10-2023

PosMLP-Video: Spatial and temporal relative position encoding for efficient video recognition

Yanbin HAO

Diansong ZHOU

Zhicai WANG

Chong-wah NGO
Singapore Management University, cwngo@smu.edu.sg

Xiangnan HE

See next page for additional authors

Citation

HAO, Yanbin; ZHOU, Diansong; WANG, Zhicai; NGO, Chong-wah; HE, Xiangnan; and WANG, Meng. PosMLP-Video: Spatial and temporal relative position encoding for efficient video recognition. (2023). 1-20.
Available at: https://ink.library.smu.edu.sg/sis_research/8256

Author

Yanbin HAO, Diansong ZHOU, Zhicai WANG, Chong-wah NGO, Xiangnan HE, and Meng WANG

PosMLP-Video: Spatial and Temporal Relative Position Encoding for Efficient Video Recognition

Yanbin Hao (haoyanbin@hotmail.com)
University of Science and Technology of China
Diansong Zhou
University of Science and Technology of China
Zhicai Wang
University of Science and Technology of China
Chong-Wah Ngo
Singapore Management University
Xiangnan He
University of Science and Technology of China
Meng Wang
University of Science and Technology of China

Research Article

Keywords: Positional encoding, spatio-temporal modeling, multi-layer perceptron, video recognition

Posted Date: October 27th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3485088/v1

Additional Declarations:
Competing interests: The authors declare no competing interests.

PosMLP-Video: Spatial and Temporal Relative Position Encoding for Efficient Video Recognition

Yanbin Hao[1]†, Diansong Zhou[1]†, Zhicai Wang[1], Chong-Wah Ngo[2], Xiangnan He[1], Meng Wang[3]

[1]School of Information Science and Technology, University of Science and Technology of China, No.96, JinZhai Road, Hefei, 230026, Anhui, China.
[2]School of Computing and Information Systems, Singapore Management University, 80 Stamford Road, 178902, Singapore.
[3]School of Computer Science and Information Engineering, Hefei University of Technology, No. 485, Danxia Road, Hefei , 230601, Anhui, China.

Contributing authors: haoyanbin@hotmail.com; zhouds1918@gmail.com; wangzhic@mail.ustc.edu.cn; cwngo@smu.edu.sg; xiangnanhe@gmail.com; eric.mengwang@gmail.com;
†These authors contributed equally to this work.

Abstract

In recent years, vision Transformers and MLPs have demonstrated remarkable performance in image understanding tasks. However, their inherently dense computational operators, such as self-attention and token-mixing layers, pose significant challenges when applied to spatio-temporal video data. To address this gap, we propose PosMLP-Video, a lightweight yet powerful MLP-like backbone for video recognition. Instead of dense operators, we use efficient relative positional encoding (RPE) to build pairwise token relations, leveraging small-sized parameterized relative position biases to obtain each relation score. Specifically, to enable spatio-temporal modeling, we extend the image PosMLP's positional gating unit to temporal, spatial, and spatio-temporal variants, namely PoTGU, PoSGU, and PoSTGU, respectively. These gating units can be feasibly combined into three types of spatio-temporal factorized positional MLP blocks, which not only decrease model complexity but also maintain good performance. Additionally, we improve the locality of modeling using window partitioning and enrich relative positional relationships using channel grouping. Experimental results demonstrate that PosMLP-Video achieves competitive speed-accuracy trade-offs compared to the previous state-of-the-art models. In particular, PosMLP-Video pre-trained on ImageNet1K achieves 59.0%/70.3% top-1 accuracy on Something-Something V1/V2 and 82.1% top-1 accuracy on Kinetics-400 while requiring much fewer parameters and FLOPs than other models. The code will be made publicly available.

Keywords: Positional encoding, spatio-temporal modeling, multi-layer perceptron, video recognition

1 Introduction

Neural networks have evolved from classic CNNs He, Zhang, Ren, and Sun (2016); Hu, Shen, and Sun (2018); Krizhevsky, Sutskever, and Hinton (2017); Szegedy, Ioffe, Vanhoucke, and Alemi (2017) to convolution-free Transformers Dosovitskiy et al.

(2020); Huang et al. (2021); Z. Liu et al. (2021); Touvron et al. (2021); W. Wang et al. (2021) to the more recently self-attention-free MLPs S. Chen, Xie, Ge, Liang, and Luo (2021); Guo et al. (2022); Hou et al. (2022); Lian, Yu, Sun, and Gao (2021); H. Liu, Dai, So, and Le (2021); Tolstikhin et al. (2021); Z. Wang et al. (2022); Yu, Li, Cai, Sun, and Li (2022). Generally, vision Transformers and MLPs perform more impressively in large-scale image processing tasks thanks to their innate capacity for long-range dependency modeling with self-attention and token-mixing. Extending these image-based networks for video processing, nevertheless, will dramatically increase the model complexity and processing time. Inspired by the image-based MLP models H. Liu et al. (2021); Z. Wang et al. (2022), which are capable of maintaining a good balance between efficiency and recognition accuracy, this paper proposes MLP-like architectures for efficient and effective video recognition.

A direct way of extending a 2D image model for 3D video processing is by involving the time axis in the spatial operator. For example, C3D Tran, Bourdev, Fergus, Torresani, and Paluri (2015) and I3D Carreira and Zisserman (2017) extend 2D convolution to 3D convolution. Similarly, Transformers can extend self-attention for spatio-temporal modeling but with T^2 more pairwise computations, where T is the time-length. Extending MLPs for video will also result in substantial growth of tokens in the token-FC layers. To alleviate this problem, the factorization of space and time has been widely adopted by video architectures. For example, P3D Qiu, Yao, and Mei (2017) decomposes the 3D convolution kernel as the combination of 1D kernel + 2D kernel. Similarly, ViViT Arnab et al. (2021), MorphMLP D.J. Zhang et al. (2022) and MLP-3D Qiu, Yao, Ngo, and Mei (2022) divide the 3D spatio-temporal tokens as spatial tokens and temporal tokens. In addition to the factorization mechanism, there are also other strategies, such as temporal Lin, Gan, and Han (2019) or token shift H. Zhang, Hao, and Ngo (2021), and window partitioning Z. Liu et al. (2021), for lightweight models. Despite the significant reduction in model complexity, these models still struggle with excessive number of parameters and pairwise self-attention computations.

In this paper, we present **PosMLP-Video**, a novel MLP-like architecture for video recognition that offers a superior speed-accuracy trade-off compared to the Transformers and MLPs, while being efficient and lightly parameterized. Our PosMLP-Video is an extension of the image PosMLP Z. Wang et al. (2022) that leverages the relative positional encoding (RPE) to compute pairwise token relations and incorporates a part-to-part element-wise gating mechanism as gMLP H. Liu et al. (2021) for cross-token interactions. In this work, we explore the potential of learnable RPE (LRPE) in constructing spatial and temporal token relations. We demonstrate that LRPE is highly sensitive to temporal order and achieves efficient pairwise token relation computation by searching from a learnable relative position bias dictionary with minimal parameters. Specifically, only $2N - 1$ position biases are parameterized for a given N tokens, compared to N^2 for token-mixing in MLPs H. Liu et al. (2021); Tolstikhin et al. (2021) and $3N^2$ for self-attention in Transformers Arnab et al. (2021); Dosovitskiy et al. (2020); Z. Liu et al. (2022). To further enhance positional relationships, we split tokens along the channel dimension into multiple groups and learn a specific relative position bias dictionary for each group. The channel grouping operation increases the LRPE-based model complexity to g (number of groups) times, while its overall model complexity remains $O(N)$ as $g << N$.

To factorize space and time modeling, we design three types of gating units: 1D positional temporal gating unit (**PoTGU**), 2D positional spatial gating unit (**PoSGU**), and 3D positional spatio-temporal gating unit (**PoSTGU**). These units separately capture axis-preferred cross-token interactions and can be configured into various MLP-like architectures. We present three spatio-temporal factorized video PosMLP blocks, including two cascaded blocks (**PoTGU→PoSGU** and **PoSGU→PoTGU**) and one paralleled block (**PoTGU+PoSGU**). In addition, we also present the joint spatio-temporal video PosMLP block (**PoSTGU**) as a comparison. As demonstrated in the experiment, the spatio-temporal factorized variants have smaller model sizes and obtain higher recognition accuracies than the joint version. Among them, the paralleled version, **PosTGU+PosSGU**, achieves the best performance. Finally, these blocks are integrated into a hierarchical framework as D.J. Zhang et al. (2022) to build up PosMLP-Video networks, which are further improved through the adoption of a window partitioning strategy similar to that used in Swin Transformer Z. Liu et al. (2021).

PosMLP-Video can be readily pre-trained on off-the-shelf image datasets by replacing the temporal unit PoTGU with a residual connection. In our experiment, we demonstrate that, despite being pre-trained on the relatively smaller dataset ImagNet1K, our PosMLP-Video can achieve comparable or even superior performance to competing methods that are pre-trained on more extensive datasets such as ImageNet21K and Kinetics-400/600. Overall, through various video recognition tasks, PosMLP-Video proves to be competitive in recognition rate with better model efficiency than the existing video CNNs, Transformers, and MLPs.

2 Related Works

We start by introducing video neural network models, including video CNNs, Transformers and MLPs. Then a review of position encoding in sequential data modeling is presented.

Video CNNs extends 2D CNN for spatio-temporal modeling. The earliest video CNNs such as C3D Tran et al. (2015) and I3D Carreira and Zisserman (2017) directly expand the 2D convolution to a 3D version. They typically have hefty parameters and high computational overheads because of the added time axis. Later, research attempts are made to devise lightweight convolutional operators, such as spatio-temporal factorized convolutions Qiu et al. (2017); Tran et al. (2018); Xie, Sun, Huang, Tu, and Murphy (2018), temporal shift module (TSM) Lin et al. (2019), group spatio-temporal network Luo and Yuille (2019), and SlowFast network Fan, Li, Xiong, Lo, and Feichtenhofer (2020). More recently, to enhance CNN features with larger receptive contexts, feature contextualization has also been studied, for example, non-local neural network X. Wang, Girshick, Gupta, and He (2018), temporal excitation and aggregation (TEA) Y. Li et al. (2020), and group contextualization Hao, Zhang, Ngo, and He (2022).

