DOI: 10.19884/j.1672-5220.202403011

An Efficient Temporal Decoding Module for Action Recognition

HUANG Qiubo*, MEI Jianmin, ZHAO Wupeng, LU Yiru, WANG Mei, CHEN Dehua
School of Computer Science and Technology, Donghua University, Shanghai 201620, China

Abstract: Action recognition, a fundamental task in the field of video understanding, has been extensively researched and applied. In contrast to an image, a video introduces an extra temporal dimension. However, many existing action recognition networks either perform simple temporal fusion through averaging or rely on pre-trained models from image recognition, resulting in limited temporal information extraction capabilities. This work proposes a highly efficient temporal decoding module that can be seamlessly integrated into any action recognition backbone network to enhance the focus on temporal relationships between video frames. Firstly, the decoder initializes a set of learnable queries, termed video-level action category prediction queries. Then, they are combined with the video frame features extracted by the backbone network after self-attention learning to extract video context information. Finally, these prediction queries with rich temporal features are used for category prediction. Experimental results on HMDB51, MSRDailyAct3D, Diving48 and Breakfast datasets show that using TokShift-Transformer and VideoMAE as encoders results in a significant improvement in Top-1 accuracy compared to the original models (TokShift-Transformer and VideoMAE), after introducing the proposed temporal decoder. The introduction of the temporal decoder results in an average performance increase exceeding 11% for TokShift-Transformer and nearly 5% for VideoMAE across the four datasets. Furthermore, the work explores the combination of the decoder with various action recognition networks, including Timesformer, as encoders. This results in an average accuracy improvement of more than 3.5% on the HMDB51 dataset. The code is available at https://github.com/huangturbo/TempDecoder.

Keywords: action recognition; video understanding; temporal relationship; temporal decoder; Transformer
CLC number: TP389.1 **Document code**: A
Article ID: 1672-5220(2025)02-0187-10

Open Science Identity (OSID)

0 Introduction

Action recognition aims to identify and classify human actions within a video sequence. With the widespread development and dissemination of video media, this field has attracted increasing research interest. As a fundamental area in computer vision, action recognition finds applications in various domains of video understanding, including video surveillance, retrieval and human activity detection. In contrast to static images, videos inherently contain temporal information. Therefore, for effective action recognition, models need to excel in feature extraction across both spatial and temporal dimensions. Spatial information represents the static details within each frame, including people, objects and backgrounds. Temporal information captures the contextual relationships between frames, requiring the integration of static information for a comprehensive understanding. Typically, the goal in action recognition tasks is to classify actions occurring between individuals, objects, or a combination of both.

For some datasets (e.g., UCF101[1]) where many categories are less related to the action and only one or a few frames of static images are often needed to recognize the action, models with excellent static information extraction capabilities can achieve high recognition accuracy[2]. However, in most real-life applications, relying solely on spatial information is insufficient. These video frames exhibit close correlations between actions over time. Enhancing the model ability to model temporal features becomes crucial for action recognition. Consequently, numerous studies have been conducted to address this issue. Early methods involved averaging the features extracted by convolutional neural networks (CNNs) for images across the temporal dimension, followed by classification[3]. However, such simple fusion methods often result in the loss of significant temporal information, hindering action classification. Subsequently, researchers replaced two-dimensional (2D) convolutions with three-dimensional (3D) counterparts to simultaneously model the temporal dimension[4], leading to improved performance. Nevertheless, the model parameters in this approach increase, resulting in substantial computational costs. To further enhance action classification accuracy, researchers incorporated optical flow[5] into the model, forming two-stream networks. However, pre-extracting optical flow

Received date: 2024-03-22
Foundation item: Shanghai Municipal Commission of Economy and Information Technology, China (No. 202301054)
* Correspondence should be addressed to HUANG Qiubo, email: huangturbo@dhu.edu.cn
Citation: HUANG Q B, MEI J M, ZHAO W P, et al. An efficient temporal decoding module for action recognition[J]. *Journal of Donghua University (English Edition)*, 2025, 42(2): 187-196.

incurs significant time and computational expenses, impeding the deployment of end-to-end models.

Following the remarkable success of the Transformer in natural language processing (NLP)[6], attempts were made to transfer the Transformer into the computer vision (CV) domain. In action recognition tasks, researchers utilized the multi-head attention mechanism of the Transformer to globally model spatiotemporal information, achieving promising recognition results. However, many existing Transformer-based action recognition models simply perform average fusion on the final output features or add a classification token representing the entire video for action recognition. This simplistic approach hampers the exploration of temporal correlations between different actions, leading to the loss of some crucial temporal information.

Therefore, we propose a straightforward Transformer temporal decoding module to model the video frame features extracted by existing action recognition networks. This module aims to further extract temporal information, thereby improving action recognition performance. In contrast to introducing a singular classification token, we pre-introduce learnable queries, termed video-level action category prediction queries. Drawing inspiration from end-to-end object detection with the Transformer[7], these prediction queries leverage the potent global modeling capabilities of the Transformer, enabling the decoder to autonomously learn spatiotemporal features within the video. Additionally, through the multi-head attention mechanism, the decoder can determine which temporal features are more critical, enhancing the overall spatiotemporal modeling capability of the entire model. We validate the effectiveness of the temporal decoder on HMDB51[8], MSRDailyAct3D[9], Diving48[10] and Breakfast[11] datasets.

