Efficient Video Transformers via Spatial-temporal Token Merging for Action Recognition

ZHANZHOU FENG, National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University, China
JIAMING XU, School of Electronic Engineering and Computer Science, Peking University, China
LEI MA, National Biomedical Imaging Center, College of Future Technology, Peking University, National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University, Beijing Academy of Artificial Intelligence, China
SHILIANG ZHANG, National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University, China

Transformer has exhibited promising performance in various video recognition tasks but brings a huge computational cost in modeling spatial-temporal cues. This work aims to boost the efficiency of existing video transformers for action recognition through eliminating redundancies in their tokens and efficiently learning motion cues of moving objects. We propose a lightweight and plug-and-play module, namely Spatial-temporal Token Merger (STTM), to merge the tokens belonging to the same object into a more compact representation. STTM first adaptively identifies crucial object clues underlying the video as meta tokens. Similarity scores between input tokens and meta tokens are hence computed and used to guide the fusion of similar tokens in both spatial and temporal domains, respectively. To compensate for motion cues lost in the merging procedure, we compute the linear aggregation of spatial-temporal positions of tokens as motion features. STTM hence outputs a compact set of tokens fusing both appearance and motion features of moving objects. This procedure substantially decreases the number of tokens that need to be processed by each Transformer block and boosts the efficiency. As a general module, STTM can be applied to different layers of various video Transformers. Extensive experiments on the action recognition datasets Kinectics-400 and SSv2 demonstrate its promising performance. For example, it reduces the computation complexity of ViT by 38% while maintaining a similar performance on Kinectics-400. It also brings 1.7% gains of top-1 accuracy on SSv2 under the same computational cost.

CCS Concepts: • **Computing methodologies → Activity recognition and understanding;**

Additional Key Words and Phrases: Efficient video recognition, deep learning, transformer, spatial-temporal information

This work is supported in part by The National Key Research and Development Program of China under Grant No. 2018YFE0118400, in part by Natural Science Foundation of China under Grant No. U20B2052, 61936011.
Authors' addresses: Z. Feng and S. Zhang, National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University, Beijing, China, 100871; e-mails: fengzz@stu.pku.edu.cn, slzhang.jdl@pku.edu.cn; J. Xu, School of Electronic Engineering and Computer Science, Peking University, Beijing, China, 100871; e-mail: 2000012915@stu.pku.edu.cn; L. Ma, National Biomedical Imaging Center, College of Future Technology, Peking University, and National Key Laboratory for Multimedia Information Processing, School of Computer Science, Peking University, and Beijing Academy of Artificial Intelligence , Beijing, China, 100871; e-mail: lei.ma@pku.edu.cn.

1551-6857/2024/01-ART120 $15.00
https://doi.org/10.1145/3633781

ACM Reference format:
Zhanzhou Feng, Jiaming Xu, Lei Ma, and Shiliang Zhang. 2024. Efficient Video Transformers via Spatial-temporal Token Merging for Action Recognition. *ACM Trans. Multimedia Comput. Commun. Appl.* 20, 4, Article 120 (January 2024), 21 pages.
https://doi.org/10.1145/3633781

1 INTRODUCTION

Over the last few years, has revolutionized the backbone architectures of some vision tasks. Vision Transformers treat an image as a set of non-overlapping patches [10]. They hence adopt the self-attention mechanism to learn arbitrary functions over patches and exhibit strong capabilities in modeling long-range spatial dependencies [50]. This property makes Transformers achieve substantial performance gains across a wide range of image recognition tasks, including classification [10, 19, 25, 48], detection [5, 71], and segmentation [52, 59, 69].

Video recognition aims to recognize semantics in a sequence of frames. It needs to refer to both the appearance and motion cues embedded in video sequences. The potential of vision Transformers in modeling long-range dependencies has inspired their adaption in video recognition. Existing video Transformers divide each frame into a set of patches as input. This straightforward strategy makes it hard to deploy video Transformers in resource-constrained scenarios and mobile devices. For instance, the number of patches is T times more than its image counterpart, where T is the number of video frames. The computational complexity of self-attention is quadratic to the number of patches, which leads to significant computational overheads and memory consumption.

This issue has drawn attention from the community. Some recent works treat video as a sequence of frames and leverage efficient Transformers from the image domain for video recognition. A commonly followed intuition is to drop unimportant tokens [20, 41, 51]. Those works have achieved decent performance. Nevertheless, treating video as a collection of individual frames is not helpful for modeling temporal contextual cues, and hence may exhibit degraded performance in tasks relying on motion cues like action recognition. This raises an important question: how to boost the efficiency of video Transformers while modeling the motion cues in videos.

A video typically consists of multiple objects moving over time. Figure 1 depicts the action of a boy playing soccer, which contains three objects: the boy, soccer, and background. To understand the semantics or action in this video clip, the model needs to learn both the appearance cues and motion cues of each object . For example, patches related to the boy can be integrated into the corresponding appearance information of a boy with blonde hair and wearing jeans, and his motion postures that of a boy kicking the soccer ball. Adjacent frames usually contain the same object but with changing poses and spatial locations. Therefore, there exist considerable redundancies in appearance cues across frames. Meanwhile, the motion cues of each object can be represented by the sparse trajectory, e.g., the bias of locations across frames. The above observation motivates us to learn compact appearance cues and motion cues with video Transformers. We spot and merge tokens showing similar cues to eliminate spatial and temporal redundancies. The spatial trajectory along the temporal axis is leveraged to compute the motion features.

