoSEMA correctly identifies the action with a medium probability of 50%, while VideoMamba misidentifies it as tasting beer. Drinking shots involves rapidly consuming a small amount of liquid, while tasting beer is a slower action. This shows that the attention in time between frames allows VideoSEMA to understand quick changes in motion between frames, while VideoMamba processes these tokens in sequential order, thus reducing its temporal capability. Lastly, example (C) shows a video of a person ripping paper recorded in reverse time. VideoSEMA classifies it correctly with low confidence of 20%, while VideoMamba predicts folding napkins. Because the video plays in reverse, the model needs strong temporal understanding. VideoSEMA with temporal attention allows the model to access frames in both directions simultaneously, which helps the model prediction. In contrast, VideoMamba's forward-direction processing observes the person putting the paper together and thus predicts folding napkins.

Table 3: Performance (ablation study) of VideoSEMA using various temporal processing units on K400 dataset. Here $F \times m \times n$ denotes by F the total FLOPs, m the number of testing segments, n the number of testing crops.

Time Component	# Params (M)	FLOPs (G)	Res	Top-1 (%)
3DConv (ker=7)	26	$40 \times 3 \times 4$	8×224^2	80.0
3DConv (ker=7)	26	$76 \times 3 \times 4$	16×224^2	79.3
3DConv (ker=13)	26	$76 \times 3 \times 4$	16×224^2	80.8
Mamba	36	$47 \times 3 \times 4$	8×224^2	76.3
Attention	31	$46 \times 3 \times 4$	8×224^2	**81.3**

5.3.2 SSv2

We present the overall performance of VideoSEMA on the SSv2 dataset in Table 2. We observe that with 8 frames inputs, VideoSEMA achieve a 2% improvement over VideoMamba-S, and with 16 frames inputs, it outperforms VideoMAE by 0.5%. Notably, using only 8 frames, VideoSEMA is able to outperform other models that utilize all 16 frames. This demonstrates that VideoSEMA achieves strong efficacy and temporal efficiency. We also note that the performance improvement with 16 frames is smaller compared to that with 8 frames. One possible explanation is the short temporal duration of videos in SSv2, thus limiting the benefit of longer input sequences.

5.4 Ablation Study

We extend the SEMA model from image domain to processing videos by adding a temporal processing unit. There are many standard choices for this component such as convolution, attention, and Mamba. In this subsection, we examine the performance of each choice. For convolution, to maintain relatively low parameter count, we employed a depthwise convolution. For Mamba, we use a single directional Mamba from VideoMamba [13]. And lastly, for attention we use softmax full attention. From Table 3, we observe that on K400, convolution performs relatively well, and

Table 4: Top-1 accuracy (%) at higher input resolutions to models pretrained on 16×224^2 videos of K400, without fine-tuning.

Method	16×224^2 (baseline)	16×512^2	16×1024^2
VideoMamba	80.8	74.4	40.8
VideoSEMA	82.4	75.3	54.6

as the number of frames increases, we need larger temporal convolutional kernel. Mamba on the other hand, performs poorly as a temporal processing unit. One possible explanation is that Mamba is applied only in the temporal dimension, resulting in independent operations on spatial positions. Lastly, since attention provides the best trade-off between accuracy and efficiency, we use attention as a temporal processing unit for VideoSEMA on the datasets here.

5.5 Scalability to Higher Resolution Videos

SEMA is designed to handle high-resolution frames. In this subsection, we examine the adaptability of the model when applied to larger input resolutions. We use the model pretrained on 224^2 and evaluate it on larger input sizes of 512^2 and 1024^2 with 16 frames on K400. From Tab. 4, we observe that VideoMamba and VideoSEMA perform similarly on 224^2; however, at 1024^2, the performance of VideoMamba degrades much more quickly compared to VideoSEMA. In particular, the performance gap between VideoMamba and VideoSEMA is about 13.8 percentage points. This demonstrates the robustness and scalability of VideoSEMA for large input resolutions.

6 Conclusion

We introduced a space-time attention model (VideoSEMA) consisting of a scalable and efficient Mamba-like attention (SEMA) in space and a regular attention in time. For moderate number of video frames, the model out-performs recent space-time transformers and video-mamba of larger (comparable) sizes on benchmark K400 (SSv2) datasets. In future work, we plan to scale the model to longer videos by using dilated/sparse attention [3] in time and evaluate the model's robustness on higher spatial resolutions across more datasets.

Acknowledgments

The work was partly supported by NSF grants DMS-2219904, DMS-2309520, and a Qualcomm Gift Award. NTT was also funded by a Faculty Endowed Fellowship and the Graduate Scholar Success Fund from the University of California, Irvine.