1 Related Work

1.1 Video action recognition

Action recognition stands as a representative task in video understanding, and with the rise of deep learning, researchers have employed 3D CNNs to address video understanding challenges. Learning spatiotemporal features with 3D convolutional networks[12] is notable. Extending ResNeXt into a 3D structure, ResNeXt3D[13] further enhances the video understanding capabilities of 3D CNNs. However, due to limited receptive fields, 3D CNN approaches struggle to extract global spatiotemporal information, hindering the recognition of actions that span longer temporal ranges. To better describe temporal relationships between frames, Simonyan et al.[14] proposed two-stream networks incorporating optical flow into action recognition tasks. Compared to using raw RGB images as input, optical flow efficiently eliminates non-motion backgrounds, simplifying the learning task. However, precomputing optical flow fields entails significant computational and storage requirements, impeding large-scale training and real-time deployment.

In recent years, the tremendous success of the Transformers in handling sequential data has led to the exploration of Transformer-based[15] methods in video action recognition. Timesformer[16], the first Transformer-based video understanding network, employs a spatiotemporal attention mechanism to enable efficient and accurate video classification. The token shift module[17] segments video sequences into multiple fragments for local processing, reducing computational complexity and memory consumption. Addressing the need for substantial training data for the Transformer, Tong et al.[18] introduced VideoMAE, utilizing video self-supervised pretraining for efficient fine-tuning in downstream tasks like action recognition. These action recognition models, leveraging self-attention mechanisms, capture spatiotemporal dependencies across entire videos, extracting global and long-range temporal information to enhance the action recognition accuracy. However, these models primarily adopt an encoder structure, often performing simple fusion operations, such as averaging frame features and employing a single video action classification token for the action category prediction, resulting in the loss of some temporal action information. Additionally, compared to CNNs, Transformers generally lack local spatial awareness, impacting their static spatial modeling capabilities.

1.2 Temporal decoding modules

Recognizing the importance of modeling temporal features in video understanding, several temporal decoding modules have been proposed. Islam et al.[19] introduced Vis4mer, incorporating structured state-space temporal layers to enhance long-range video modeling capabilities while reducing model computational complexity. However, due to a lack of fine-grained action expression capabilities, it may struggle to identify subtle actions in shorter videos. Lin et al.[20] proposed an efficient video learner (EVL) decoder module that facilitated video action classification on the application to pre-trained image processing models. Yet, it requires the integration of multiple attention modules and imposes specific structural requirements on the backbone network, making it challenging to reuse in other models. Moreover, some work has migrated models from the NLP domain into video understanding networks to enhance temporal expression capabilities. Neimark et al.[21] employed Longformer[22] as a temporal decoding module for video action recognition. Kalfaoglu et al.[23] directly appended the bidirectional encoder representation from the Transformer[24] model to a 3D CNN-based action recognition network to boost temporal feature extraction capabilities of the 3D CNNs. While these decoding models exhibit strong temporal information extraction capabilities, their implementations are more complex, demanding larger computational resources. In contrast, our proposed model employs standard self-attention modules directly, making it simpler and more versatile,

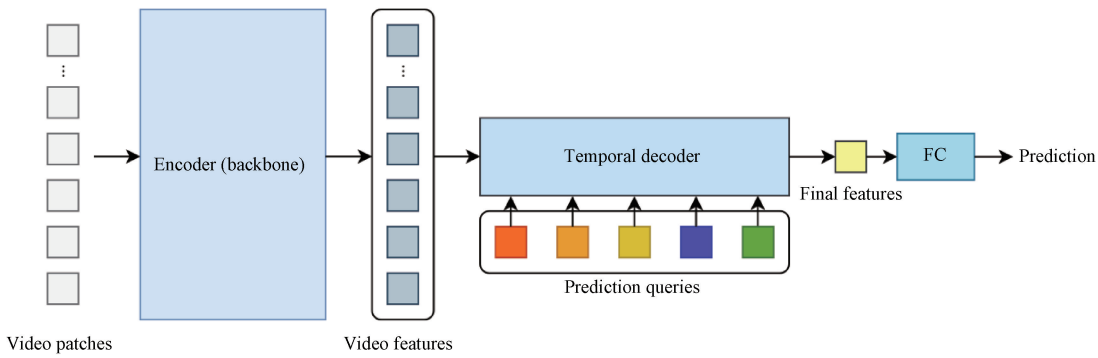
capable of being appended to any action recognition or image classification backbone network.

2 Proposed Temporal Decoding Module

To construct a simple yet efficient video temporal decoder, we propose a module to introduce category prediction queries into the temporal decoder. Similar to the visual Transformer, we utilize a video action recognition Transformer as the encoder. The sampled video frames are converted into patches and fed into the encoder, yielding frame features with rich spatial semantics. Subsequently, our proposed temporal decoder, which efficiently incorporates action category prediction queries, is employed to further analyze the temporal information within the frame features extracted by the encoder.

2.1 Overall architecture

The fundamental framework for video action recognition in this work is depicted in Fig. 1, presenting an encoder-decoder architecture. The encoder is responsible for extracting features from sampled RGB frames. Two video action recognition networks, TokShift-Transformer and VideoMAE, are used as encoders in experiments across multiple datasets. In TokShift-Transformer, we stack these frame features to construct spatiotemporal features, while VideoMAE uses the first frame feature as the spatiotemporal feature. To better leverage the encoded features, we do not directly use them for action category prediction. Instead, we feed these features into the proposed Transformer decoder to further explore temporal information and enhance the action recognition accuracy. This decoder initially performs self-attention operations on N_p learnable prediction queries, obtaining preliminary action learning information, where N_p is the number of query sequences. Subsequently, these N_p queries are combined with the previously encoded spatiotemporal features for a multi-head cross-attention operation. The resulting decoded feature vector contains rich temporal action information which is then fed into a linear layer to obtain action category predictions.