This article introduces the **Spatial-temporal Token Merger (STTM)** module to learn the compact spatial-temporal representation. Given a sequence of input tokens for a video Transformer block, STTM adopts two stages to ensure merging efficiency: (1) token merging inside each frame and (2) token merging across frames. The token merging in two stages follows the same intuition by (1) first computing meta tokens representing important object cues and (2) merging tokens similar to the same meta token as a uniform one. Specifically, meta tokens are input adaptive and generated by selecting most orthogonal features in the given token sequence following [42]. The

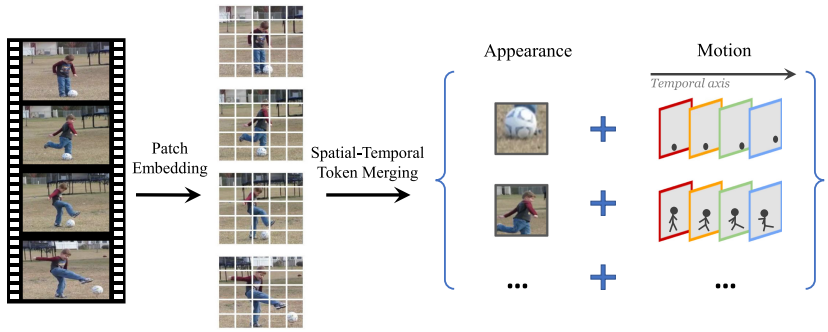

Fig. 1. **An example video clip of a boy playing.** The appearance cues of the football to soccer and the boy are redundant across frames. Their motion cues can be modeled by a sparse trajectory. Unlike the original patch embedding module, the proposed STTM module integrates similar patches and their motion information into uniform representations.

module hence measures the correspondence scores between each input token and meta tokens. The correspondence scores guide the token merging procedure; i.e., tokens depicting the same object or semantic are aggregated into a more compact one.

The merging in two stages aggregates appearance cues of input tokens in both the spatial and temporal domains, leading to a more compact token representation robust to appearance variances. To further extract the trajectory cues, we aggregate the high-dimension spatial-temporal positional embeddings of tokens as the motion feature. Since the high-dimensional positional embeddings can be quasi-orthogonal [10], their combination effectively represents variance of locations over time. Each merged token by STTM is fused with an aggregated appearance feature and a motion feature and hence is regarded as the input of the subsequent Transformer block. The STTM model can be flexibly inserted into different Transformer blocks to decrease the number of their input tokens. More details will be presented in Section 3.2.

We validate the effectiveness of STTM by applying it to popular video Transformer models [3, 10] on two large-scale action recognition datasets, Kinetics-400 [26] and **Something-Something-v2 (SSv2)** [18]. Experimental results demonstrate that the STTM effectively improves the efficiency of video Transformers while maintaining their original recognition accuracy. When applied to TimeSformer [3], STTM reduces the computation cost measured in **Giga floating-point operations (GFLOPs)** by 38%, with a marginal top-1 performance drop of 0.9% on the SSv2 dataset. It decreases the computation complexity of ViT [10] by 36%, with a 0.8% drop in top-1 accuracy on SSv2. A more efficient Transformer allows us to increase the input video resolution while keeping a consistent computational cost. This operation consistently boosts the performance of existing video Transformers, e.g., by 0.8% and 1.7% for ViT and TimeSformer, respectively.

The proposed method aims to enhance the efficiency of video Transformers by reducing the computational complexity without compromising the performance. Extensive experiments have shown the competitive performance of STTM. For instance, as shown in Table 6, applying our method to ViT-S [10] backbones reduces the computation cost by 41.8% with only a 1.3% and 0.7% performance drop. With the same computation as the baseline method, the proposed method substantially improves the model performance. Upon TimeSformer [3], the proposed method yields +0.6% and +0.6% performance gains on the K400 and SSv2 datasets, respectively, with fewer computations and parameters compared to the baseline. Additionally, fine-grained patch-level merging incurs lower training costs. For example, on the Kinetics 400 dataset, our method outperforms X3D [13] by +0.4%, with 273 times less computation cost and 60.7% less computational complexity.

Therefore, the proposed method achieves similar performance to the state of the art while being computationally more efficient. Moreover, fine-grained patch-level merging enhances model interpretability by revealing correlations between merged patches established by the model.

To our best knowledge, this is an early attempt to explore efficient video Transformers by progressively merging tokens into a more compact representation. It jointly eliminates redundant appearance cues and captures motion cues of input tokens to boost the robustness and discriminative power of the resulting representation. This token merge intuition is different from works dropping unimportant tokens, and it exhibits substantially better performance. We hope the proposed method can inspire future work in this direction.

2 RELATED WORK

This work is closely related to vision Transformer, video recognition, and efficient token pooling. This section briefly reviews recent advances in those fields and discusses differences between this work and previous ones.

2.1 Vision Transformer

Inspired by the success of the Transformer in natural language processing [9, 50], some studies have introduced the Transformer architecture into computer vision. Transformer has demonstrated promising performance in various image recognition tasks, including image classification [10, 19, 25, 48], object detection [5, 71], and image segmentation [52, 59, 69], especially with pretrained models [16]. Compared with the popular CNN models [63], the performance improvement brought by the Transformer is more distinct with massive training data and large-scale models [21, 32].

Recently, a series of works [2, 3, 8, 36, 40] have explored the application of the Transformer to the video tasks and obtained excellent performance. ViT [10] can be used to process features in temporal and spatial dimensions simultaneously via spatial-temporal positional embeddings. TimeSformer [3] studies five spatial-temporal self-attention schemes and finds that "divided attention," i.e., temporal and spatial attention separately applied within each block, leads to the best video classification performance. VideoSwin [36] adopts the window-based Swin Transformer designed for image recognition into video modeling to incorporate the inductive bias of locality. Motionformer [42] proposes to model the temporal correspondence and designs an efficient trajectory attention along the motion path. While offering superior results, the computation cost of video Transformers is intensive and scales up rapidly with the video frame resolution and video duration. Therefore, it is appealing to study more efficient video Transformers.