References

[1] Mido Assran, Adrien Bardes, David Fan, Quentin Garrido, Russell Howes, Mojtaba, Komeili, Matthew Muckley, Ammar Rizvi, Claire Roberts, Koustuv Sinha, Artem Zholus, Sergio Arnaud, Abha Gejji, Ada Martin, Francois Robert Hogan, Daniel Dugas, Piotr Bojanowski, Vasil Khalidov, Patrick Labatut, Francisco Massa, Marc Szafraniec, Kapil Krishnakumar, Yong Li, Xiaodong Ma, Sarath Chandar, Franziska Meier, Yann LeCun, Michael Rabbat, and Nicolas Ballas. V-jepa 2: Self-supervised video models enable understanding, prediction and planning. *arXiv:2506.09985*, 2025.

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021.

[3] J. Ding, S. Ma, L. Dong, X. Zhang, S. Huang, W. Wang, N. Zheng, and F. Wei. Longnet: Scaling transformers to 1,000,000,000 tokens. *arXiv preprint arXiv:2307.02486*, 2023.

[4] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6824–6835, October 2021.

[5] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[6] R. Glowinski, S. J. Osher, and W. Yin. Splitting methods in communication, imaging, science, and engineering. *Springer*, 2017.

[7] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. The "something something" video database for learning and evaluating visual common sense. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5843–5851, 2017.

[8] Albert Gu and Tri Dao. Mamba: Linear-time sequence modeling with selective state spaces. In *First Conference on Language Modeling*, 2024.

[9] Albert Gu, Tri Dao, Stefano Ermon, Atri Rudra, and Christopher Ré. Hippo: Recurrent memory with optimal polynomial projections. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1474–1487. Curran Associates, Inc., 2020.

[10] Dongchen Han, Ziyi Wang, Zhuofan Xia, Yizeng Han, Yifan Pu, Chunjiang Ge, Jun Song, Shiji Song, Bo Zheng, and Gao Huang. Demystify Mamba in Vision: A Linear Attention Perspective. *NeurIPS*, 2024.

[11] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *arXiv:1705.06950*, 2017.

[12] Kunchang Li, Xianhang Li, Yali Wang, Jun Wang, and Yu Qiao. CT-Net: Channel tensorization network for video classification. In *International Conference on Learning Representations*, 2021.

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

[14] Kunchang Li, Yali Wang, Gao Peng, Guanglu Song, Yu Liu, Hongsheng Li, and Yu Qiao. Uniformer: Unified transformer for efficient spatial-temporal representation learning. In *International Conference on Learning Representations*, 2022.

[15] Yanghao Li, Chao-Yuan Wu, Haoqi Fan, Karttikeya Mangalam, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. MViTv2: Improved multiscale vision transformers for classification and detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4804–4814, June 2022.

[16] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 10012–10022, October 2021.

[17] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3202–3211, June 2022.

[18] Tan Nguyen, Tam Nguyen, Nhat Ho, Andrea Bertozzi, Richard Baraniuk, and Stanley Osher. A primal-dual framework for transformers and neural networks. *in Proc. of ICLR*, 2023.

[19] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and Joao F. Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[20] G. Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, 1968.

[21] Ruochen Sun, Tian Zhang, Yiming Wan, Feng Zhang, and Jian Wei. WLiT: Windows and linear transformer for video action recognition. *Sensors (Basel, Switzerland)*, 23(3):1616, 2023.

[22] Xue-Cheng Tai, Hao Liu, Lingfeng Li, and Raymond H Chan. A mathematical explanation of transformers. *arXiv:2510.03989*, 2025.

[23] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 10078–10093. Curran Associates, Inc., 2022.

[24] Nhat Thanh Tran, Fanghui Xue, Shuai Zhang, Jiancheng Lyu, Yunling Zheng, Yingyong Qi, and Jack Xin. SEMA: a scalable and efficient mamba like attention via token localization and averaging. *arXiv:2506.08297; in Proc. of ICML*, 2026.

[25] Chenting Wang, Kunchang Li, Tianxiang Jiang, Xiangyu Zeng, Yi Wang, and Limin Wang. Make your training flexible: Towards deployment-efficient video models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 23880–23891, October 2025.

[26] Limin Wang, Zhan Tong, Bin Ji, and Gangshan Wu. Tdn: Temporal difference networks for efficient action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1895–1904, June 2021.

[27] Yi Wang, Kunchang Li, Xinhao Li, Jiashuo Yu, Yinan He, Guo Chen, Baoqi Pei, Rongkun Zheng, Zun Wang, Yansong Shi, Tianxiang Jiang, Songze Li, Jilan Xu, Hongjie Zhang, Yifei Huang, Yu Qiao, Yali Wang, and Limin Wang. Internvideo2: Scaling foundation models for multimodal video understanding. In Aleš Leonardis, Elisa Ricci, Stefan Roth, Olga Russakovsky, Torsten Sattler, and Gül Varol, editors, *Computer Vision – ECCV 2024*, pages 396–416, Cham, 2025. Springer Nature Switzerland.

[28] Y. Zheng, Z. Xu, F. Xue, B. Yang, J. Lyu, S. Zhang, Y. Qi, and J. Xin. AFIDAF: Alternating Fourier and Image Domain Adaptive Filters as an Efficient Alternative to Attention in ViTs. *International Symposium of Visual Computing, Reno, NV*, 15046:17–30, 2024.