FC—full connection.

Fig. 1 Overall architecture of temporal decoding module

2.2 Encoder processing

Before performing action classification, we need to preprocess the input video frame sequence by using an encoder to extract rich temporal features. For this purpose, we can use the backbone network before the classification layer in any video action or image classification model as the encoder to extract features from video frames. Due to the powerful global modeling capabilities of Transformers, they have found widespread applications in video understanding domains like action recognition. We illustrate the encoding preprocessing of video frames using the TokShift-Transformer, a Transformer-based action recognition network.

Firstly, we have a video $\boldsymbol{V} \in \mathbf{R}^{T\times H\times W\times 3}$, where T, H and W represent the number of sampled video frames and the height and width of the spatial resolution of each frame, respectively; 3 represents the three channels of RGB in the image. This video is reshaped into a sequential tensor $\boldsymbol{V}^* \in \mathbf{R}^{T\times N\times d}$, where N represents the number of patches and $N=HW/P^2$, and P is the size of the patch; d denotes the number of RGB pixels in each patch, and $d=3P^2$.

Next, a linear layer module $\boldsymbol{E} \in \mathbf{R}^{d\times D}$ is applied to each patch $\boldsymbol{x}_i \in \mathbf{R}^{T\times d}$ in $\boldsymbol{V}^*$, where $\boldsymbol{x}_i$ indicates the ith patch, projecting the dimension of each patch to D. An additional classification token tensor $\boldsymbol{c} \in \mathbf{R}^{T\times D}$ is concatenated to the patches, representing the global information for each frame. Simultaneously, a position encoding embedding $\boldsymbol{E}_{\text{pos}} \in \mathbf{R}^{(N+1)\times D}$ is added to each patch, resulting in $z \in \mathbf{R}^{T\times(N+1)\times D}$:

$$z=[\boldsymbol{c}, \boldsymbol{x}_1\boldsymbol{E}, \boldsymbol{x}_2\boldsymbol{E}, \cdots, \boldsymbol{x}_i\boldsymbol{E}, \cdots, \boldsymbol{x}_N\boldsymbol{E}]+\boldsymbol{E}_{\text{pos}}. \quad (1)$$

This transforms the input video $\boldsymbol{V}^*$ into z. z is then input into multiple TokShift-MSA operations of the stacked TokShift-Transformer, resulting in $\tilde{z}$. $\tilde{z}$ has the same dimensions as z:

$$\tilde{z}=\text{TokShift-MSA}(z), \quad (2)$$

where TokShift-MSA$(\cdot)$ is the multi-head self-attention operation inherent to the TokShift model itself.

Finally, $\tilde{z}[:, 0, :]$ is taken as the output of the encoder $z' \in \mathbf{R}^{T\times D}$, representing the video features for T frames:

$$z' = \tilde{z}[:, 0, :]. \tag{3}$$

2.3 Proposed temporal decoder

2.3.1 Decoder structure

The proposed structure of the temporal decoder is illustrated in Fig. 2. The decoder consists of L layers of decoding units, and the core of each decoding unit consists of two attention modules. The key operation in these attention modules is the multi-head attention (MHA) mechanism.

We construct queries for action category prediction. The queries are initialized to zero and represented as $\boldsymbol{q} \in \mathbf{R}^{N_p \times D}$. We add learnable position encodings $\boldsymbol{Q}_{pos} \in \mathbf{R}^{N_p \times D}$ to form $\boldsymbol{q}'$. Performing a multi-head self-attention operation on $\boldsymbol{q}'$ facilitates initial learning among queries, resulting in a sequence $\boldsymbol{Q}_p \in \mathbf{R}^{N_p \times D}$ with certain temporal action representation capabilities:

$$\boldsymbol{q}' = \boldsymbol{q} + \boldsymbol{Q}_{pos}, \tag{4}$$

$$\boldsymbol{Q}_p = \mathrm{MHSA}(\boldsymbol{q}'), \tag{5}$$

where MHSA $(\cdot)$ is the multi-head self-attention operation.

The video features z' extracted from the encoder undergo a cross-time spatial convolution module, aimed at enhancing the spatial modeling capability of the decoder. This process yields $z'' \in \mathbf{R}^{T \times D}$. Similar to the prediction queries, we add position encoding $\boldsymbol{P}_{KV} \in \mathbf{R}^{N_{pos} \times D}$ to z'', obtaining frame features $\boldsymbol{K}_f \in \mathbf{R}^{T \times D}$ and $\boldsymbol{V}_f \in \mathbf{R}^{T \times D}$ with enhanced spatial expression capabilities:

$$z'' = \mathrm{CTSC}(z'), \tag{6}$$

$$\boldsymbol{K}_f = \boldsymbol{V}_f = z'' + \boldsymbol{P}_{KV}[:, \mathrm{T}], \tag{7}$$

where CTSC $(\cdot)$ is the cross-time spatial convolution operation. The position encoding uses trigonometric absolute position encoding, and N_{pos} represents the pre-set number of position encodings ($N_{pos} = 2\ 048$ in this work).

$$\begin{cases} \boldsymbol{P}_{KV}(k, 2i) = \sin(k/10\ 000^{2i/d}), \\ \boldsymbol{P}_{KV}(k, 2i+1) = \cos(k/10\ 000^{2i/d}), \end{cases} \tag{8}$$

where $k = 0, 1, \cdots, N_{pos}-1$; $i = 0, 1, \cdots, D/2-1$.