2.2 Efficient Video Recognition

Over the past few years, lots of efforts have been made to study efficient video recognition models. Some works propose lightweight 3D CNN architectures using network architecture search [27] or making an approximation with 2D convolutions [13, 72]. MoViNets [27] employs neural architecture search to generate efficient 3D CNN architectures within the designed video network search space. It also proposes a Stream Buffer technique that decouples memory from the video clip duration. X3D [13] progressively expands a small 2D image architecture into the spatial-temporal dimension to allow networks to scale or decrease computations under different computational budgets. Xu et al. [61] propose the SDS-CL framework, which aims to enhance semi-supervised skeleton-based action recognition by jointly contrasting spatial-squeezing features and temporal-squeezing features and learning spatiotemporal-decoupling representations. Similarly, PSP [60] was proposed to divide the skeleton-based action from the spatial granularity, making the task easier to learn. Xu et al. [62] propose a novel attention-consistency loss metric to enable the

temporal stream to concentrate on consistent discriminative regions with the spatial stream simultaneously. Moreover, SC-RNN [45] takes into account the spatial coherence among joints and temporal evolution among skeletons to predict human motion in spatiotemporal space.

Recently some works [17, 28, 58] have proposed to sample the salient frames from a long video for efficient inference. Scsampler [28] utilizes a lightweight subnet to assign a saliency score to each video clip and identifies the most salient temporal clips within a long video. Adaframe [58] adaptively selects relevant frames with reinforcement learning, augmented with a global memory, for fast video recognition. Tang et al. [46] propose an unsupervised method to retrieve the key video frames to automate key frame selection. Gowda et al. [17] propose to consider the relationship between selected frames and assign saliency scores jointly. Zhang et al. [66] propose a TokShift module to tackle temporal information interaction in video tasks. This module employs a novel partial temporal shift back-and-forth approach to model temporal relations. MM-ViT [6] utilizes the compressed video formats, such as MPEG4 and H.264, consisting of I-frames and P-frames, to separately represent the video appearance and motion information.

Motion information is a crucial part of video content underlying the frame sequence. For efficient motion and interaction recognition, Tang et al. [47] introduce the **Spatio-temporal Context Coherence (STCC)** constraint and the **Global Context Coherence (GCC)** constraint to capture relevant group activities in a single frame and improve group activity recognition accuracy. Zhao et al. [68] utilize a bidirectional Transformer with generative adversarial loss to effectively capture long-term human motion. Meanwhile, Shu et al. [44] propose considering the long-term inter-related dynamics among a group of individuals to enhance human interaction recognition in videos by using the relationships between people.

There are also some efficient video models that exploit the audio modality [55]; train on compressed video data such as H.264, HEVC, and so forth [6, 57]; or use variable mini-batch shapes [56]. HCMS [55] operates on a low-cost audio modality by default and uses three LSTMs to decide on the fly whether to use computationally expensive modalities, i.e., image or video, to supplement the recognition. Li et al. [29] propose to build a hierarchical structure for video to explore temporal dependencies with various scales. Zhu et al. [70] propose a **lightweight audio-visual saliency (LAVS)** model to utilize audio cues assisting the monomodal video model. Wu et al. [57] propose to train neural networks on more information-dense compressed video formats. Wu et al. [56] introduce a multi-grid method using variable mini-batch shapes with different spatial-temporal resolutions to accelerate model training. Nevertheless, these works focus on accelerating CNN-based video models. In this article, we focus on improving the efficiency of video Transformers, which require local patches and tokens as input.

2.3 Efficient Token Pooling

Vision Transformer divides the input visual signals into non-overlapping tokens and keeps the number of tokens fixed through the inference process [10]. Notably, some works [33, 39, 43] propose that there are a large number of redundant or irrelevant tokens in the image that can be pruned without compromising the recognition accuracy. In the image domain, DynamicViT [43] and EViT [33] propose to prune irrelevant tokens according to saliency scores. Marin et al. [39] further propose to prune tokens with clustering. Similarly, Zeng et al. [65] propose to cluster the background tokens to retain more details on human-centric features. Token Merger [15] proposes to adaptively merge similar contents and further boosts the model efficiency. To avoid global self-attention computation, GPVIT [64] groups features first, then calculates cross-attention between grouped tokens and image tokens. It reduces the computational complexity of self-attention from quadratic to linear. The STA [22] proposed by Huang et al. aggregates features into super pixels, then calculates self-attention in the super pixel space. Long et al. [37] propose to perform pruning

for retaining important information and merging for retaining contextual details simultaneously. This method fails to consider object shape and trajectory cues, making it unsuitable for video tasks, where motion and spatial cues are important. Recently, a similar work, ToMe [4], improves model efficiency by combining similar tokens through bipartite matching. Because the same object could appear in multiple video frames, the bipartite graph matching may degrade the efficiency of ToMe in merging features across frames. Our approach utilizes generated meta tokens to identify crucial visual cues in videos, leading to improved computational efficiency in spotting and merging similar tokens.

Recently, some works [20, 51] successfully introduced spatial-temporal token pooling into the video domain. Turbo training [20] proposes to train the Transformer on sparsely sampled visual tokens and achieves competitive performance on long-schedule video-language learning and long-video classification tasks. STTS [51] utilizes a lightweight scorer network to estimate the importance scores of each token and drop unimportant ones in both temporal and spatial dimensions. Wang et al. [54] improve the existing frame sample method and design a differentiable surrogate loss to make a video frame sampler end-to-end trained with the recognition backbone. Different from these works, our STTM does not drop tokens. It removes token redundancies by merging similar tokens. Our method allows to produce a compact feature representation and explore motion features in those merged tokens.

2.4 Differences with Existing Works

Compared with recent attention-based approaches, this work focuses on improving the efficiency of video Transformers by developing a novel approach to adaptively merge video tokens and preserve their motion cues. Ours is different from recent efficient video transformer works.

MM-ViT [6] employs distinct channels of compressed video signal to represent appearance and optic-flow motion cues. Our method directly adopts video frames as input. PSN [54] and VidTr [67] select salient frames with strong localized attention. Our method eliminates spatial-temporal redundancies by performing a more fine-grained patch-level merging. Park et al. [41] and STTS [51] sample K-centered salient video patches and drop others. Our approach adaptively identifies crucial semantic cues to guide the merging process, leading to a more information-efficient approach.