Video Transformers are developed using the cutting-edge technology Transformer Vaswani et al. (2017). Vision Transformers Dong et al. (2022); Dosovitskiy et al. (2020); Z. Liu et al. (2021); Touvron et al. (2021); W. Wang et al. (2021); B. Wu et al. (2020); Yuan et al. (2021) have demonstrated the promising potential of Transformer in vision tasks. Unfortunately, due to the high pairwise computational cost, directly applying 3D self-attention to process spatio-temporal video material is not a viable option. As a result, video Transformers Arnab et al. (2021); Bulat, Perez Rua, Sudhakaran, Martinez, and Tzimiropoulos (2021); J. Chen and Ho (2022); Fan et al. (2021); Y. Li et al. (2022); Z. Liu et al. (2022); Yan et al. (2022); H. Zhang et al. (2021) focus their efforts mostly on developing efficient space-time self-attention modules. ViViT Arnab et al. (2021) and TimeSformer Bertasius, Wang, and Torresani (2021), for example, investigate alternative factorized space-time attentions and find superior speed-accuracy trade-offs than 3D self-attention. To enable temporal interaction between tokens, TokShift H. Zhang et al. (2021) employs the zero-parameter and zero-FLOPs channel shift operation. MViT Fan et al. (2021) leverages multi head pooling attention to achieve multiscale hierarchical feature aggregation while concurrently reducing computation cost. Video Swin Z. Liu et al. (2022), on the other hand, proposes to reduce the computation of 3D self-attention by slicing the space-time area into smaller windows while preserving spatio-temporal locality.

Video MLPs. In recent years, vision MLPs S. Chen et al. (2021); Guo et al. (2022); Hou et al. (2022); Lian et al. (2021); H. Liu et al. (2021); Tolstikhin et al. (2021); Z. Wang et al. (2022); Yu et al. (2022) merely utilize token-mixing and channel-mixing MLP to achieve cross-token and cross-channel interaction, respectively. Particularly, the token-mixing (i.e., token-FC) layer is expected to replace the self-attention layer in a Transformer. When processing video data, however, a huge number of MLP layers might still result in dense parameters and computations. The existing video MLPs include MorphMLP D.J. Zhang et al. (2022) and MLP-3D Qiu et al. (2022). MorphMLP designs two types of MLP layers, i.e., MorphFC$_s$ and MorphFC$_t$, where MorphFC$_s$ captures the spatial semantics by gradually increasing the token receptive field while MorphFC$_t$ achieves long-range temporal dependency by applying a linear projection to the temporal concatenated feature chunk. The space-time factorization can significantly reduce the computation cost. MLP-3D decomposes the token mixing MLP along height, width and time axes and aggregates their outputs using weighted summation. To further improve model efficiency, they suggest a grouped time mixing operation that can also achieve parameter sharing across projections.

Position encoding in sequential data modeling. The order information is important for understanding sequential data such as natural language and video. However, since self-attention is typically order-independent, Transformers commonly utilize the position encoding (PE) to maintain order information Raffel et al. (2020). Two widely used PE methods are absolute position encoding (APE), e.g., the sinusoidal position signal Vaswani et al. (2017) and learned position embeddings, and relative position encoding (RPE), e.g., learnable RPE (LRPE) Shaw, Uszkoreit, and Vaswani (2018) and quadratic PE (QPE) Cordonnier, Loukas, and Jaggi (2019). APE is usually used as a simple way of adding the signal or embedding to the corresponding token. Many text and vision Transformers Arnab et al. (2021); J. Chen and Ho (2022); Dosovitskiy et al. (2020); Fan et al. (2021); Vaswani et al. (2017); J. Wang and Torresani (2022); Yan et al. (2022); Yang et al. (2022) has demonstrated its positive effect on sequential modeling. In contrast to APE, RPE computes the relative position embeddings based on the relative position offset. Since the position offset exactly overlaps the key-query offset of self-attention, it is a common method for collapsing the embedding to a learnable scalar and adding it to the corresponding attention score, which is also the principle of LRPE. Compared with APE, RPE regularly provides more significant performance improvement for Transformers d'Ascoli et al. (2021); Y. Li et al. (2022); Z. Liu et al. (2021, 2022); Shaw et al. (2018); Z. Wang et al. (2022); C.-Y. Wu et al. (2022) in language and vision tasks. In addition, despite the lack of explicit use of PE in CNNs and MLPs, it has been demonstrated that both can capture position information implicitly in their model learning Islam, Kowal, Jia, Derpanis, and Bruce (2021a, 2021b); Kayhan and Gemert (2020); Z. Wang et al. (2022). These findings motivate us to develop a pure RPE-based operator for efficiently processing time-order-dependent video data.

3 Approach

In this section, we first introduce LRPE. Then, we elaborate on the design details of the LRPE-based spatial and temporal gating units, and the four types of video PosMLP blocks. Finally, we stack the video PosMLP blocks hierarchically to construct a set of PosMLP-Video network backbones.

3.1 Learnable Relative Position Encoding

RPE computes the pairwise relationships between tokens based on the position offset. Following works He, Gkioxari, Dollár, and Girshick (2017); Z. Liu et al. (2021, 2022); Raffel et al. (2020), a relatively efficient learnable RPE (LRPE) is introduced to determine the relevance score of each position pair by using a learned relative position bias dictionary. The relative position offset of a token pair totally dictates the positional bias "choosing". Therefore, LRPE is order-sensitive. We verify the sensitivity in the experiment. LRPE is often utilized to enhance self-attention. Given a 2D $M \times M$ image window as an example, the enhanced self-attention in Z. Liu et al. (2021) is calculated for each head as follows

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{Softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}} + \mathbf{R}\right)\mathbf{V}, \tag{1}$$

where $\mathbf{Q}, \mathbf{K}, \mathbf{V} \in \mathbb{R}^{M^2 \times d}$ denote the query, key and value matrices in a transformer, d is the feature dimension, M^2 is the total number of tokens in a 2D window, and $\mathbf{R} \in \mathbb{R}^{M^2 \times M^2}$ is the relative position bias matrix. Specifically, each element r_{ij} in $\mathbf{R}$ is searched from the learned bias dictionary $\mathbf{P} \in \mathbb{R}^{(2M-1)\times(2M-1)}$ indexed by the position offset.

3.2 Positional Spatial and Temporal Gating Units

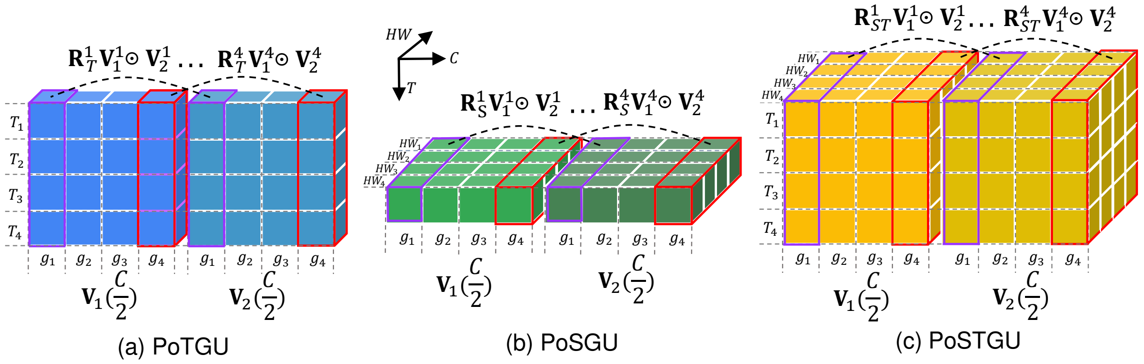
Utilizing the aforementioned LRPE approach, we design three positional spatial and temporal gating units for video data modeling: **PoTGU** (temporal), **PoSGU** (spatial), and **PoSTGU** (spatio-temporal). To capture various axial relations, they individually take into account the 1D temporal (PoTGU), 2D spatial (PoSGU), and 3D spatio-temporal (PoSTGU) variances.

The computation pipeline of the three units follows image PosMLP Z. Wang et al. (2022) and gMLP H. Liu et al. (2021). Formally, we have the general formula as

$$\mathbf{Z} = \mathbf{R}\mathbf{X}_1 \odot \mathbf{X}_2, \tag{2}$$

where $\mathbf{X}_1, \mathbf{X}_2 \in \mathbb{R}^{M^2 \times \frac{d}{2}}$ are two independent parts of $\mathbf{X}$ along the channel dimension, $\mathbf{Z} \in \mathbb{R}^{M^2 \times \frac{d}{2}}$ is the output with cross-token interactions, and $\odot$ denotes element-wise multiplication. Note that, for the sake of simplicity, we omit the bias term in the equation, as well as in the subsequent ones. Here, $\mathbf{R}$ plays a similar role as the linear projection weight matrix of gMLP' spatial gating unit (SGU) that contracts information over tokens. But, $\mathbf{R}$ further considers the relative position differences between tokens which makes it more suitable for sequential dependency modeling. In the experiment, we verify this by replacing all position units with gMLP's units and observing significant performance drops.

In PosMLP-Video, all position units act on the 3D visual window. Firstly, let's denote the token embeddings of a 3D window as $\mathbf{V} \in \mathbb{R}^{T \times H \times W \times C}$,



$\mathbf{R}_T^1\mathbf{V}_1^1 \odot \mathbf{V}_2^1 \dots \mathbf{R}_T^4\mathbf{V}_1^4 \odot \mathbf{V}_2^4$
T_1
T_2
T_3
T_4
g_1 g_2 g_3 g_4
$\mathbf{V}_1(\frac{C}{2})$
$\mathbf{V}_2(\frac{C}{2})$
(a) PoTGU
HW
C
T
$\mathbf{R}_S^1\mathbf{V}_1^1 \odot \mathbf{V}_2^1 \dots \mathbf{R}_S^4\mathbf{V}_1^4 \odot \mathbf{V}_2^4$
HW_1 HW_2 HW_3 HW_4
(b) PoSGU
$\mathbf{R}_{ST}^1\mathbf{V}_1^1 \odot \mathbf{V}_2^1 \dots \mathbf{R}_{ST}^4\mathbf{V}_1^4 \odot \mathbf{V}_2^4$
(c) PoSTGU

Fig. 1 Positional spatial and temporal gating units.