Next, $\boldsymbol{K}_f$ and $\boldsymbol{V}_f$ are combined with $\boldsymbol{Q}_p$ for a multi-head cross attention operation, enabling global perception of spatiotemporal information through attention mechanisms. The output is a vector $z_{cross} \in \mathbf{R}^{N_{pos} \times D}$ with rich temporal features:

$$z_{cross} = \mathrm{MHCA}(\boldsymbol{Q}_p, \boldsymbol{K}_f, \boldsymbol{V}_f), \tag{9}$$

where MHCA $(\cdot)$ is the multi-head cross attention operation.

After multiple layers of stacking operations, the output z_{cross} undergoes further processing through the proposed temporal convolution module. This module performs convolution fusion on the temporal dimension, resulting in a feature $z_{out} \in \mathbf{R}^D$ with rich spatial-temporal semantic information for the final linear classification prediction.

$$z_{out} = \text{Temp-Conv}(z_{cross}). \tag{10}$$

where Temp-Conv $(\cdot)$ is the temporal convolution operation.

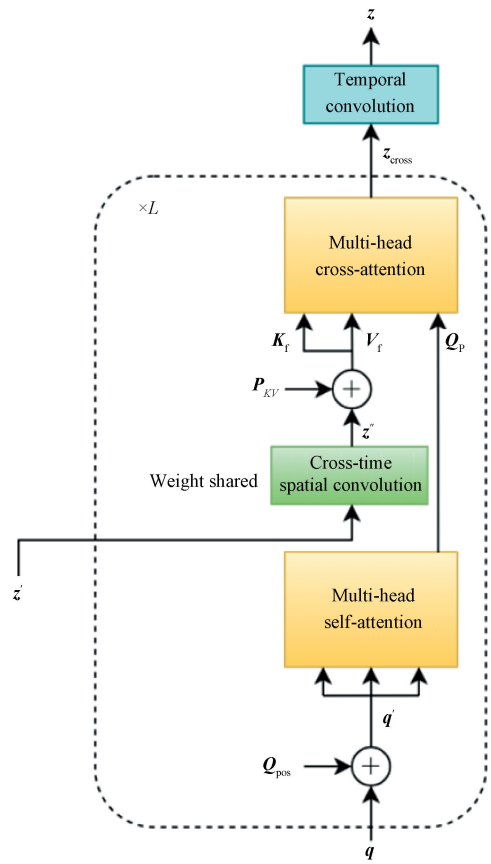


Fig. 2 Schematic diagram of temporal decoder structure

2.3.2 Cross-time spatial convolution and temporal convolution modules

Despite the powerful global temporal modeling capabilities of the Transformer attention mechanism, its ability for local spatial perception is relatively weak. To address this limitation, we introduce a weight-shared cross-time spatial convolution module. This module participates in convolutional operations on the frame features z' before entering the cross attention module, thereby enhancing the spatial modeling capability of the decoder. Generally, assuming that the input is $\boldsymbol{X}_f \in \mathbf{R}^{T \times D}$ and the feature after convolution is $\boldsymbol{Y}_{cross} \in \mathbf{R}^{T \times D}$, the operation is defined as follows.

$$\boldsymbol{Y}_{cross}(t, d) = \boldsymbol{b}_{cross} + \sum_{\Delta t, \Delta d} \boldsymbol{W}_{cross}(\Delta t, \Delta d)\, \boldsymbol{X}_f(t + \Delta t, d + \Delta d), \tag{11}$$

where $\boldsymbol{b}_{cross}$ and $\boldsymbol{W}_{cross}$ are the bias and weight parameters of cross-time spatial convolution, respectively; $\Delta t, \Delta d \in \{-1, 0, 1\}$. For Δt, -1 and 1 represent the neighboring frames before and after the current frame in the video,

respectively. For Δd, -1 and 1 represent the positions before and after the current position d, respectionly. As shown in Fig. 3, to achieve spatial feature fusion across frames for the current temporal position, we convolve with the neighboring frames in the temporal and spatial dimensions, using zero-padding at the boundaries. The elements with the same color in Fig. 3 are multiplied and accumulated element-wise, and the result is added to obtain the instance corresponding to the above operation.

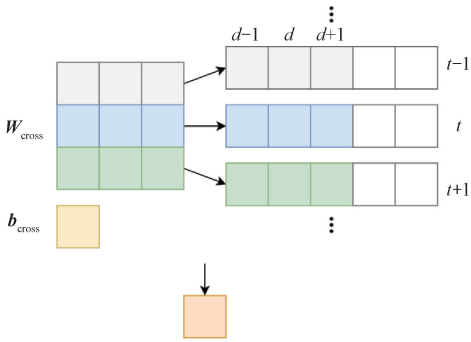


Fig. 3 Cross-time spatial convolution operation

To enhance the temporal fusion capability of the decoder, we also introduce a temporal convolution module, acting on the output feature of the last layer's multi-head cross attention operation. This module performs convolution fusion on the temporal dimension, producing the final output feature used for video action classification. Generally, assuming the input is $X_{\text{cross}} \in \mathbf{R}^{T\times D}$ and the feature after temporal convolution is $Y_{\text{temp}} \in \mathbf{R}^{D}$, the temporal convolution module's fusion operation is

$$\boldsymbol{Y}_{\text{temp}}(d) = \boldsymbol{b}_{\text{temp}} + \sum_{t=0}^{T-1} \boldsymbol{W}_{\text{temp}}(t)\boldsymbol{X}_{\text{cross}}(t,d), \quad (12)$$

where $\boldsymbol{b}_{\text{temp}}$ and $\boldsymbol{W}_{\text{temp}}$ are the bias and weight parameters of the temporal convolution module. The convolution weights learned by the network and the input features are multiplied and accumulated in the temporal dimension, resulting in an output feature with more temporal information.