Some other methods, such as HAT-Net [35] and Video Swin Transformer [36], use a simple grid pooling layer to merge adjacent tokens. Those works fail to merge similar tokens across the spatial-temporal domains. Our STTM takes into account the distribution of informative patches to guide the merging process and thus shows better efficiency in spotting and eliminating redundancies. TCFormer [65] clusters background tokens to preserve task-related human-centric features specific to a single image. It disregards spatial layout cues. Different from Motionformer [42], this work further merges related tokens based on spatial and temporal correspondence. Our method hence reduces not only attention calculation but also the overall computation cost. In contrast to GPVIT [64] and STA [22], our method not only optimizes the self-attention quadratic complexity but also boosts overall model efficiency by merging similar feature tokens. Our method hence enjoys better efficiency when compared with those works. Besides, we maintain the essential spatial-temporal location of each merged feature as the motion representation, making it more suited for action recognition tasks.

3 METHODOLOGY

The proposed STTM merges input tokens of the Transformer block, leading to an updated architecture consisting of a Transformer as the backbone and several STTM modules incorporated into it. In this work, we use plain lowercase letters to denote vectors (e.g., $x \in \mathbb{R}^D$) and boldface uppercase letters for multi-dimensional matrices (e.g., $\boldsymbol{X} \in \mathbb{R}^{L \times D}$). Other letters are used for

scalars unless specified (e.g., $L, D \in \mathbb{R}$). Letters with subscripts i represent the ith elements in a sequence (e.g., x_i denotes the ith token in a sequence of tokens $\boldsymbol{X} \in \mathbb{R}^{L \times D}$).

3.1 Overview

To apply vision Transformers in video recognition, the input video is commonly split into non-overlapping spatial-temporal patches [3, 10]. Those patches are projected with linear layers into a sequence of features $\boldsymbol{X} \in \mathbb{R}^{L \times D}$, named tokens, where L is the number of patch embeddings and D is the embedding dimension. The number of tokens is proportional to the spatial and temporal resolution of the video, i.e., $L = T \times N$, where T and N denote the number of video frames and tokens in each frame, respectively, which is invariant during the model inference procedure.

Simply dividing the input frames into local patches leads to a considerable number of similar tokens and brings redundant computation costs. In order to reduce computational complexity, the proposed module aims to transform the original token sequence into a more compact representation.

Specifically, given a sequence of input tokens $\boldsymbol{X} \in \mathbb{R}^{L \times D}$ and embeddings of their spatial-temporal locations $\boldsymbol{E} \in \mathbb{R}^{L \times D}$, the STTM module aims to compress $\boldsymbol{X}$ and $\boldsymbol{E}$ into a more compact representation $\boldsymbol{Y} \in \mathbb{R}^{L' \times D}$ with $L' < L$. We denote the STTM as a function $\mathrm{F}(\cdot; \theta)$ that

$$\boldsymbol{Y} = \mathrm{F}\left(\boldsymbol{X}, \boldsymbol{E}; \theta\right), \boldsymbol{Y} \in \mathbb{R}^{\mathrm{L' \times D}}, \tag{1}$$

where θ denotes parameters to be learned, and $L' = L \times (1 - \rho)$ is the number of output tokens controlled by the merge ratio $0 < \rho < 1$. A larger ρ means a larger merge ratio and leads to fewer output tokens after merging.

To compute the $\boldsymbol{Y}$, STTM first aggregates $\boldsymbol{X}$ and $\boldsymbol{E}$ into appearance features $\boldsymbol{A} \in \mathbb{R}^{L' \times D}$ and motion features $\boldsymbol{P} \in \mathbb{R}^{L' \times D}$, respectively. It integrates appearance and motion features as the module output $\boldsymbol{Y}$. Specifically, for each pair of $\boldsymbol{A}_i$ and $\boldsymbol{P}_i$, we concatenate them with a linear layer to keep a consistent feature dimensionality:

$$\boldsymbol{Y}_i = \varphi([\boldsymbol{A}_i, \boldsymbol{P}_i]), 1 \leq i \leq L', \boldsymbol{Y}_i \in \mathbb{R}^D, \tag{2}$$

where $\varphi(\cdot)$ denotes a linear layer. Concatenation of every $\boldsymbol{Y}_i$ results in the fused token $\boldsymbol{Y}$. More details of the implementation to $F(\cdot; \theta)$ will be presented in the following part.

It is costly to directly merge L tokens across spatial and temporal dimensions because a large number of tokens lead to expensive global attention computation. As shown in Figure 2, STTM modules are inserted into multiple Transformer layers to perform token merging [15] with two stages to ensure better efficiency.

Spatial merging is first inserted into a shallow layer to compress the raw video tokens $\boldsymbol{X} \in \mathbb{R}^{N \times D}$ and positional embeddings $\boldsymbol{E} \in \mathbb{R}^{N \times D}$ within each frame into more compact features $\boldsymbol{Y} \in \mathbb{R}^{N' \times D}$ and its corresponding motion representation $\boldsymbol{P} \in \mathbb{R}^{N' \times D}$, where $N' = N \times (1 - \rho)$.

Spatial-temporal merging is hence applied to a deeper layer to perform token merging across the temporal dimension. As illustrated in Figure 2, the output token sequence from the previous Transformer block is regarded as its input token sequence $\boldsymbol{X}^* \in \mathbb{R}^{(T \times N') \times D}$. It takes the motion features $\boldsymbol{P}$ produced from the previous spatial merging STTM module as its input positional embeddings $\boldsymbol{E}^*$. The spatial-temporal merging decreases the token number from $(T \times N')$ to N^* with $N^* = (T \times N') \times (1 - \rho)$. The spatial-temporal merging also relies on the $\mathrm{F}(\cdot; \theta)$ and does not differentiate the spatial and temporal dimensions.

Note that the STTM in spatial merging and spatial-temporal merging shares the same parameters θ. The following parts proceed to introduce our implementation to produce appearance features and motion features in Equation (2).