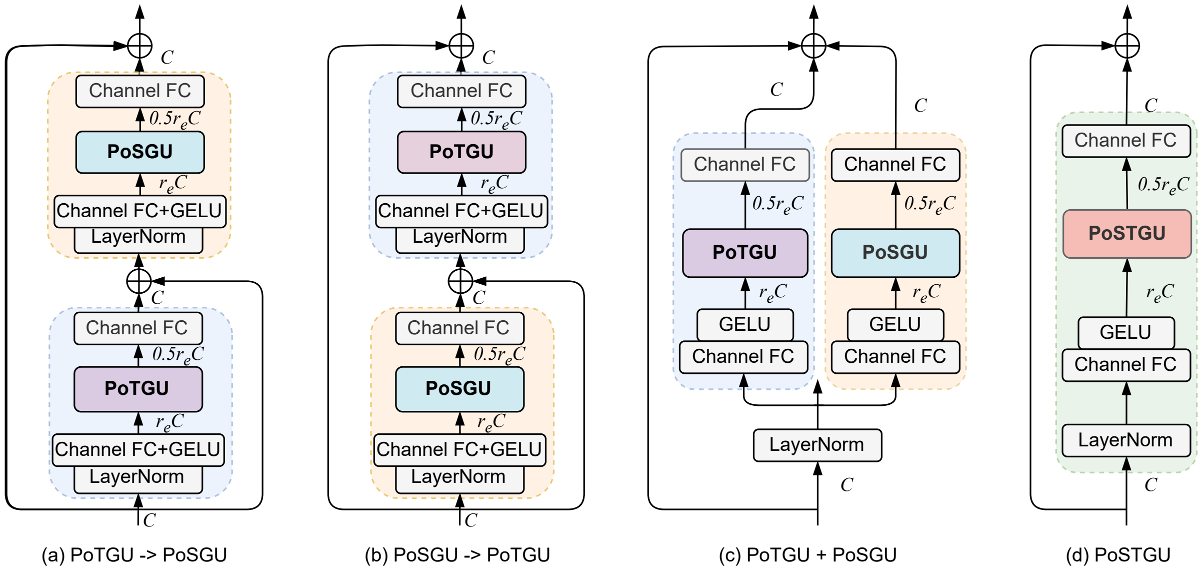


Fig. 2 The schema of the four factorized spatio-temporal PosMLP blocks. The channel expansion ratio r_e is set to 2 and 4 in our implementation.

where $T, H/W, C$ are the window-time, window-height/width and channel dimension, respectively. We first split $\mathbf{V}$ into $\mathbf{V}_1 \in \mathbb{R}^{T\times H\times W\times \frac{C}{2}}$ and $\mathbf{V}_2 \in \mathbb{R}^{T\times H\times W\times \frac{C}{2}}$, and then divide each into g feature groups: $\{\mathbf{V}_1^1, \mathbf{V}_1^2, \cdots, \mathbf{V}_1^g\} \in \mathbb{R}^{T\times H\times W\times \frac{C}{2g}}$ and $\{\mathbf{V}_2^1, \mathbf{V}_2^2, \cdots, \mathbf{V}_2^g\} \in \mathbb{R}^{T\times H\times W\times \frac{C}{2g}}$. The channel grouping strategy follows Hao et al. (2022); Z. Wang et al. (2022), which is to increase the multiformity of relative position biases. Below, we present the designs of PoTGU, PoSGU, and PoSTGU in detail.

PoTGU aims to model the 1D temporal relation between tokens, as shown in Figure 1(a). PoTGU parameterizes g temporal relative position dictionaries $\{\mathbf{P}_T^1, \mathbf{P}_T^2, \cdots, \mathbf{P}_T^g\} \in \mathbb{R}^{2T-1}$, based on which we build g temporal relative position matrices $\{\mathbf{R}_T^1, \mathbf{R}_T^2, \cdots, \mathbf{R}_T^g\} \in \mathbb{R}^{T\times T}$ following the LRPE principle. Specifically, for the i-th token group (i.e., $i \in [1, 2, \cdots, g]$) and j-th spatial position (i.e., $j \in [1, 2, \cdots, HW]$), the refined output token feature $\mathbf{Z}_T$ is computed as

$$\mathbf{Z}_T^{i,j} = \mathbf{R}_T^i \mathbf{V}_1^{i,j} \odot \mathbf{V}_2^{i,j}, \tag{3}$$

where $\mathbf{V}_1^{i,j}, \mathbf{V}_2^{i,j}, \mathbf{Z}_T^{i,j}$ are with the size of $T \times \frac{C}{2g}$.

PoSGU, in contrast to PoTGU, pays attention to the 2D spatial relation between tokens, as shown in Figure 1(b). Similarly, PoSGU learns g spatial relative position dictionaries $\{\mathbf{P}_S^1, \mathbf{P}_S^2, \cdots, \mathbf{P}_S^g\} \in \mathbb{R}^{(2H-1)\times(2W-1)}$, where there is a total of $(2H-1)\times$

$(2W-1)$ relative position biases in each group. By indexing the position offset, we can construct g spatial relative position matrices $\{\mathbf{R}_S^1, \mathbf{R}_S^2, \cdots, \mathbf{R}_S^g\} \in \mathbb{R}^{HW \times HW}$. For each frame, the spatial refined token embeddings $\mathbf{Z}_S$ can be calculated as

$$\mathbf{Z}_S^{i,l} = \mathbf{R}_S^i \mathbf{V}_1^{i,l} \odot \mathbf{V}_2^{i,l}, \tag{4}$$

where $\mathbf{V}_1^{i,l}, \mathbf{V}_2^{i,l}, \mathbf{Z}_S^{i,l} \in \mathbb{R}^{HW \times \frac{C}{2g}}$, $i \in [1, 2, \cdots, g]$ and $l \in [1, 2, \cdots, T]$.

PoSTGU treats the 3D feature tensor as THW spatio-temporal tokens and captures their correlations from the spatio-temporal view. Specifically, by employing LRPE, PoSTGU also constructs the g spatio-temporal relative position matrices $\{\mathbf{R}_{ST}^1, \mathbf{R}_{ST}^2, \cdots, \mathbf{R}_{ST}^g\} \in \mathbb{R}^{THW \times THW}$ based on g learned spatio-temporal relative position dictionaries $\{\mathbf{P}_{ST}^1, \mathbf{P}_{ST}^2, \cdots, \mathbf{P}_{ST}^g\} \in \mathbb{R}^{(2T-1)\times(2H-1)\times(2W-1)}$. The final refined token embedding matrix $\mathbf{Z}_{ST}$ is hereby computed as

$$\mathbf{Z}_{ST}^i = \mathbf{R}_{ST}^i \mathbf{V}_1^i \odot \mathbf{V}_2^i, \tag{5}$$

where $\mathbf{V}_1^i, \mathbf{V}_2^i, \mathbf{Z}_{ST}^i \in \mathbb{R}^{THW \times \frac{C}{2g}}$, and $i \in [1, 2, \cdots, g]$. Figure 1(c) shows the pipeline.

Complexity Comparison. We list the parameters counting of gMLP's SGU and our positional units in Table 1. It can be found that the proposed PoTGU/-PoSGU/PoSTGU have much fewer parameters than SGU. For example, by setting the 3D window size as $\{T = 16, H = 7, W = 7\}$ and $g = 8$, PoTGU, PoSGU and PoSTGU have 248, 1,352 and 41,912 parameters compared to 615,440 of SGU. In the comparison with self-attention, since there is no pairwise computation in position units, our PosMLP-Video also show much lower FLOPs than video Transformers as demonstrated by the experiment.

Table 1 Comparison of parameters between gMLP's SGU and our position variants.

Module	Params.
SGU (gMLP)	$THW \times (THW + 1)$
PosTGU	$g \times (2T - 1)$
PosSGU	$g \times (2H - 1) \times (2W - 1)$
PosSTGU	$g \times (2T - 1) \times (2H - 1) \times (2W - 1)$

3.3 Video PosMLP Blocks

PoTGU/PoSGU/PoSTGU model the cross-token interaction. To further enhance the cross-channel interaction, we follow gMLP H. Liu et al. (2021) and add channel FC layer before and after the pos gating unit, resulting in three corresponding PosMLP modules. In practice, they have a similar network structure, as shown in Figure 2. Specifically, we first add a LayerNorm and a channel FC layer followed by GELU activation before the pos unit. The channel is expanded with a ratio r_e in this FC layer. Then another channel FC layer is put after the pos unit to reshape the channel size to C.

To achieve the modeling of spatial-temporal interactions among the tokens in a 3D view, we introduce three factorized spatio-temporal PosMLP blocks by combining PoTGU and PoSGU, including two cascaded blocks: **PoTGU→PoSGU** and **PoSGU→PoTGU**, and one paralleled block: **PoTGU+PoSGU**. As a comparison, we also present the joint spatio-temporal PosMLP block by using **PoSTGU**. Their architectural designs are shown in Figure 2. Particularly, the factorized spatio-temporal design shares a similar spirit with the existing works in video CNNs Hao et al. (2022); Qiu et al. (2017); Xie et al. (2018), Transformers Arnab et al. (2021); Bertasius et al. (2021) and MLPs Qiu et al. (2022); D.J. Zhang et al. (2022). As demonstrated in their experiments as well as ours, the factorized architectures can achieve a preferable balance between accuracy and efficiency.

3.4 PosMLP-Video Architecture

We adopt the hierarchical architecture design used by MorphMLP D.J. Zhang et al. (2022) as the basic network framework. Figure 3 illustrates the overall architecture. Particularly, the patch embedding block at the top of the model receives the input of a raw video and outputs token embeddings. The spatial downsampling layers between the four stages are to reduce the spatial resolution with a ratio of 2. Following Z. Wang et al. (2022), we utilize the spatial window partitioning strategy on stages 1-4 to produce multiple non-overlapping token embedding windows, on which the video PosMLP blocks act. The used window sizes are 14×14 for stages 1-3 and 7×7 for the last stage 4. Two obvious advantages of using window partitioning are: (1) small windows enhance the locality of modeling Z. Liu et al. (2021, 2022); and (2) parameters

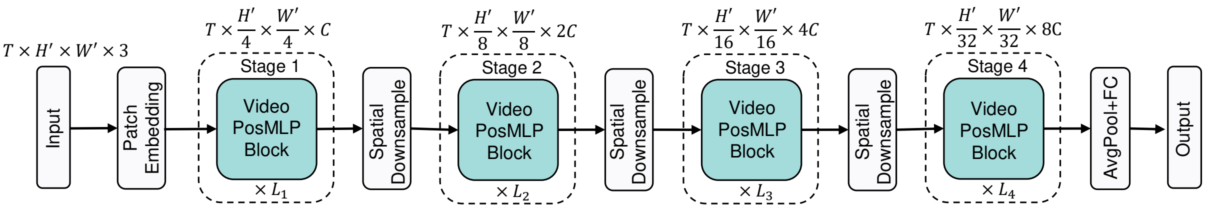

$T \times H' \times W' \times 3$
Input
Patch Embedding
$T \times \frac{H'}{4} \times \frac{W'}{4} \times C$
Stage 1
Video PosMLP Block
$\times L_1$
Spatial Downsample
$T \times \frac{H'}{8} \times \frac{W'}{8} \times 2C$
Stage 2
Video PosMLP Block
$\times L_2$
Spatial Downsample
$T \times \frac{H'}{16} \times \frac{W'}{16} \times 4C$
Stage 3
Video PosMLP Block
$\times L_3$
Spatial Downsample
$T \times \frac{H'}{32} \times \frac{W'}{32} \times 8C$
Stage 4
Video PosMLP Block
$\times L_4$
AvgPool+FC
Output

Fig. 3 Overall architecture of PosMLP-Video.