3 Experiments

Extensive experiments were conducted on four representative datasets for video action recognition, and the accuracy was used as the evaluation metric.

3.1 Datasets

Four datasets are used for the experiments, namely HMDB51, MSRDailyAct3D, Diving48 and Breakfast. The HMDB51 dataset comprises approximately 7 000 video clips, encompassing a total of 51 action categories. These video clips involve some similar actions, with a wealth of action information that cannot be identified solely through static frame images. The dataset defines three data splits, and split 1 is used in the experiment. The MSRDailyAct3D dataset consists of indoor video clips capturing human activities, featuring 16 action categories. Due to minimal background variation in this dataset, the model needs to exhibit robust action modeling capabilities. The Diving48 dataset is a fine-grained dataset of competitive diving videos, consisting of 48 precisely defined diving actions and approximately 18 000 video clips, and each diving action is divided into three sub-stages. As all videos are shot in the same setting with subtle differences between various diving actions, prolonged dynamic fine-grained modeling is necessary. The Breakfast dataset includes 10 complex actions related to breakfast preparation, executed by 52 different individuals across 18 distinct kitchens. This dataset comprises long-duration videos in a natural environment, with an average video duration exceeding 2 min, placing demands on the model ability for long-range modeling.

3.2 Implementation details

The AdamW optimizer was used for model training, and the initial learning rate was 1×10^{-4}. Additionally, ReduceLROnPlateau was used to dynamically adjust the learning rate based on the accuracy. For HMDB51 and Diving48 datasets, we randomly sampled 16 frames uniformly as input; for the MSRDailyAct3D dataset, we uniformly sampled eight frames; for the Breakfast dataset, we randomly sampled 32 frames as input video frames. All video frames from each dataset were cropped to 224 pixel × 224 pixel before input. Random cropping, random horizontal flipping and normalization were employed as data augmentation techniques during training.

3.3 Main results

To validate the effectiveness of the proposed temporal decoding module, we employed the typical Transformer-based video action classification models (TokShift-Transformer and VideoMAE) which were pre-trained on the Kinetics-400 dataset. We integrated the temporal decoder before the classification layer of these models while keeping the rest of the architecture unchanged. The comparison of the Top-1 accuracy results with and without the temporal decoder for both models is presented in Table 1.

Table 1 Comparison of Top-1 accuracy with and without temporal decoder

Model	Top-1 accuracy/%			
	HMDB51	MSRDailyAct3D	Diving48	Breakfast
TokShift-Transformer	54. 64	73. 44	68. 93	69. 30
TokShift-Transformer with temporal decoder	**67. 78**	**84. 50**	**76. 80**	**84. 23**
VideoMAE	73. 30	90. 62	75. 33	74. 93
VideoMAE with temporal decoder	**80. 52**	**98. 44**	**77. 26**	**77. 46**

It is evident that the integration of the proposed temporal decoder into both encoder models significantly improves action classification accuracy. Specifically, on HMDB51, MSRDailyAct3D and Breakfast datasets, the accuracy improvement of TokShift-Transformer with the proposed temporal decoder exceeds 10% compared to that of the original model. Therefore, the proposed temporal decoder can effectively parse temporal dynamic information and enhance the discriminative ability of action classification. On the Diving48 dataset, both models show a performance improvement, indicating that the decoder enhances the overall models fine-grained recognition capability. Furthermore, the inclusion of the decoder also improves the classification performance on the long-range video dataset (Breakfast), demonstrating that the model can alleviate issues related to long-range temporal dependencies.

In addition, further comparisons were conducted by training these two networks for 20 epochs on the HMDB51 dataset, as shown in Fig. 4. The results indicate that the temporal decoder significantly enhances the model ability to extract temporally differentiated features. For the TokShift-Transformer, convergence is achieved in approximately three epochs with the temporal decoder, compared to around eight epochs without it. This improvement in the recognition capability, coupled with a reduction in training iterations, further validates the effectiveness of the proposed temporal decoder.

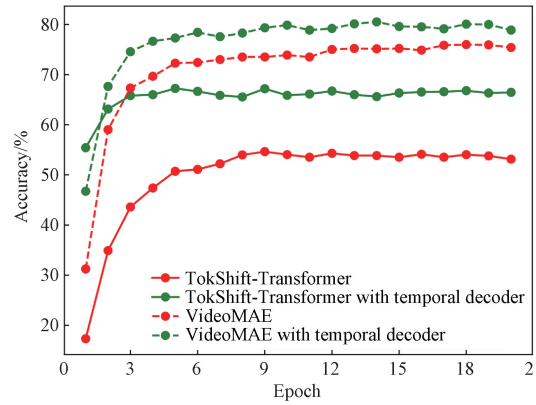


Fig. 4 Training results for different models on HMDB51 dataset

To further validate the effectiveness of the proposed temporal decoder, a performance comparison with other previously proposed efficient temporal decoders was conducted on the HMDB51 dataset. We used both TokShift-Transformer and VideoMAE as encoders for the encoding process. The results are presented in Table 2. The computing performance of a processor is measured in terms of billions of floating point operations it can perform in one second, namely Giga floating point operations per second (GFLOPs). Params refer to the parameters or variables that define the structure and behavior of a statistical or machine learning model.