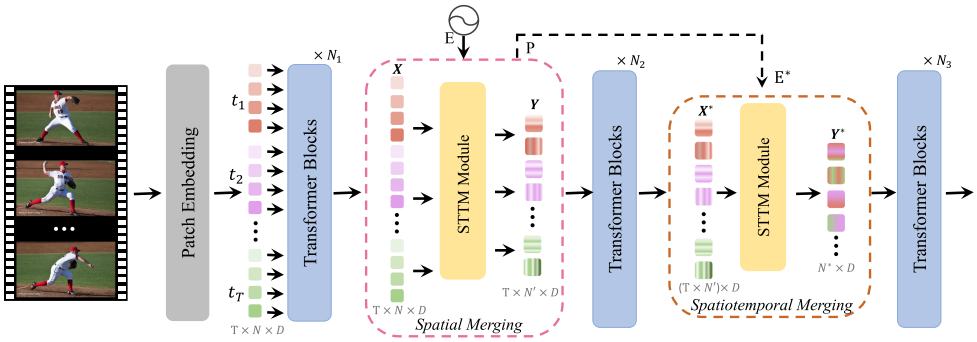

Fig. 2. **The framework of the proposed method.** It consists of a video Transformer backbone and two inserted STTM modules for spatial merging and spatial-temporal merging, respectively. The STTM module aims to shorten the input token sequence X into a more compact sequence Y while preserving crucial recognition information.

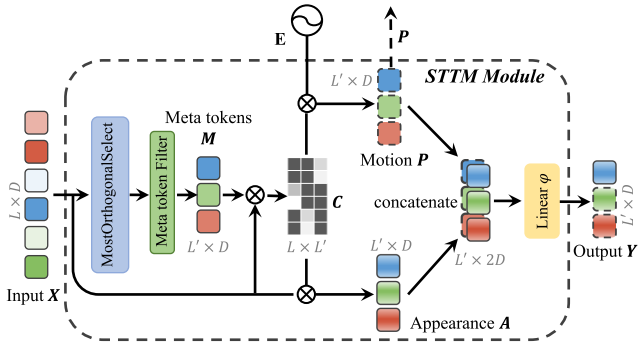

Fig. 3. **SSTM computation flowchart.** The module first extracts meta tokens from the input token sequence, then captures appearance and motion information based on meta tokens. It fuses motion features and appearance features as the final output.

3.2 STTM Module

$\boldsymbol{X}$ is the video token sequence produced from the Transformer blocks. $\boldsymbol{E}$ is a set of trainable parameters, namely positional embeddings proposed by the Transformer [50]. Note that positional embeddings are optimized end to end with the network. More details can be found from [50]. Instead of discarding or pruning tokens, STTM spots redundancies in token sequence $\boldsymbol{X}$ and aggregates them and their position embedding $\boldsymbol{E}$ into a more general representation.

The flowchart of the STTM module is illustrated in Figure 3. STTM involves three steps: (1) recognize the crucial object clues underlying the input token sequence as meta tokens $\boldsymbol{M}$ and calculate the correspondence scores $\boldsymbol{C}$ of each token with meta tokens, (2) aggregate the input token sequence $\boldsymbol{X}$ and position embedding $\boldsymbol{E}$ into appearance features $\boldsymbol{A}$ and motion features $\boldsymbol{P}$ based on $\boldsymbol{C}$, and (3) integrate appearance features and motion features according to Equation (2) to produce the module output $\boldsymbol{Y}$. The following part proceeds to present details in those steps.

3.2.1 Adaptive Meta Token Generation. Despite the complex cues conveyed by each video, it usually consists of a limited number of moving objects whose spatial locations and poses change over time. This makes it possible to learn a set of tokens to represent critical objects in the video.

To this end, we introduce a series of meta tokens $\boldsymbol{M} \in \mathbb{R}^{L' \times D}$ to represent crucial object clues underlying the input token sequence. In other words, the set of meta tokens $\boldsymbol{M}$ aims to reconstruct cues conveyed by the input token sequence $\boldsymbol{X}$, meanwhile minimizing its size as much as possible.

To learn $\boldsymbol{M}$, we follow [42] to iteratively select a set of tokens from the input token sequence, meanwhile ensuring each newly selected token is maximally orthogonal to the already selected ones, i.e.,

$$\boldsymbol{M} = \text{MostOrthogonalSelect}(\boldsymbol{X}, L'), \tag{3}$$

where $\boldsymbol{X}$ is the input token sequence and L' denotes the number of output meta tokens. Equation (3) is implemented by a greedy selection strategy. Specifically, we start by randomly selecting one token from the input token sequence. Then, we incrementally pick L' meta tokens that are maximally orthogonal to those already chosen. This greedy strategy is dynamic, since it selects a meta token from the current set of input tokens and favors tokens that are well separated. It strives to preserve as many object clues as possible by guaranteeing a high entropy in selected meta tokens. See [42] for more details.

Simply ensuring a high entropy in $\boldsymbol{M}$ may involve many noisy tokens on the cluttered backgrounds and degrade the performance of video recognition. To overcome this issue, we select $\alpha \times L'$ tokens, with $\alpha = 1.5$ using Equation (3), and filter noisy ones. This is implemented by computing attentiveness scores between them and the class token (CLS) following EViT [33] from the image domain. The top L' tokens with the highest attentive scores are kept as the generated meta tokens. For a more detailed implementation, please refer to [33]. This process ensures that only relevant meta tokens for the recognition task are retained, ensuring high computational efficiency.

Meta token generation introduces only marginal computational overheads. The computational complexity of meta token generation can be written as $\Omega(\text{MetaToken}) = L \times L' \times D + L' \times D$, where L is the number of input tokens, L' denotes the number of generated meta tokens, and D is the number of channels. The $L \times L' \times D$ and $L' \times D$ in the equation correspond to the computational overhead of Most Orthogonal Select [42] and token filter [33], respectively. For a typical STTM method built upon the ViT-B backbone, the computation of meta token generation only accounts for 0.0086% of the total computation.