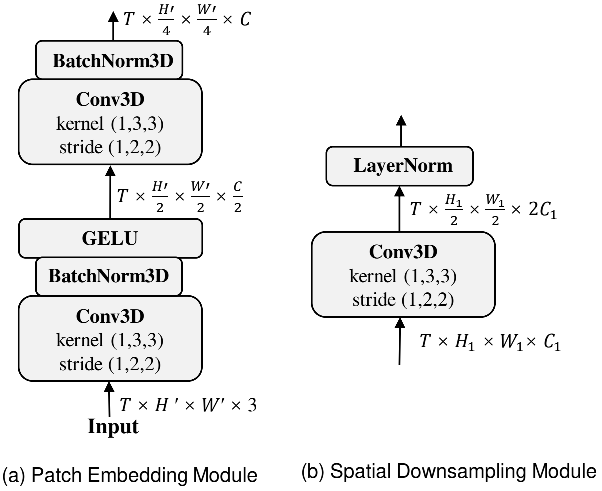

$T \times \frac{H'}{4} \times \frac{W'}{4} \times C$
BatchNorm3D
Conv3D
kernel (1,3,3)
stride (1,2,2)
$T \times \frac{H'}{2} \times \frac{W'}{2} \times \frac{C}{2}$
GELU
BatchNorm3D
Conv3D
kernel (1,3,3)
stride (1,2,2)
$T \times H' \times W' \times 3$
Input
LayerNorm
$T \times \frac{H_1}{2} \times \frac{W_1}{2} \times 2C_1$
Conv3D
kernel (1,3,3)
stride (1,2,2)
$T \times H_1 \times W_1 \times C_1$
(a) Patch Embedding Module
(b) Spatial Downsampling Module

Fig. 4 Architecture of patch embedding and spatial downsampling modules.

of a video PosMLP block are shared among windows, significantly reducing the model complexity.

We provide three PosMLP-Video variants, including PosMLP-Video-S (small), PosMLP-Video-B (base) and PosMLP-Video-L (large). Their architecture settings are listed in Table 2. Their differences lie in r_e and layer numbers of stages 1-4. It is worth noting that the temporal length T keeps unchanged throughout the network, which is different from MorphMLP and MLP-3D that reduce T to $\frac{T}{2}$ after patch embedding. We find this setting can result in much better performance.

We present the architecture details of the patch embedding and the spatial downsampling modules of PosMLP-Video. As shown in Figure 4(a), the patch embedding module consists of two standard convolution layers with the kernel (1,3,3) and stride (1,2,2). A batchnorm layer (BatchNorm3D) is added after each convolution. The activation GELU is inserted between the two convolution layers. The spatial downsampling module is implemented as a standard convolution layer with the kernel (1,3,3) and stride (1,2,2) followed by a LayerNorm, as shown in Figure 4(b).

Table 2 Architecture settings for Stage 1-4 of PosMLP-Video variants.

Models	r_e	**Layer numbers** $\{L_1, L_2, L_3, L_4\}$	**Layer channels** $\{C, 2C, 3C, 4C\}$
PosMLP-Video-S	2	$\{3, 4, 9, 3\}$	$\{72, 144, 288, 576\}$
PosMLP-Video-B	2	$\{4, 6, 15, 4\}$	
PosMLP-Video-L	4	$\{4, 6, 15, 4\}$	

Our PosMLP-Video can also benefit from the pretraining on large-scale image datasets like ImageNet1K. Since only the PoTGU and PoSTGU need to perform on the time axis, we can simply replace them with a residual connection to facilitate image modeling. When dealing with videos, the only parameters, i.e., the relative position biases, of PoTGU and PoSTGU will be randomly initialized.

4 Experiment

We examine our PosMLP-Video models on video classification tasks, for both coarse-grained and fine-grained video actions. Following prior art, top-1 and top-5 accuracies (%) are adopted to evaluate the performance. Parameters and FLOPs are also reported to show the model complexity.

4.1 Datasets

We use 5 standard video benchmark datasets, including Kinetics-400 (K400) Kay et al. (2017), Something-Something V1 (SSV1) and V2 (SSV2) Goyal et al. (2017), Diving48 Y. Li, Li, and Vasconcelos (2018) and EGTEA Gaze+ Y. Li, Liu, and Rehg (2018). **Kinetics-400** is a large-scale video dataset, containing ∼246k/20k training/validation videos for 400 human action classes. The actions in Kinetics-400 are relatively coarse and prefer spatial context

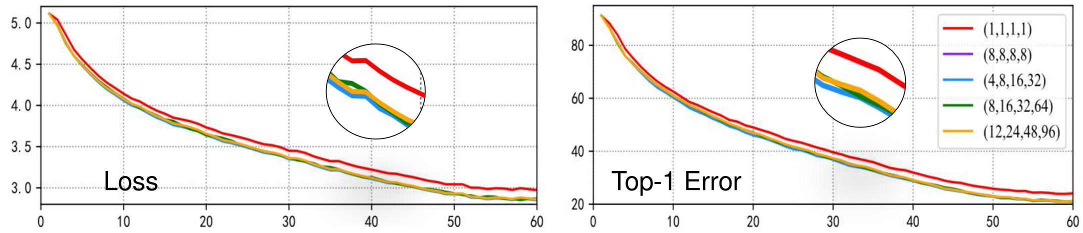


Fig. 5 Training loss and top-1 error curves of PosMLP-Video-S with different group numbers g on SSV1.

Table 3 Performance comparison with different video PosMLP blocks on SSV1 dataset. The results are obtained without pertaining except the last one. TGU is a temporal version of SGU. "IN-1K" means that the model is pretrained on ImageNet1K.

Video PosMLP Block	Params	GFLOPs	Top-1	Top-5
PoSGU	7.95M	25.08	6.25	20.64
PoTGU	7.65M	20.32	25.44	51.36
PoSTGU	17.19M	103.09	40.89	70.55
PoTGU→PoSGU	13.51M	40.49	44.11	74.70
PoSGU→PoTGU	13.51M	40.49	42.40	72.36
PoTGU+PoSGU	13.51M	40.49	**46.31**	**75.34**
SGU+TGU (gMLP)	13.83M	40.76	42.94	72.17
PoTGU+PoSGU (IN-1K)	13.51M	40.49	**52.24**	**78.96**

Table 4 Performance comparison with different group numbers g on SSV1 dataset. The used window size is the default setting. The results are obtained without pretraining.

g Stage 1-4	Params	GFLOPs	Top-1	Top-5
$(1,1,1,1)$	13.20M	40.49	43.90	73.19
$(8,8,8,8)$	13.24M	40.49	45.62	74.74
$(4,8,16,32)$	13.35M	40.49	44.87	74.53
$(8,16,32,64)$	13.51M	40.49	**46.31**	**75.34**
$(12,24,48,96)$	13.67M	40.49	45.96	75.15

to temporal context. **Something-Something V1** and **V2** cover 174 fine-grained human performing activities and require temporal modeling more. Particularly, V1 is the smaller version and has ∼86K/12K training/validation videos, while the larger V2 contains ∼169k/25k videos. **Diving48** is another fine-grained video dataset, consisting of ∼18k trimmed video clips of 48 unambiguous dive sequences. Here, the newly released dataset version (V2) is used. **EGTEA Gaze+** is a first-person video dataset covering 106 fine-grained daily action categories, where the split-1 that contains ∼8.3k/3.8k training/validation clips is selected for use.

4.2 Implementation Details

The implementation of PosMLP-Video variants is built upon the PySlowFast Fan et al. (2020) repository and mostly follows MViT Fan et al. (2021) and MorphMLP D.J. Zhang et al. (2022) for training and validation protocols. All experiments are run on servers with 4×3090 or 4×A100 GPUs.

Pretraining. As explained in Section 3.4, our PosMLP-Video variants can be pre-trained on the image dataset. The pertaining settings follow Z. Wang et al. (2022). After obtaining the pertained weights, we can easily use them to initialize most of the video model layers and blocks except for the relative position biases of PoTGU and PoSTGU. In the experiment, we adopt ImageNet1K for pertaining and observe comparable or even better performance than those pre-trained on the larger-scale ImageNet21K.

Training. For all the five datasets, each frame of a video is firstly resized to 256×320 and then cropped to 224×224 as model input. The temporal length T is set to 16/24. Particularly, for Kinetics-400, the dense sampling strategy is adopted to select T video frames, and the training configurations are set as follows: warm-up epoch 10, total epoch 60, batch size 8 per GPU, and base learning rate 2e-4 and weight decay 0.05 for AdamW optimizer. The random horizontal flip is also adopted. Stochastic depth rates are set to 0.05/0.1/0.2 for S/B/L. For other datasets, the sparse sampling strategy is used. Most of the training settings are the same with Kinetics-400, except warm-up epoch 5 and base learning rate 4e-4. Here, stochastic depth rates are set to 0.1/0.3/0.5 for S/B/L.

Inference. For Kinetics-400, we uniformly sample four clips from each test video with three crops Feichtenhofer, Fan, Malik, and He (2019). While, for other datasets, we only extract one clip with each having one or three crops. The form of "$A \times B \times C$" denotes A frames (T), B crops and C clips in the tables.

Table 5 Performance comparison with different window sizes (time length (T)×space (H, W)) on SSV1 dataset. The used g is $(8, 16, 32, 64)$ for Stage 1-4. The results are obtained without pretraining.

Window size Stage (S) 1-4	Params	GFLOPs	Top-1
8×14^2 (S1-3),8×7^2 (S4)	13.50M	20.38	42.33
16×7^2 (S1-4)	13.30M	36.64	44.21
16×14^2 (S1-S3),16×7^2 (S4)	13.51M	40.49	**46.31**
16×28^2 (S1), 16×14^2 (S2-3),16×7^2 (S4)	13.57M	46.86	46.29

Table 6 Performance comparison of PosMLP-Video-S on shuffling frame inference on K400 and SSV2 datasets.

Dataset	Shuffling	Top-1	Top-5
SSV2	✘	68.1	91.3
	✔	17.1 (-51.0)	44.2 (-47.1)
K400	✘	78.5	93.9
	✔	58.6 (-19.9)	81.0 (-12.9)

Table 7 Performance comparison of GQPE and LRPE using PosMLP-Video-S on SSV1 dataset. The results are obtained without pertaining.