Table 2 Performance comparison with other temporal decoders on HMDB51 dataset

Temporal decoder	Top-1 accuracy/%		Computing performance/GFLOPs		Params/MB	
	TokShift-Transformer	VideoMAE	TokShift-Transformer	VideoMAE	TokShift-Transformer	VideoMAE
Vis4mer	61.11	78.95	**270.09**	103.56	**87.13**	**66.46**
BERT	66.80	79.54	270.43	**101.89**	106.94	86.27
Longformer	66.21	79.80	270.84	102.22	110.49	89.82
Proposed temporal decoder	**67.78**	**80.52**	270.19	101.97	108.15	87.47

We compared the proposed temporal decoder with Vis4mer, BERT and Longformer, three decoders known for their temporal feature extraction capabilities. The input frame number was 16 with a resolution of 224 pixel × 224 pixel. As shown in Table 2, Vis4mer achieves lower parameters, but the proposed temporal decoder model, under the TokShift-Transformer baseline, exhibits a more significant improvement in the recognition accuracy compared to Vis4mer. Additionally, the proposed temporal decoder outperforms both BERT and Longformer when considering similar computational complexity and parameters. Notably, the proposed temporal decoder is simpler to implement, facilitating deployment and application. In summary, the temporal decoding module strikes a good balance between the computing performance and computational complexity.

To validate the scalability and generality of the proposed temporal decoder, experiments were conducted on the HMDB51 dataset by inserting the proposed temporal decoder into various action recognition encoders (backbones) in addition to TokShift-Transformer and VideoMAE. The results are summarized in Table 3. It is demonstrated that integrating the temporal decoder leads to varying degrees of improvement in the recognition accuracy on the HMDB51 dataset for all three backbone networks. This further validates the temporal feature extraction capabilities of the temporal decoder and indicates its versatility in extending to other domains of video understanding.

Table 3 Performance comparison with different encoders on HMDB51 dataset

Encoder	Top-1 accuracy/%	
	Without decoder	With decoder
Timesformer	63.59	**69.54**
ResNeXt3D-101	64.58	**66.14**
ViT-B	52.48	**61.96**

3.4 Ablation experiments

A series of ablation experiments were conducted based on TokShift-Transformer.

3.4.1 Number of queries for video-level action category prediction

To investigate the impact of different query quantities for action category prediction on model performance, various sizes of N_p were used in the experiment. The results are presented in Table 4. It is evident that N_p has varying effects on the experimental results. Initially, as the number of queries increases, the model incorporates richer action information, leading to a continuous improvement in the action recognition accuracy. However, as the number of queries continues to increase, it exhibits a declining trend. Based on the experimental results, a query number of six achieves the best results. Therefore, in subsequent experiments, unless otherwise specified, we set the number of queries to six.

Table 4 Ablation experiment on different numbers of queries for action category prediction on HMDB51 dataset

Number of queries	Top-1 accuracy/%
1	65.29
3	66.14
6	**67.78**
10	66.80
16	65.42

3.4.2 Number of decoder layers

To assess the impact of different numbers of stacked attention modules in the temporal decoder on the model, various numbers of decoder layers L were used in the experiments. The model performance is presented in Table 5.

Table 5 Ablation experiment on different numbers of decoder layers on HMDB51 dataset

Number of decoder layers	Top-1 accuracy/%
1	65.36
3	66.34
6	**67.78**
12	66.21

Similar to the impact of the number of queries for action category prediction, the model accuracy on the HMDB51 dataset shows an increasing trend followed by a decline as the number of decoder layers increases. Comparative analyses of the results reveal that the optimal performance is achieved when L is six. As L continues to increase, the accuracy does not improve. This may be attributed to the fact that excessively deep stacking of attention modules can lead to the extraction of overly consistent spatiotemporal features, weakening the model ability to represent differences between various actions and consequently reducing its action recognition capability.

3.4.3 Cross-time spatial convolution module

We compared the impact of the cross-time spatial convolution module on the accuracy on the HMDB51 dataset. The results are shown in Table 6. Additionally, the effect of a single weight-shared convolution or multiple layers of non-weight-shared convolution was also analyzed when incorporating the cross-time spatial convolution module.

Table 6 Impact of cross-time spatial convolution module on accuracy on HMDB51 dataset

Model	Cross-time spatial convolution	Shared weight	Top-1 accuracy/%
TokShift-Transformer	×	—	66.27
TokShift-Transformer	√	×	66.14
TokShift-Transformer	√	√	**67.78**

The introduction of the cross-time spatial convolution module leads to an improvement in the accuracy by more than 1%. This module models spatial relationships between adjacent temporal regions and integrates with the attention mechanism of the Transformer, aiding in a more comprehensive extraction of spatiotemporal information and thereby enhancing the overall model performance. Additionally, compared to using a single weight-shared convolution, employing multiple layers of non-weight-shared convolution leads to a decline in the model performance. This suggests that an excessive introduction of this module is detrimental to network optimization and increases parameters, adding a burden to the network. Therefore, the decoder is constructed with a single weight-shared convolution.