The above strategy leads to a flexible meta token selection. It can identify meta tokens both for the entire video and for each frame and hence can be applied to both spatial merging and spatial-temporal merging. It also allows to perform meta token selection for a specific spatial-temporal region and hence is also applicable to video Transformers requiring locality. The following parts proceed to introduce the aggregation of the appearance feature and motion feature based on meta tokens.

3.2.2 Appearance Feature Aggregation. As the meta tokens $\boldsymbol{M}$ represent crucial object cues in the input tokens, they can be adopted to recognize tokens representing similar cues and aggregate them. To achieve this, we maintain a correspondence score $\boldsymbol{C} \in \mathbb{R}^{L \times L'}$ between input tokens $\boldsymbol{X}$ and meta tokens $\boldsymbol{M}$ to indicate which object the token belongs to. An encoder layer ϕ maps these tokens to l_2-normalized d-dimension vectors, which is adopted to calculate pair-wise similarity scores. We then convert the pair-wise similarity scores into non-negative values by applying the softmax function over the column and row, respectively. The softmax function applied on the column prevents a certain input token from dominating too many output tokens, which might lead to rank collapse. The softmax function applied on the row converts the pair-wise similarity scores into probabilities for the sake of the back-propagation of gradients. The computation of correspondence scores $\boldsymbol{C}$ can be denoted as

$$\boldsymbol{C} = \text{Softmax}(\phi(\boldsymbol{X})\phi(\boldsymbol{M})^T). \tag{4}$$

With the correspondence scores, the appearance feature $\boldsymbol{A}_i$ of ith the object can be represented by aggregating its corresponding tokens such that

$$\boldsymbol{A} = \boldsymbol{C}^T \boldsymbol{X}, \tag{5}$$

where $\boldsymbol{A} \in \mathbb{R}^{L' \times D}$ denotes the appearance feature of output tokens and $\boldsymbol{X}$ is the input token sequence.

3.2.3 Motion Feature Aggregation. Fused appearance features by Equation (5) do not convey motion cues of objects in the video. As motion cues are important for video recognition, STTM explicitly aggregates position embeddings $\boldsymbol{E}$ as a complementary motion feature.

In the video Transformer, the tokens are added with learnable positional embedding before being fed into the network. The positional embedding $\boldsymbol{E} \in \mathbb{R}^{L \times D}$ is a sequence of high-dimension vectors corresponding to a 3D spatial-temporal location. Positional embeddings hence encode the spatial and temporal cues of tokens and can be processed by the Transformer to extract meaningful cues.

Previous research [10, 50] has shown that each dimension of positional encoding exhibits a sinusoidal structure and quasi-orthogonal nature, ensuring that their linear combinations do not interfere with each other. This property makes it possible to fuse multiple positional embeddings with a simple linear combination. We hence linearly combine position embeddings of merged tokens to represent their spatial and temporal cues. For instance, the combination of multiple spatial locations inside each frame could represent shape cues of an object. The combination of multiple spatial-temporal locations represents the motion of a certain object. We hence define the motion feature P as the linear combination of positional embeddings. This effectively preserves shape and motion information in the original token sequence without introducing additional parameters and substantial computational costs.

Specifically, the motion feature is computed by fusing positional embeddings $\boldsymbol{E}$ of tokens according to the correspondence scores, which can be written as

$$\boldsymbol{P} = \boldsymbol{C}^T \boldsymbol{E}, \tag{6}$$

where $\boldsymbol{P}_i$ denotes the motion feature corresponding to the appearance feature $\boldsymbol{A}_i$. $\boldsymbol{P}_i$ and $\boldsymbol{A}_i$ are fused with Equation (2) to produce the fused token $\boldsymbol{Y}_i$.

After the first STTM, each merged token no longer corresponds to a specific frame location, e.g., its motion feature P merges multiple position embedings and could represent shape cues. Thus, in the subsequent STTM module, we use the motion feature P from the previous module as the position embedding. The subsequent STTM module continues linear combinations to represent more intricate position and motion cues, creating a more complicated motion feature. The sinusoidal structure and quasi-orthogonal nature of positional embeddings ensure that their linear combinations do not interfere with each other.

As illustrated in Figure 2, two STTM modules are inserted into the Transformer to perform spatial merging and spatial-temporal merging, respectively. The spatial merging and spatial-temporal merging follow the same procedure but adopt different inputs. Details of their insert locations and performance will be given in Section 4.

4 EXPERIMENT

To demonstrate the effectiveness of STTM, we apply it to commonly used video Transformer backbones on two large-scale video recognition datasets. We introduce the datasets in Section 4.1, demonstrate the experimental setup in Section 4.2, present ablation studies and analysis in Section 4.3, and finally exhibit experimental comparisons with recent works in Section 4.4.

4.1 Datasets

Kinetics-400 [26] is a widely used large-scale dataset for action recognition, consisting of 400 human action classes with at least 400 video clips for each class. The video clips in Kinetics are collected from YouTube. They are 10 seconds long, and the frame rate is 25 frames per second. Kinetics-400 only releases video URL. Because some videos may be deleted from YouTube, its size may vary slightly. Our training and validation dataset contains approximately 234K and 19K video clips, respectively.

Something-Something V2 (SSv2) [18] is a challenging action recognition dataset containing more than 200K videos across 174 classes, with 169K in the training set and 25K in the validation set. Video durations range from 2 to 6 seconds. Compared to Kinetics, the background and objects in SSv2 remain consistent across different classes, which posts higher requirements for learning temporal features.

Evaluation metrics. We report the top-1 and top-5 accuracy on the validation set to measure the performance. For the computational cost, we utilize hardware-independent GFLOPs to measure the computational complexity.

4.2 Experimental Setup

Backbones. We apply STTM to commonly used video Transformer backbones. For a fair comparison, all the models are initialized with ImageNet-1K pre-trained parameters. **ViT** [10] is chosen as a baseline Transformer backbone with default settings to evaluate STTM on the original Transformer. Besides learnable spatial positional embeddings in [10], we extend ViT with temporal positional embedding to adapt to the video domain following [41]. **TimeSformer** [3] improves the global attention computation with divided attention, which separately applies temporal attention and spatial attention within each block of the network. TimeSformer with eight frames as an input is adopted in our experiments.