RPE method	Params	GFLOPs	Top-1	Top-5
GQPE	13.19M	40.56	38.06	67.45
LRPE (used)	13.51M	40.49	**46.31**	**75.34**

4.3 Ablation Study

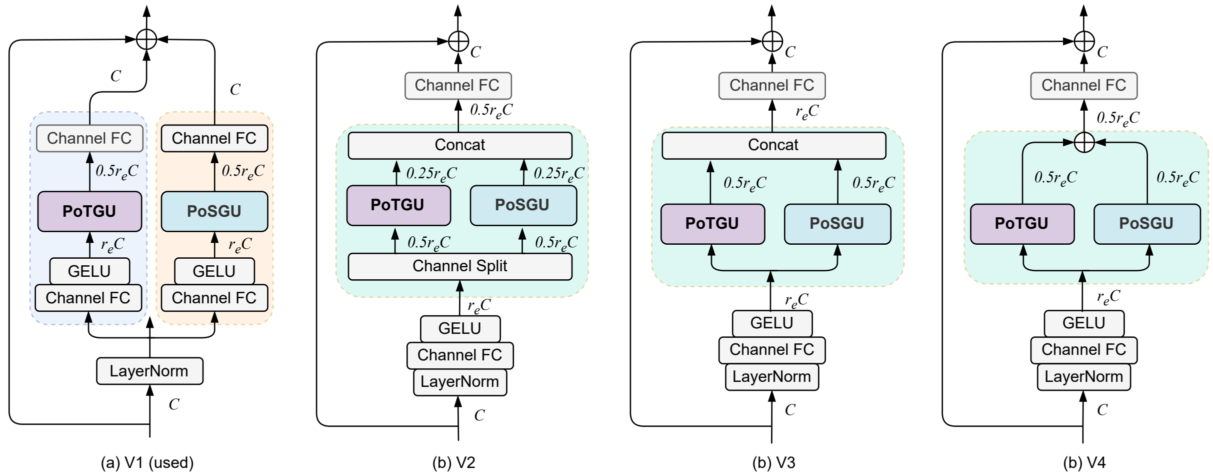
In the ablation study, we show the examinations of different hyperparameters and settings, including video PosMLP block variants, group numbers g, window sizes and RPE methods, using PosMLP-Video-S on the Something-Something V1 dataset. All the results are obtained with $16 \times 1 \times 1$ frames.

Video PosMLP block variants. Firstly, we compare the different PosMLP block variants, including both single pos-based versions and combined versions, with g=$(8, 16, 32, 64)$ in Stage 1-4 and the default window size setting. As shown in Table 3, the single temporal PoTGU can obtain higher top-1 accuracy than the single spatial PoSGU, while the spatio-temporal PoSTGU surpasses PoTGU and PoSGU. Since categorizing videos in SSV1 requires specific spatio-temporal relation modeling, this result trend is expected. Also, the three combined versions consistently outperform the single versions. Among them, the paralleled PoTGU+PoSGU achieves the best performance. In addition, we replace the position units in each block with gMLP's spatial SGU and temporal TGU (extended from SGU) and observe a significant top-1 accuracy drop (3.37%) and higher computation cost than ours. In the last line, we also provide the results of PoTGU+PoSGU with pre-training on ImageNet1K, improving the top-1 accuracy of w/o pre-training from 46.31% to 52.24%.

Different group number g. Secondly, we examine the impact of group number g. We can find in Table 4 that increasing g will monotonically raise the model size (parameters) while boosting performance significantly. This is mainly because that larger g leads to more relative position bias dictionaries and also enlarges the model capacity for diverse spatio-temporal relations. In Figure 5, We also show the training loss and top-1 error curves. It can be observed that multi-group (i.e., $g > 1$) settings have lower training losses and higher convergence speeds than single-group one (red line). However, the performance does not increase without bound. When increasing g to $(12, 24, 48, 96)$, the performance is relatively lower than that with $(8, 16, 32, 64)$. Finally, considering the trade-off between model size and accuracy, we set g to $(8, 16, 32, 64)$ for stages 1-4.

Different window size. Thirdly, we compare the performance of different window sizes. Window partitioning is used to split the whole spatio-temporal video clip into several non-overlapped parts, whose effect has been demonstrated by Swin Transformers Z. Liu et al. (2021, 2022). As shown in Table 5, we find that (1) although temporal partitioning can greatly reduce the computation cost (FLOPs), e.g., $T = 8$ compared with $T = 16$, it leads to noticeable performance degradation (42.33% vs. 46.31%), and (2) larger spatial window size results in more parameters and FLOPs while performance does not show monotonic increasing. In other words, the large temporal receptive field is more important for action recognition on SSV1. By considering the complexity-accuracy trade-off, we select $16 \times 14 \times 14$ for Stage 1-3 and $16 \times 7 \times 7$ for Stage 4 as the ultimate window size settings.

Frame shuffling. Finally, we verify the order sensitivity of our PosMLP-Video model by testing on randomly shuffled frames. It should be noted that the model training does not use random frame shuffling. As observed from Table 6, there are significant



(a) V1 (used)

(b) V2

(b) V3

(b) V4

Fig. 6 PoTGU+PoSGU versions. "V1" is the used version of PosMLP-Video. "V2" adopts the channel splitting before inputting into the pos units and then concatenates the outputs of PoTGU and PoSGU along the channel dimension. "V3" separately inputs the feature into PoTGU and PoSGU and then concatenates their outputs along the channel dimension. In contrast to V3, "V4" elementwisely adds the outputs of PoTGU and PoSGU.

Table 8 Performance comparison of different PoTGU+PoSGU versions using PosMLP-Video-S on SSV1 dataset. The results are obtained without pertaining.

Versions	r_e	Params	GFLOPs	Top-1	Top-5
V1 (used)	2	13.51M	40.49	**46.31**	75.34
V2	4	13.49M	40.35	45.92	**75.43**
V3	2	9.80M	30.46	43.98	73.34
V4	2	7.96M	25.52	42.52	72.31

performance drops with frame shuffling on both two datasets, e.g., -51.0 (Top-1) on SSV2 and -19.9 (Top-1) on K400. This demonstrates that the proposed positional module can successfully capture temporal order information. The larger performance drop on SSV2 than that on K400, on the other hand, indicates that videos of SSV2 are more sensitive to temporal order than videos of K400.

LRPE vs. GQPE. In addition to LRPE, we also test the performance of another RPE method called GQPE. Table 7 compares the performance of both methods, and it was observed that LRPE significantly outperforms GQPE. Despite having fewer parameters, with the model complexity of GQPE being $O(1)$ compared to $O(N)$ of LRPE, the model capacity of GQPE is probably constrained. This is due to the fact that the RPE-based relation score in GQPE is determined solely by the relative position and is not learnable.

PoTGU+PoSGU variants. In Figure 6, we present four PoTGU+PoSGU block versions (V1-4). These versions incorporate various feature operations such as channel splitting (V2), channel concatenation (V2, V3), and elementwise addition (V4). By adjusting the value of r_e, the model size can be conveniently controlled. The experimental results, as presented in Table 8, suggest that a larger model size (i.e., more parameters) generally leads to better recognition performance. Interestingly, the use of channel splitting and concatenation (V2) does not improve the Top-1 accuracy, despite having similar model sizes when compared to V1. Conversely, although elementwise addition (V4) reduced the channel length significantly, it also led to a degradation in recognition accuracy, as observed in V3 and V4.

4.4 Comparison with State-of-the-Art

We compare PosMLP-Video networks with various state-of-the-art networks, including video CNNs, Transformers and MLPs, on many video recognition tasks. All the competing methods adopt RGB frames as input and are pre-trained on ImageNet1K (IN-1K), ImageNet21K (IN-21K), Kinetics-400 (K400), Kinetics-600 (K600) or None.

Something-Something V1&V2. The two datasets share the same human-performing action categories and only differ in scale. Their video actions focus on more temporal relationships, for example, "Putting something ...", "Lifting something ..." and "Pretending to ...". Tables 9 and 10 show the performance comparison on V1 and V2, respectively. On the smaller SSV1, our PosMLP-Video-S achieves a higher

Table 9 Comparison of performance on Something-Something V1 dataset.

Method	Pretrain	Params	Fr.×Cr.×Cl.	GFLOPs	Top-1	Top-5
CNNs						
TSM Lin et al. (2019)	IN-1K	23.9M	16×3×2	197.4	48.4	78.1
SmallBig X. Li, Wang, Zhou, and Qiao (2020)		—	16×3×2	—	50.0	79.8
STM Jiang, Wang, Gan, Wu, and Yan (2019)		24.0M	16×3×10	999	50.7	80.4
TEINet Z. Liu et al. (2020)		30.4M	16×3×10	1980	51.0	—
MSNet Kwon, Kim, Kwak, and Cho (2020)		24.6M	16×1×1	67	52.1	**82.3**
TEA Y. Li et al. (2020)		—	16×3×10	2100	52.3	81.9
CT-NET K. Li, Li, Wang, Wang, and Qiao (2021)		—	16×3×2	447	53.4	81.7
TDN L. Wang, Tong, Ji, and Wu (2021)		26.1M	16×1×1	72.0	53.9	82.1
GC-TDN Hao et al. (2022)		27.4M	16×1×1	73.4	**55.0**	**82.3**
MLPs						
MLP-3D-S Qiu et al. (2022)	IN-1K	74.1M	64×3×1	324	55.2	83.2
MLP-3D-B Qiu et al. (2022)		88.3M	64×3×1	549	56.2	83.5
MLP-3D-L Qiu et al. (2022)		149.4M	64×3×1	1008	56.5	83.5
MorphMLP-S D.J. Zhang et al. (2022)		46.9M	16×3×1	201	53.9	81.3
MorphMLP-B D.J. Zhang et al. (2022)		67.6M	16×3×1	294	55.5	82.4
MorphMLP-B D.J. Zhang et al. (2022)		68.5M	32×3×1	591	57.4	84.5
PosMLP-Video-S	IN-1K	13.5M	16×3×1	122	55.6	82.1
PosMLP-Video-B		19.0M	16×3×1	177	58.2	**84.6**
PosMLP-Video-L		35.4M	16×3×1	338	**59.0**	84.3

Top-1 accuracy of 55.6% compared to all the competing video CNNs (48.4%-55.0%). In comparison with other video MLPs, such as MorphMLP and MLP-3D, PosMLP-Video variants (S, B, L) consistently outperform them and spend much lower computations. For example, PosMLP-Video-L achieves the highest Top-1 accuracy of 59.0% with only 35.4M parameters/338G FLOPs, which significantly surpasses MLP-3D-L's 56.5% with 149.4M parameters/1008G FLOPs and MorphMLP-B's 57.4% with 68.5M parameters/591G FLOPs. Particularly, as our PosMLP-Video has a similar backbone architecture with MorphMLP, the notable performance improvements further demonstrate the superiority of the proposed positional regimes for spatio-temporal relation modeling.