4 Model Visualization

4.1 Visualization of cross-attention feature maps in decoder

A video clip from the HMDB51 dataset was randomly selected as input to the model and visualized the cross-attention feature maps for each layer of the temporal decoder, as shown in Fig. 5. The horizontal axis represents the temporal locations of video frame features, indicating the time dimension, while the vertical axis represents the sequence in the action category prediction queries.

From Fig. 5, it can be observed that in the first layer of the decoder, the features between action states are relatively chaotic, and the boundaries are not clearly defined. From the second layer to the fifth layer, the temporal information is gradually learned by the network, and corresponding features for each time period are gradually formed, presenting a pattern of vertical lines. By the sixth layer, the model assigns different weights based on the importance of different states within the entire action, aiding the model in better distinguishing

action categories. Furthermore, compared to those of the fourth layer, the feature map weights in the sixth layer tend to be mostly towards zero, reducing the weights of more video frame features and filtering out a significant amount of redundant information. The attention mechanism segregates the most important parts, further assisting the model in more accurately identifying and determining video actions.

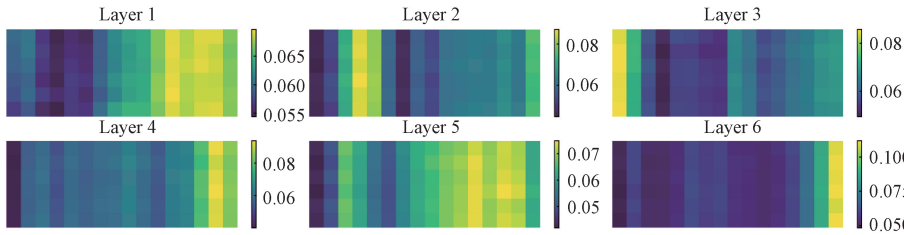


Fig. 5 Visualization of cross-attention feature maps in decoder

4.2 Qualitative analysis

We selected several video clips from the HMDB51 dataset and compared action category predictions between the TokShift-Transformer and the model with the proposed temporal decoder, as shown in Fig. 6. In each case, the odd-numbered rows represent the original video frames, while the even-numbered rows represent the corresponding attention feature maps from the TokShift-Transformer encoder with the proposed temporal decoder. The entries within parentheses represent the predicted action categories by TokShift-Transformer, while those preceding the parentheses indicate the actual categories predicted after incorporating the proposed decoder.

Fig. 6 Qualitative comparisons of samples on HMDB51 dataset: (a) cartwheel (flic_flac); (b) flic_flac (cartwheel); (c) hit (swing_baseball); (d) pick (golf); (e) run (dribble); (f) sword_exercise (draw_sword)

Upon comparison, it is evident that when faced with highly similar actions such as "cartwheel" and "flic_flac", misclassifications occur in these video clips by using the original model, whereas the model with the

temporal decoder correctly identifies them. This suggests that the temporal decoder can further extract spatiotemporal features and effectively capture subtle differences in actions. Additionally, comparing the prediction results for the last two clips, it is observed that with the temporal decoder, the model not only focuses on local state information but also expresses global temporal dynamics. This further illustrates that the proposed temporal decoder can analyze contextual information in videos, enhancing the overall temporal modeling capability of the model. Moreover, by examining the attention feature maps, it is apparent that the model emphasizes key parts where actions occur, aiding in the model ability to determine action categories.

5 Conclusions

We have proposed an efficient temporal decoding module for video action recognition based on action classification prediction queries. This module, by pre-constructing a certain number of prediction queries and leveraging the powerful global temporal modeling capability of Transformer attention mechanisms, autonomously learns to classify different actions. The temporal decoder is conceptually and implementation-wise straightforward, making it easily extensible to other action recognition models with relatively low additional cost for efficient temporal modeling, thus improving the action classification accuracy. Moreover, incorporating this temporal decoder enhances the discriminative ability for subtle actions, particularly helpful in scenarios involving similar action sequences. Our experiments demonstrate that the constructed decoder improves the recognition of actions in longer-duration videos, effectively addressing issues related to long-range dependencies. This study may contribute to similar tasks in the field of video understanding in the future.

References

[1] SOOMRO K, ZAMIR A R, SHAH M. UCF101: a dataset of 101 human actions classes from videos in the wild[EB/OL]. (2012-12-03)[2024-03-01]. https://arxiv.org/pdf/1212.0402v1.

[2] ZHU Y, LI X Y, LIU C H, et al. A comprehensive study of deep video action recognition[EB/OL]. (2020-12-11)[2024-03-01]. https://arxiv.org/pdf/2012.06567.

[3] KARPATHY A, TODERICI G, SHETTY S, et al. Large-scale video classification with convolutional neural networks[C]//Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. New York, USA: IEEE, 2014: 1725-1732.

[4] JI S W, XU W, YANG M, et al. 3D convolutional neural networks for human action recognition[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, 35(1): 221-231.

[5] HORN B K P, SCHUNCK B G. Determining optical flow[J]. *Artificial Intelligence*, 1981, 17(1/2/3): 185-203.

[6] VASWANI A, SHAZEER N, PARMAR N, et al. Attention is all you need[EB/OL]. (2023-08-02)[2024-03-01]. https://arxiv.org/pdf/1212.0402v1.

[7] CARION N, MASSA F, SYNNAEVE G, et al. End-to-end object detection with transformers[M]//Lecture Notes in Computer Science. Cham, Switzerland: Springer International Publishing AG, 2020: 213-229.