Implementation details. We insert our STTM into different layers of the video Transformer and fine-tune the pre-trained models. For ViT [10] and TimeSformer [3] with 12 Transformer layers in total, the proposed modules are inserted at the sixth and ninth layers to perform spatial and spatiotemporal merging, respectively.

We choose ViT-Base (ViT-B, $R = 12, N_H = 12, d = 768$) as our base architecture, where R is the number of Transformer layers, with a self-attention block of N_H heads and hidden dimension d. Parameters in STTM modules are randomly initialized before fine-tuning. Merge ratios are set to 0.5 by default. The input video resolution is 224×224 for both training and testing unless otherwise specified.

For training strategies and optimization options, we follow most of the training techniques used in the original TimeSformer [3]. The models are trained with an AdamW optimizer [38] for 30 epochs with an initial learning rate of 0.005, a weight decay of 0.0001, a scheduler that uses linear learning rate decay, and a momentum of 0.9. The backbones are initialized with weights pre-trained on ImageNet-1K. For inference, we follow the common practice that processes multiple views of a video and average per-view logits to obtain the final result. Specifically, we sample a fixed-length clip from the video and spatially crop three different spatial views from the clip.

4.3 Ablation Studies

(1) Effectiveness of appearance and motion features. STTM aggregates input tokens and positional embeddings to appearance features and motion features. It hence fuses those features as the final output. To validate the effectiveness of the motion features, we compare the SSv2 recognition performance of two variants of STTM with or without motion features. STTM without motion

Table 1. Comparison of Two Variants of STTM with or without Motion Features

Used features	GFLOPs	Top-1. acc.	Top-5. acc.
Only appearance embedding	363	56.6	84.2
+ motion embedding	368	58.4	85.4

With a minor computation overhead, the motion features boost the SSv2 top-1 accuracy by 1.5%.

Table 2. Comparison of the Proposed Method with or without Meta Token Filtering

Model	Meta filtering	GFLOPs	Top-1. acc.	Top-5. acc.
ViT-S+STTM		98	54.1	82.3
ViT-S+STTM	✓	99	55.6	82.7
ViT-B+STTM		344	54.6	82.5
ViT-B+STTM	✓	345	56.5	83.9
TimeSformer+STTM		367	57.5	84.6
TimeSformer+STTM	✓	368	58.4	85.4

We generate 1.5× meta tokens and filter trivial ones via comparing similarity with the CLS token. The merge ratios are set to 0.5.

Table 3. Performance Comparison with Different Merge Ratios

Merge ratio	GFLOPs	Top-1. acc.	Top-5. acc.
0.8	332	54.8	82.6
0.6	349	56.2	84.0
0.5	368	58.4	85.4
0.4	412	58.9	85.9
0.2	459	59.2	86.0

The experiments are conducted on the SSv2 dataset with ImageNet-1K pre-trained parameters and fine-tuned for 15 epochs.

features is implemented by removing the motion feature and fusion layer and only using the appearance features. The modules are inserted at the sixth and ninth layers of TimeSformer. As shown in Table 1, the motion features boost the SSv2 top-1 accuracy by 1.5% with only 5 GFLOPs overhead. It shows that simple linear combination with Equation (6) is effective in learning motion features and boosts the action recognition accuracy.

(2) Effectiveness of filtering meta tokens. STTM filters generated meta tokens by comparing them with classification tokens to alleviate the disturbance on cluttered backgrounds. This experiment tests the effectiveness of this filtering method. Experiments are conducted on ViT-S, ViT-B, and TimeSformer, with merging ratios of 0.5. As shown in Table 2, meta token filtering brings significant performance improvements for all three Transformer backbones. It also indicates that token merging brings marginal computation overheads. It shows the effectiveness of token filtering; i.e., the video contains tokens not helpful for recognition, and filtering them effectively boosts the recognition accuracy.

(3) Effect of merging ratios. We test the model performances with different merge ratios and summarize the results in Table 3. As the number of meta tokens and output tokens is proportional to

Table 4. Ablation Study on Insertion Location of the STTM

Insert Layers	GFLOPs	Top-1. acc.	Top-5. acc.
[3]	330	55.2	83.0
[6]	418	58.2	85.3
[9]	505	58.9	85.6
[3,6]	271	54.8	82.8
[6,9]	368	58.4	85.4

We compare the SSv2 recognition accuracy by inserting one or two STTM modules into different layers of the TimeSformer model.

Table 5. Effectiveness of the Two-stage Merging Strategy

Merge type	GFLOPs	Top-1. acc.	Top-5. acc.
Spatial	389	57.7	84.6
Global	459	58.4	85.5
Two-stage	368	58.4	85.4

"Spatial" presents the result of performing spatial merging within every frame. "Global" presents the result of directly performing spatial-temporal merging without spatial merging. "Two-stage" represents our method.

the merge ratio, a larger ratio decreases the token number and the computation cost. Experimental results show that the model performance reasonably decreases as we increase the merging ratio. Ratio = 0.5 substantially decreases the GFLOPs from 459 of ratio = 0.2 to 368 and meanwhile only degrades the top-5 accuracy by 0.7%. Therefore, setting the merge ratio as 0.4 or 0.5 could be an appropriate tradeoff between accuracy and complexity. Note that the merge ratio can be flexibly controlled for different scenarios according to the budget of computational cost.

(4) Insertion location of STTM module. This part explores the impact of different insertion locations of STTM on SSv2. We insert STTM at different layers with the same merge ratio of 0.5 and summarize the results in Table 4. For the case of inserting one STTM, we apply the spatial merging STTM to avoid the high computation of the global attention computation. As shown in Table 4, inserting spatial merging STTM at the shallow third layer substantially boosts the efficiency but is more harmful for the model performance than inserting it at the ninth layer. Inserting it at the sixth layer achieves a reasonable tradeoff between efficiency and accuracy. This is because tokens generated by a deeper layer are more discriminative to semantics. They are more suited for STTM, which spots and merges tokens depicting similar semantics. Table 4 also indicates that inserting the spatial-temporal merging STTM further boosts the efficiency without substantially decreasing the accuracy. For example, inserting two STTMs at the sixth and ninth layers reduces the GFLOPs to 368 and only brings a minor top-1 accuracy degradation of 0.1% compared to inserting one STTM at the sixth layer.