On the larger SSV2, PosMLP-Video variants consistently outperform video CNNs, Transformers and MLPs. In particular, PosMLP-Video-L pre-trained on IN-1K achieves the highest Top-1 accuracy of 70.3% with $16 \times 3 \times 1$ frames input, which even outstrips the video Transformers such as X-ViT, RViT-XL, MFormer-L, MViTv2-S and Swin-B that are pratrained on the larger-scale datasets (e.g., IN-21K, K400 and K600) and use more frames (e.g., $32 \times 3 \times 1$ and $64 \times 3 \times 1/3$) as input. More importantly, PosMLP-Video-L has only 35.4M parameters, which is 40% of Swin-B. The FLOPs of PosMLP-Video-L is 338G, which is only 35% of Swin-B. Moreover, compared to the video MLPs, i.e., MLP-3D and MorphMLP, our PosMLP-Video, regardless of network versions, consistently outperforms them with large performance improvements (0.2%-2.5%) while requiring much less computational costs (about 18%-51% parameters and 33%-60% FLOPs).

Kinetics-400. K400 is a large-scale video recognition dataset, whose video categories depend not so much on temporal relations. We list the performance comparison with the SOTA methods in Table 11. Similar to observations as on SSV1 and SSV2, PosMLP-Video models perform consistently better than the competing video CNNs such as GC-TDN, SlowFast101+NL and X3D-XXL. By comparing with Transformer-based models, our PosMLP-Video-L pre-trained on ImageNet-1K achieves the best Top-1 accuracy of 82.1% with $24 \times 3 \times 4$ frames. Compared to MTV-B which obtains the second best 81.8% Top-1 accuracy pre-trained on ImageNet-21K and has 310M parameters and 4790 GFLOPs, PosMLP-Video-L only requires 35.5M parameters and 2037 GFLOPs. In other words, the model size of our PosMLP-Video-L is as small as 11% of MTV-B's, and the computational cost is only its 43%. Also, compared to MorphMLP and MLP-3D, the proposed PosMLP-Video obtains higher recognition performance and requires less computational burden.

Table 10 Comparison of performance on Something-Something V2 dataset.

Method	Pretrain	Params	Fr.×Cr.×Cl.	GFLOPs	Top-1	Top-5
CNNs						
TSM Lin et al. (2019)	IN-1K	23.9M	16×1×2	131.6	63.1	88.2
SlowFast Feichtenhofer et al. (2019)		53.3M	40×3×2	636	63.1	87.6
SmallBig X. Li et al. (2020)		—	16×3×2	—	63.8	88.9
STM Jiang et al. (2019)		24.0M	16×3×10	999	64.2	89.8
TEINet Z. Liu et al. (2020)		30.4M	16×1×10	990	64.7	—
MSNet Kwon et al. (2020)		24.6M	16×1×1	67	64.7	89.4
TEA Y. Li et al. (2020)		—	16×3×10	2100	65.1	89.9
TDN L. Wang et al. (2021)		26.1M	16×1×1	72.0	65.3	89.5
CT-NET K. Li et al. (2021)		—	16×3×2	447	**65.9**	**90.1**
GC-TDN Hao et al. (2022)		27.4M	16×1×1	73.4	**65.9**	90.0
Transformers						
TimeSformer-HR Bertasius et al. (2021)	IN-21K	121.4M	16×3×1	5109	62.5	—
ViViT-L/16×2 Arnab et al. (2021)	IN-21K	352.1M	16×3×4	11892	65.4	89.8
DVT J. Wang and Torresani (2022)	IN-1K	73.9M	16×3×1	385	66.7	90.8
X-ViT Bulat et al. (2021)	K600	92.0M	16×3×1	850	67.2	90.8
MM-ViT J. Chen and Ho (2022)	IN-21K	158.1M	16×3×1	4530	67.4	90.6
MTV-B Yan et al. (2022)	IN-21K	310M	32×3×1	963	67.6	90.1
MViT-B Fan et al. (2021)	K400	36.6M	64×3×1	1365	67.7	90.9
RViT-XL,64×3 Yang et al. (2022)	K400	107.7M	64×3×3	3990	67.9	91.2
ORViT MF Herzig et al. (2022)	IN-21K+K400	148M	16×3×1	405	67.9	90.5
MFormer-L Patrick et al. (2021)	IN-21K+K400	—	32×3×1	3555	68.1	91.2
MViTv2-S,16×4 Y. Li et al. (2022)	K400	34.4M	16×3×1	194	68.2	91.4
Swin-B Z. Liu et al. (2022)	K400	88.8M	16×3×1	963	**69.6**	**92.7**
MLPs						
MLP-3D-S Qiu et al. (2022)	IN-1K	74.1M	64×3×1	324	67.2	91.3
MLP-3D-M Qiu et al. (2022)		88.3M	64×3×1	549	68.0	91.7
MLP-3D-L Qiu et al. (2022)		149.4M	64×3×1	1008	68.5	92.0
MorphMLP-S D.J. Zhang et al. (2022)		46.9M	16×3×1	201	67.1	90.9
MorphMLP-B D.J. Zhang et al. (2022)		67.6M	16×3×1	294	67.6	91.3
MorphMLP-B D.J. Zhang et al. (2022)		68.5M	32×3×1	591	70.1	**92.8**
PosMLP-Video-S	IN-1K	13.5M	16×3×1	122	68.1	91.3
PosMLP-Video-B		19.0M	16×3×1	177	70.1	92.5
PosMLP-Video-L		35.4M	16×3×1	338	**70.3**	92.3

Diving48. It contains unambiguous dive sequences. Dives entail several stages and necessitate long-range temporal modeling. Table 12 provides a performance comparison with other competing methods. The results show that our proposed PosMLP-Video-L model with 16 frames attains the highest Top-1 accuracy of 88.9% among the competing methods, which include both video CNNs and Transformers. Particularly, the proposed model outperforms the CNN-based TFCNet's accuracy of 88.3% by 0.6% and the Transformer-based ORViT TimeSformer's accuracy of 88.0% by 0.9%.

EGTEA Gaze+. This dataset consists of videos showing cooking activities that involve intricate spatio-temporal hand-object and object-object interactions. Table 13 presents a comparison of the results obtained from different methods. Our PosMLP-Video-L achieves the highest Top-1 accuracy of 72.5%. This remarkable result significantly surpasses the previous methods by a considerable margin (6.0%-9.9%), which provides compelling evidence for the superior ability of our model in spatio-temporal modeling.

4.5 Visualization

Figure 7 displays the pairwise token relation matrix learned by PoSGU and PoTGU of PosMLP-Video-S on the SSV2 dataset. We select the first and last layers of PosMLP-Video-S and showcase the learned spatial and temporal token-to-token relations of the two channel groups. Firstly, it can be found that different layers and groups learn various relationship types,

Table 11 Comparison of performance on Kinetics-400 dataset.

Method	**Pretrain**	**Params**	**Fr.×Cr.×Cl.**	**GFLOPs**	**Top-1**	**Top-5**
CNNs						
TSM Lin et al. (2019)	IN-1K	24.3M	16×1×10	660	74.7	91.4
NL-I3D X. Wang et al. (2018)	IN-1K	35.3M	32×1×10	2820	74.9	91.6
TEINet Z. Liu et al. (2020)	IN-1K	30.8M	16×3×10	1980	76.2	92.5
SmallBig-R101 X. Li et al. (2020)	IN-1K	—	32×3×4	5016	77.4	93.3
TDN-R101 L. Wang et al. (2021)	IN-1K	—	16×3×10	3960	78.5	93.9
CT-NET-R101 K. Li et al. (2021)	IN-1K	—	16×3×4	1746	78.8	93.7
GC-TDN-R50 Hao et al. (2022)	IN-1K	27.4M	16×3×10	2202	78.8	93.8
SlowFast101+NL Feichtenhofer et al. (2019)	—	59.9M	80×3×10	7020	79.8	93.9
X3D-XXL Feichtenhofer (2020)	—	20.3M	−×3×10	5820	**80.4**	**94.6**
Transformers						
TokShift H. Zhang et al. (2021)	IN-21K	85.9M	16×3×10	8085	78.2	93.8
SACS-H H. Zhang, Cheng, Hao, and Ngo (2022)	IN-21K	40M	32×3×5	5190	79.7	94.1
Mformer-L Patrick et al. (2021)	IN-21K	—	32×3×10	35553	80.2	94.8
ViViT-L/16×2 Arnab et al. (2021)	IN-21K	310.8M	16×3×4	17357	80.6	94.7
MViT-B,16×4 Fan et al. (2021)	—	36.6M	16×1×5	353	78.4	93.5
MViT-B,32×3 Fan et al. (2021)	—	36.6M	32×1×5	850	80.2	94.4
Swin-S Z. Liu et al. (2022)	IN-1K	49.8M	32×3×4	1992	80.6	94.5
Swin-B Z. Liu et al. (2022)	IN-1K	88.1M	32×3×4	3384	80.6	94.6
TimeSformer-L Bertasius et al. (2021)	IN-21K	121.4M	96×3×1	7140	80.7	94.7
X-ViT Bulat et al. (2021)	IN-21K	92.0M	16×3×2	1700	80.7	94.7
DVT J. Wang and Torresani (2022)	IN-1K	73.9M	16×1×5	640	80.8	**95.0**
MViTv2-S,16×4 Y. Li et al. (2022)	—	34.5M	16×1×4	320	81.0	94.6
RViT-XL,32×3×1 Yang et al. (2022)	IN-21K	107.7M	32×3×3	2010	80.3	94.4
RViT-XL,64×3×1 Yang et al. (2022)	IN-21K	107.7M	64×3×3	11900	81.5	**95.0**
MTV-B Yan et al. (2022)	IN-21K	310M	32×3×4	4790	**81.8**	**95.0**
MLPs						
MorphMLP-S D.J. Zhang et al. (2022)	IN-1K	47.0M	16×1×4	268	78.7	93.8
MorphMLP-B D.J. Zhang et al. (2022)		67.8M	16×1×4	392	79.5	94.4
MorphMLP-B D.J. Zhang et al. (2022)		68.5M	32×1×4	788	80.8	94.9
MLP-3D-S Qiu et al. (2022)		68.5M	64×3×4	1224	80.2	93.8
MLP-3D-M Qiu et al. (2022)		80.5M	64×3×4	2040	81.0	94.9
MLP-3D-L Qiu et al. (2022)		135.6M	64×3×4	3696	81.4	95.2
PosMLP-Video-S	IN-1K	13.6M	16×1×4	162	78.5	93.9
PosMLP-Video-B		19.1M	16×1×4	236	80.3	94.6
PosMLP-Video-L		35.4M	16×1×4	450	81.2	94.7
PosMLP-Video-L		35.4M	16×3×4	1350	81.6	94.9
PosMLP-Video-L		35.5M	24×1×4	679	81.7	95.2
PosMLP-Video-L		35.5M	24×3×4	2037	**82.1**	**95.3**

indicating that the channel grouping mechanism successfully enriches the relative position relationship types. Secondly, by comparing the relation pattern differences between the first and last layers, we observe that high relation scores mainly exist in local spatial and temporal neighborhoods in the first layer, while they spread all over the region in the last layer. This can be attributed to the fact that there is no cross-token interaction in the first layer, whereas both spatial and temporal tokens have been fully fused in the last layer.