[8] KUEHNE H, JHUANG H, GARROTE E, et al. HMDB: a large video database for human motion recognition[C]//2011 International Conference on Computer Vision. New York, USA: IEEE, 2011: 2556-2563.

[9] WANG J, LIU Z C, WU Y, et al. Mining actionlet ensemble for action recognition with depth cameras[C]//2012 IEEE Conference on Computer Vision and Pattern Recognition. New York, USA: IEEE, 2012: 1290-1297.

[10] LI Y W, LI Y, VASCONCELOS N. RESOUND: towards action recognition without representation bias[M]//Lecture Notes in Computer Science. Cham, Switzerland: Springer International Publishing AG, 2018: 520-535.

[11] KUEHNE H, GALL J, SERRE T. An end-to-end generative framework for video segmentation and recognition[C]//2016 IEEE Winter Conference on Applications of Computer Vision (WACV). New York, USA: IEEE, 2016: 1-8.

[12] TRAN D, BOURDEV L, FERGUS R, et al. Learning spatiotemporal features with 3D convolutional networks[C]//Proceedings of the IEEE International Conference on Computer Vision. New York, USA: IEEE, 2015: 4489-4497.

[13] GESSERT N, SCHLÜTER M, SCHLAEFER A. A deep learning approach for pose estimation from volumetric OCT data[J]. *Medical Image Analysis*, 2018, 46: 162-179.

[14] SIMONYAN K, ZISSERMAN A. Two-stream convolutional networks for action recognition in videos[J]. *Advances in Neural Information Processing Systems*, 2014, 1: 568-576.

[15] DOSOVITSKIY A, BEYER L, KOLESNIKOV A, et al. An image is worth 16×16 words: Transformers for image recognition at scale[EB/OL]. (2020-10-22)[2024-03-01]. https://arxiv.org/pdf/2010.11929v1.

[16] BERTASIUS G, WANG H, TORRESANI L. Is space-time attention all you need for video understanding?[EB/OL]. (2021-06-09)[2024-

03-01]. https://arxiv.org/pdf/2102.05095.
[17] ZHANG H, HAO Y B, NGO C W. Token shift transformer for video classification [C]// Proceedings of the 29th ACM International Conference on Multimedia. New York, USA: ACM, 2021: 917-925.
[18] TONG Z, SONG Y B, WANG J, et al. VideoMAE: masked autoencoders are data-efficient learners for self-supervised video pre-training[EB/OL]. (2022-10-18)[2024-03-01]. https://arxiv.org/pdf/2203.12602v3.
[19] ISLAM M M, BERTASIUS G. Long movie clip classification with state-space video models [M]//Lecture Notes in Computer Science. Cham, Switzerland: Springer Nature Switzerland AG, 2022: 87-104.
[20] LIN Z Y, GENG S J, ZHANG R R, et al. Frozen CLIP models are efficient video learners [M]//Lecture Notes in Computer Science. Cham, Switzerland: Springer Nature Switzerland AG, 2022: 388-404.
[21] NEIMARK D, BAR O, ZOHAR M, et al. Video transformer network[EB/OL]. (2021-08-17)[2024-03-01]. https://arxiv.org/pdf/2102.00719.
[22] BELTAGY I, PETERS M E, COHAN A. Longformer: the long-document transformer[EB/OL]. (2020-12-02)[2024-03-01]. https://arxiv.org/pdf/2004.05150.
[23] KALFAOGLU M E, KALKAN S, ALATAN A A. Late temporal modeling in 3D CNN architectures with BERT for action recognition [M]//BARTOLI A, FUSIELLO A, eds. Lecture Notes in Computer Science. Cham, Switzerland: Springer International Publishing AG, 2020: 731-747.
[24] DEVLIN J, CHANG M W, LEE K, et al. BERT: pre-training of deep bidirectional transformers for language understanding [EB/OL]. (2020-12-02)[2024-03-01]. https://arxiv.org/pdf/1810.04805.

用于动作识别的高效时序解码模块

黄秋波*，梅建敏，赵武鹏，卢怡如，王 梅，陈德华
东华大学 计算机科学与技术学院，上海 201620

摘 要：动作识别作为视频理解领域的基础性任务，得到了广泛研究和应用。相比图像，视频媒介增加了时间维度。然而，目前许多动作识别网络只是对时序信息进行简单平均融合，或者是由图像的预训练模型迁移而来，对时序信息的抽取能力较弱。该文提出了一个可拼接在任意动作识别骨干网络上的高效时序解码模块，以进一步关注视频帧之间的时序关系。该解码器首先初始化一定数量的可学习查询张量，即视频级的动作类别预测查询张量，进行自注意力学习后与骨干网络得到的视频帧特征相结合来提取视频上下文信息，最后再使用这些具有丰富时序特征的预测查询张量作分类预测。在 HMDB51、MSRDailyAct3D、Diving48 及 Breakfast 数据集上将 TokShift-Transformer 和 VideoMAE 作为编码器进行实验。实验结果显示，与原始模型 TokShift-Transformer 和 VideoMAE 相比，引入所提出的时序解码器后，Top-1 准确率得到明显提升。在这四个数据集上，TokShift-Transformer 的性能提高超过 11%，在 VideoMAE 上的准确率也平均提高了近 5%。该工作还将包括 Timesformer 在内的其他动作识别网络作为编码器，使其与解码器进一步结合。实验结果表明，在 HMDB51 数据集上，准确率平均提高 3.5%以上。代码见 https://github.com/huangturbo/TempDecoder。

关键词：动作识别；视频理解；时序关系；时序解码器；Transformer