In Figure 8, we present the heatmaps obtained from visualizing the class activation maps of different video PosMLP blocks using the Grad-CAM Selvaraju et al. (2017) technique. We select action categories with varying moving directions, such as "*Moving something closer to something*", "*Pushing something from right to left*", and "*Turning something upside down*", which involve short-term and long-term temporal interactions between objects. Our observations from the figure reveal that the paralleled PoTGU+PoSGU focuses more on the crucial areas

Table 12 Comparison of performance on Diving48 dataset.

Method	**Fr.×Cr.×Cl.**	**Top-1**
CNNs		
SlowFast-R101,16x8 Feichtenhofer et al. (2019) from Bertasius et al. (2021)	(64+16)×3×1	77.6
TQN C. Zhang, Gupta, and Zisserman (2021)	all frames	81.8
GC-TDN Hao et al. (2022)	16×1×1	87.6
TFCNet S. Zhang (2022)	32×3×1	88.3
Transformers		
TimeSformer-L Bertasius et al. (2021)	96×3×1	81.0
VIMPAC Tan, Lei, Wolf, and Bansal (2021)	−×3×10	85.5
ORViT TimeSformer Herzig et al. (2022)	32×3×1	88.0
MLPs		
PosMLP-Video-L	16×1×1	**88.9**

Table 13 Comparison of performance on EGTEA Gaze+ dataset.

Method	**Fr.×Cr.×Cl.**	**Top-1**
CNNs		
SAP X. Wang, Wu, Zhu, and Yang (2020)	64×1×1	64.1
GST-R50 Luo and Yuille (2019)	8×1×1	64.4
GC-TSM Hao et al. (2022)	8×1×1	66.5
Transformers		
ViT (Video) Dosovitskiy et al. (2020) from H. Zhang et al. (2021)	8×1×1	62.6
TokShift (HR) H. Zhang et al. (2021)	8×1×1	65.8
LAPS (H) H. Zhang et al. (2021)	32×1×1	66.1
MLPs		
PosMLP-Video-L	16×1×1	**72.5**

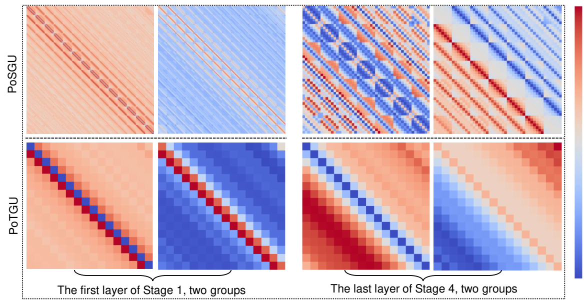


Fig. 7 Visualization of the pairwise token relation matrix learned by PoSGU and PoTGU of PosMLP-Video-S on SSV2 dataset.

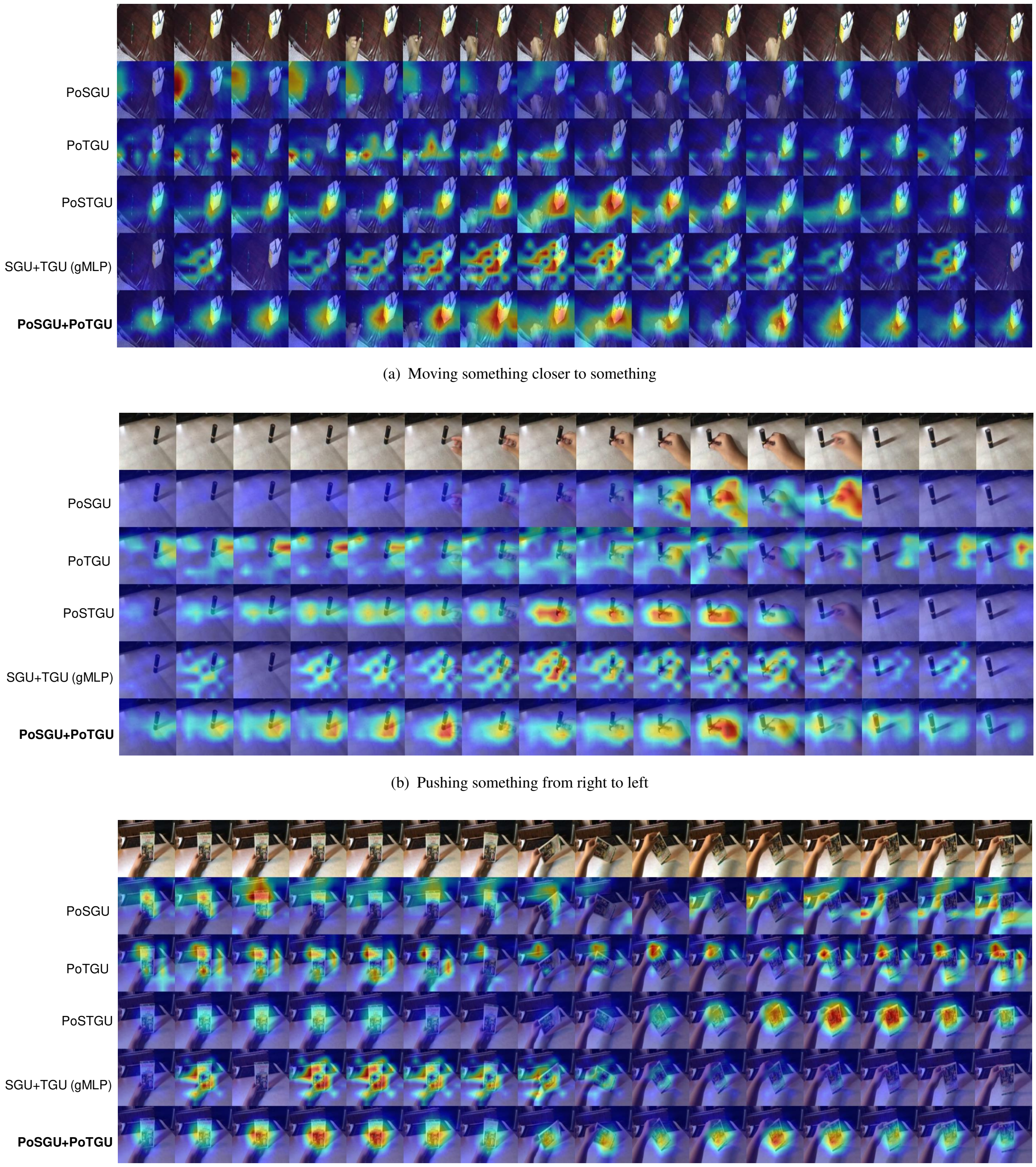
of video frames when compared to both the single pos blocks (PoTGU/PoSGU/PoSTGU) and the similar combined SGU+TGU (gMLP).

5 Conclusion

We have presented a novel MLP-like architecture, PosMLP-Video, for efficient and effective video recognition. Our approach leverages the relative position encoding as a key component, leading to improved pairwise token relations modeling. We introduced a family of positional spatial and temporal gating units (PoTGU, PoSGU, and PoSTGU) that are both more parameters- and FLOPs-efficient than self-attention and token-mixing layers. These units are integrated into spatio-temporal factorized network blocks to promote spatial and temporal modeling. Experimental results on video recognition tasks demonstrate that PosMLP-Video outperforms other competing video models. Furthermore, we show that PosMLP-Video is highly sensitive to temporal order, as we observed significant performance drops with randomly shuffled frames. Overall, our results highlight the efficacy of the proposed PosMLP-Video and its potential for advancing the field of video recognition.

References

Arnab, A., Dehghani, M., Heigold, G., Sun, C., Lučić, M., Schmid, C. (2021). Vivit: A video vision transformer. *Proceedings of the ieee/cvf international conference on computer vision* (pp.



(a) Moving something closer to something

(b) Pushing something from right to left

(c) Turning something upside down

Fig. 8 Visualization examples of class activation maps on SSV1 dataset. The first row shows the original 16 frames.

6836–6846).

Bertasius, G., Wang, H., Torresani, L. (2021). Is space-time attention all you need for video understanding? *Icml* (Vol. 2, p. 4).

Bulat, A., Perez Rua, J.M., Sudhakaran, S., Martinez, B., Tzimiropoulos, G. (2021). Space-time mixing attention for video transformer. *Advances in neural information processing systems* (pp. 19594–19607).

Carreira, J., & Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. *proceedings of the ieee conference on computer vision and pattern recognition* (pp. 6299–6308).

Chen, J., & Ho, C.M. (2022). Mm-vit: Multi-modal video transformer for compressed video action recognition. *Proceedings of the ieee/cvf winter conference on applications of computer vision* (pp. 1910–1921).

Chen, S., Xie, E., Ge, C., Liang, D., Luo, P. (2021). Cyclemlp: A mlp-like architecture for dense prediction. *arXiv preprint arXiv:2107.10224*, ,

Cordonnier, J.-B., Loukas, A., Jaggi, M. (2019). On the relationship between self-attention and convolutional layers. *arXiv preprint arXiv:1911.03584*, ,

Dong, X., Bao, J., Chen, D., Zhang, W., Yu, N., Yuan, L., ... Guo, B. (2022). Cswin transformer: A general vision transformer backbone with cross-shaped windows. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 12124–12134).

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., ... others (2020). An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, ,

d'Ascoli, S., Touvron, H., Leavitt, M.L., Morcos, A.S., Biroli, G., Sagun, L. (2021). Convit: Improving vision transformers with soft convolutional inductive biases. *International conference on machine learning* (pp. 2286–2296).

Fan, H., Li, Y., Xiong, B., Lo, W.-Y., Feichtenhofer, C. (2020). *Pyslowfast*. https://github.com/facebookresearch/slowfast.

Fan, H., Xiong, B., Mangalam, K., Li, Y., Yan, Z., Malik, J., Feichtenhofer, C. (2021). Multiscale vision transformers. *Proceedings of the ieee/cvf international conference on computer vision* (pp. 6824–6835).

Feichtenhofer, C. (2020). X3d: Expanding architectures for efficient video recognition. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 203–213).

Feichtenhofer, C., Fan, H., Malik, J., He, K. (2019). Slowfast networks for video recognition. *Proceedings of the ieee/cvf international conference on computer vision* (pp. 6202–6211).

Goyal, R., Ebrahimi Kahou, S., Michalski, V., Materzynska, J., Westphal, S., Kim, H., ... others (2017). The" something something" video database for learning and evaluating visual common sense. *Proceedings of the ieee international conference on computer vision* (pp. 5842–5850).

Guo, J., Tang, Y., Han, K., Chen, X., Wu, H., Xu, C., ... Wang, Y. (2022). Hire-mlp: Vision mlp via hierarchical rearrangement. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 826–836).

Hao, Y., Zhang, H., Ngo, C.-W., He, X. (2022). Group contextualization for video recognition. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 928–938).

He, K., Gkioxari, G., Dollár, P., Girshick, R. (2017). Mask r-cnn. *Proceedings of the ieee international conference on computer vision* (pp. 2961–2969).

He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 770–778).

Herzig, R., Ben-Avraham, E., Mangalam, K., Bar, A., Chechik, G., Rohrbach, A., … Globerson, A. (2022). Object-region video transformers. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 3148–3159).

Hou, Q., Jiang, Z., Yuan, L., Cheng, M.-M., Yan, S., Feng, J. (2022). Vision permutator: A permutable mlp-like architecture for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(1), 1328–1334,

Hu, J., Shen, L., Sun, G. (2018). Squeeze-and-excitation networks. *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 7132–7141).

Huang, Z., Ben, Y., Luo, G., Cheng, P., Yu, G., Fu, B. (2021). Shuffle transformer: Rethinking spatial shuffle for vision transformer. *arXiv preprint arXiv:2106.03650*, ,

Islam, M.A., Kowal, M., Jia, S., Derpanis, K.G., Bruce, N.D. (2021a). Global pooling, more than meets the eye: Position information is encoded channel-wise in cnns. *Proceedings of the ieee/cvf international conference on computer vision* (pp. 793–801).

Islam, M.A., Kowal, M., Jia, S., Derpanis, K.G., Bruce, N.D. (2021b). Position, padding and predictions: A deeper look at position information in cnns. *arXiv preprint arXiv:2101.12322*, ,

Jiang, B., Wang, M., Gan, W., Wu, W., Yan, J. (2019). Stm: Spatiotemporal and motion encoding for action recognition. *Proceedings of the ieee/cvf international conference on computer vision* (pp. 2000–2009).

Kay, W., Carreira, J., Simonyan, K., Zhang, B., Hillier, C., Vijayanarasimhan, S., … others (2017). The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, ,

Kayhan, O.S., & Gemert, J.C.v. (2020). On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 14274–14285).

Krizhevsky, A., Sutskever, I., Hinton, G.E. (2017). Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84–90,

Kwon, H., Kim, M., Kwak, S., Cho, M. (2020). Motionsqueeze: Neural motion feature learning for video understanding. *Computer vision–eccv 2020: 16th european conference, glasgow, uk, august 23–28, 2020, proceedings, part xvi 16* (pp. 345–362).

Li, K., Li, X., Wang, Y., Wang, J., Qiao, Y. (2021). Ct-net: Channel tensorization network for video classification. *arXiv preprint arXiv:2106.01603*, ,

Li, X., Wang, Y., Zhou, Z., Qiao, Y. (2020). Smallbignet: Integrating core and contextual views for video classification. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 1092–1101).

Li, Y., Ji, B., Shi, X., Zhang, J., Kang, B., Wang, L. (2020). Tea: Temporal excitation and aggregation for action recognition. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 909–918).

Li, Y., Li, Y., Vasconcelos, N. (2018). Resound: Towards action recognition without representation bias. *Proceedings of the european conference on computer vision (eccv)* (pp. 513–528).

Li, Y., Liu, M., Rehg, J.M. (2018). In the eye of beholder: Joint learning of gaze and actions in first person video. *Proceedings of the european conference on computer vision (eccv)* (pp. 619–635).

Li, Y., Wu, C.-Y., Fan, H., Mangalam, K., Xiong, B., Malik, J., Feichtenhofer, C. (2022). Mvitv2: Improved multiscale vision transformers for classification and detection. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 4804–4814).

Lian, D., Yu, Z., Sun, X., Gao, S. (2021). As-mlp: An axial shifted mlp architecture for vision. *arXiv preprint arXiv:2107.08391*, ,

Lin, J., Gan, C., Han, S. (2019). Tsm: Temporal shift module for efficient video understanding. *Proceedings of the ieee/cvf international conference on computer vision* (pp. 7083–7093).

Liu, H., Dai, Z., So, D., Le, Q.V. (2021). Pay attention to mlps. *Advances in Neural Information Processing Systems*, *34*, 9204–9215,

Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., ... Guo, B. (2021). Swin transformer: Hierarchical vision transformer using shifted windows. *Proceedings of the ieee/cvf international conference on computer vision* (pp. 10012–10022).

Liu, Z., Luo, D., Wang, Y., Wang, L., Tai, Y., Wang, C., ... Lu, T. (2020). Teinet: Towards an efficient architecture for video recognition. *Proceedings of the aaai conference on artificial intelligence* (Vol. 34, pp. 11669–11676).

Liu, Z., Ning, J., Cao, Y., Wei, Y., Zhang, Z., Lin, S., Hu, H. (2022). Video swin transformer. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 3202–3211).

Luo, C., & Yuille, A.L. (2019). Grouped spatial-temporal aggregation for efficient action recognition. *Proceedings of the ieee/cvf international conference on computer vision* (pp. 5512–5521).

Patrick, M., Campbell, D., Asano, Y., Misra, I., Metze, F., Feichtenhofer, C., ... Henriques, J.F. (2021). Keeping your eye on the ball: Trajectory attention in video transformers. *Advances in neural information processing systems*, *34*, 12493–12506,

Qiu, Z., Yao, T., Mei, T. (2017). Learning spatio-temporal representation with pseudo-3d residual networks. *proceedings of the ieee international conference on computer vision* (pp. 5533–5541).

Qiu, Z., Yao, T., Ngo, C.-W., Mei, T. (2022). Mlp-3d: A mlp-like 3d architecture with grouped time mixing. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 3062–3072).

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., ... Liu, P.J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *The Journal of Machine Learning Research*, *21*(1), 5485–5551,

Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. *Proceedings of the ieee international conference on computer vision* (pp. 618–626).

Shaw, P., Uszkoreit, J., Vaswani, A. (2018). Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, ,

Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. *Proceedings of the aaai conference on artificial intelligence* (Vol. 31).

Tan, H., Lei, J., Wolf, T., Bansal, M. (2021). Vimpac: Video pre-training via masked token prediction and contrastive learning. *arXiv preprint arXiv:2106.11250*, ,

Tolstikhin, I.O., Houlsby, N., Kolesnikov, A., Beyer, L., Zhai, X., Unterthiner, T., ... others (2021). Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, *34*, 24261–24272,

Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H. (2021). Training data-efficient image transformers & distillation through attention. *International conference on machine learning* (pp. 10347–10357).

Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M. (2015). Learning spatiotemporal features with 3d convolutional networks. *Proceedings of the ieee international conference on computer vision* (pp. 4489–4497).

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 6450–6459).

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., ... Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30*, ,

Wang, J., & Torresani, L. (2022). Deformable video transformer. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 14053–14062).

Wang, L., Tong, Z., Ji, B., Wu, G. (2021). Tdn: Temporal difference networks for efficient action recognition. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 1895–1904).

Wang, W., Xie, E., Li, X., Fan, D.-P., Song, K., Liang, D., ... Shao, L. (2021). Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *Proceedings of the ieee/cvf international conference on computer vision* (pp. 568–578).

Wang, X., Girshick, R., Gupta, A., He, K. (2018). Non-local neural networks. *Proceedings of the ieee conference on computer vision and pattern recognition* (pp. 7794–7803).

Wang, X., Wu, Y., Zhu, L., Yang, Y. (2020). Symbiotic attention with privileged information for egocentric action recognition. *Proceedings of the aaai conference on artificial intelligence* (Vol. 34, pp. 12249–12256).

Wang, Z., Hao, Y., Gao, X., Zhang, H., Wang, S., Mu, T., He, X. (2022). Parameterization of cross-token relations with relative positional encoding for vision mlp. *Proceedings of the 30th acm international conference on multimedia* (pp. 6288–6299).

Wu, B., Xu, C., Dai, X., Wan, A., Zhang, P., Yan, Z., ... Vajda, P. (2020). Visual transformers: Token-based image representation and processing for computer vision. *arXiv preprint arXiv:2006.03677*, ,

Wu, C.-Y., Li, Y., Mangalam, K., Fan, H., Xiong, B., Malik, J., Feichtenhofer, C. (2022). Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 13587–13597).

Xie, S., Sun, C., Huang, J., Tu, Z., Murphy, K. (2018). Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. *Proceedings of the european conference on computer vision (eccv)* (pp. 305–321).

Yan, S., Xiong, X., Arnab, A., Lu, Z., Zhang, M., Sun, C., Schmid, C. (2022). Multiview transformers for video recognition. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 3333–3343).

Yang, J., Dong, X., Liu, L., Zhang, C., Shen, J., Yu, D. (2022). Recurring the transformer for video action recognition. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 14063–14073).

Yu, T., Li, X., Cai, Y., Sun, M., Li, P. (2022). S2-mlp: Spatial-shift mlp architecture for vision. *Proceedings of the ieee/cvf winter conference on applications of computer vision* (pp. 297–306).

Yuan, L., Chen, Y., Wang, T., Yu, W., Shi, Y., Jiang, Z.-H., ... Yan, S. (2021). Tokens-to-token vit: Training vision transformers from scratch on imagenet. *Proceedings of the ieee/cvf international conference on computer vision* (pp. 558–567).

Zhang, C., Gupta, A., Zisserman, A. (2021). Temporal query networks for fine-grained video understanding. *Proceedings of the ieee/cvf conference on computer vision and pattern recognition* (pp. 4486–4496).

Zhang, D.J., Li, K., Wang, Y., Chen, Y., Chandra, S., Qiao, Y., … Shou, M.Z. (2022). Morphmlp: An efficient mlp-like backbone for spatial-temporal representation learning. *Computer vision–eccv 2022: 17th european conference, tel aviv, israel, october 23–27, 2022, proceedings, part xxxv* (pp. 230–248).

Zhang, H., Cheng, L., Hao, Y., Ngo, C.-w. (2022). Long-term leap attention, short-term periodic shift for video classification. *Proceedings of the 30th acm international conference on multimedia* (pp. 5773–5782).

Zhang, H., Hao, Y., Ngo, C.-W. (2021). Token shift transformer for video classification. *Proceedings of the 29th acm international conference on multimedia* (pp. 917–925).

Zhang, S. (2022). Tfcnet: Temporal fully connected networks for static unbiased temporal reasoning. *arXiv preprint arXiv:2203.05928*, ,