(5) Effectiveness of two-stage merging method. To test the effectiveness of our two-stage merging method, this part compares three merging strategies, i.e., spatial merging within each frame, directly merging spatial-temporal tokens, and our two-stage merging. Because video consists of many frames, directly merging tokens across all frames requires costly global attention calculations. To avoid this, we split the STTM into two types: spatial merging and spatiotemporal merging. The Spatial Merging module merges tokens within each frame by identifying important object

Table 6. Comparison of Three Baseline Models with or without STTM

Backbone	STTM	GFLOPs	Params	Frames	K400		SSv2	
					Top-1	Top-5	Top-1	Top-5
ViT-S [10]		170	21.8M	8	67.4	87.2	56.3	83.3
	✓	99	22.7M	8	66.1	86.6	55.6	82.7
	✓	164	22.7M	13	**68.6**	**89.0**	**57.5**	**84.9**
ViT-B [10]		540	86.1M	8	75.0	91.7	57.3	84.4
	✓	345	89.6M	8	73.3	90.5	56.5	83.9
	✓	537	89.6M	11	**76.1**	**92.6**	**59.3**	**85.2**
TimeSformer [3]		590	121.6M	8	75.8	92.7	59.5	85.7
	✓	368	107.4M	8	74.4	91.9	58.4	85.4
	✓	572	107.4M	11	**76.4**	**93.3**	**61.1**	**86.2**

We report classification top-1 and top-5 accuracy on Kinetics-400 and SSv2 datasets. We also increase the number of frames of the input video from 8 to 13 to make computational complexity consistent with the baseline model. Models are with the ImageNet-1k pretraining backbone.

cues and results in a more compact frame feature representation. This process doesn't alter original temporal cues and keeps frames T and feature dimension D unchanged. However, the first STTM module largely decreases the number of tokens and reduces global self-attention calculation costs by 75%, hence ensuring efficient global feature merging in the Spatiotemporal Merging stage.

Experimental results with TimeSformer on the SSv2 dataset are shown in Table 5. It can be observed that "Global" is more expensive than "Spatial" because it performs attention calculations on the input tokens and suffers from quadratic complexity. Among those three methods, our two-stage merging strategy achieves the best tradeoff between accuracy and efficiency. It achieves comparable accuracy with the expensive global merging and meanwhile exhibits better efficiency than the spatial merging. We hence conclude that our two-stage merging method is reasonable and effective.

4.4 Comparisons with Recent Works

To test the performance of STTM in video action recognition, we insert it into three baseline video Transformers, i.e., ViT-S [10], ViT-B [10], and TimeSformer [3], and report their top-1 and top-5 classification accuracy on Kinectics-400 and SSv2, respectively. As shown in Table 6, our method significantly reduces the computational cost of all video Transformers, meanwhile maintaining a comparable performance. For example, STTM reduces the GFLOPS for ViT-B by 36% while only decreasing the top-1 and top-5 accuracy by 0.9% and 0.5% on SSv2, respectively. Similar observation can be made on the TimeSformer and ViT-S; i.e., it decreases the GFLOPs by 38% and degrades the top-1 and top-5 accuracy by 1.1% and 0.3% for TimeSformer and decreases the GFLOPs by 42% and degrades the top-1 and top-5 accuracy by 0.7% and 0.6% for ViT-S.

We further increase the number of input frames from 8 to 11 to make the computation complexity of the proposed method equal to the one of the baseline. Under the setting of similar computation cost, the proposed method consistently improves the model performance. For example, on ViT-B [10], STTM boosts the top-1 accuracy by 1.1% and 1.7% on Kinetics-400 and SSv2, respectively, with fewer GFLOPs. When the input frame is expanded to 13 frames for ViT-S [10], the proposed method can improve the top-1 accuracy on Kinetics-400 and SSv2 by 1.2% and 1.2%, respectively.

Additionally, STTM resulted in only a marginal increase in the model parameters. For instance, the ViT-S and ViT-B models witnessed a mere 4.1% addition of parameters. On the other hand, for

Table 7. Comparison between the Proposed Method and Some Recent Works on Kinetics-400

Model	Pre-train	GFLOPs × views	Top-1. acc.
R(2+1)D [49]	-	1,750	72.0
I3D [53]	IN-1K	1,110	71.0
bLVNet [11]	Kinetics	560	72.0
Oct-I3D+NL [7]	-	840	75.7
SlowFast [14]	-	1,970	75.6
AC Loss 4×16, R50 [62]	-	1,026	75.7
TEA [31]	IN-1K	2,100	76.1
VTN-R101 [40]	IN-21K	1,989	72.1
TSM [34]	IN-1K	650	74.7
ViT-B [10]	IN-1K	540	75.0
TimeSformer [3]	IN-1K	590	75.8
K-centered-ViT [41]	IN-1K	540	75.7
TAda2D [23]	IN-1K	2,580	77.4
X3D [13]	-	5,823	80.4
En-VidTr-M [30]	IN-21K	6,600	79.7
Motionformer [42]	IN-21K	11,085	79.7
TokShift [66]	IN-21K	41,928	79.8
Ours+ViT-B	IN-1K	537	76.1
Ours+TimeSformer	IN-1K	572	76.4
Ours+ViT-B	IN-21K	2,148	80.2
Ours+TimeSformer	IN-21K	2,288	**80.8**

"-" denotes that the models are trained from scratch.

Table 8. Comparison between the Proposed Method and Some Recent Works on Something-Something-V2

Model	Pre-train	GFLOPs × views	Top-1. acc.
TimeSformer-L [3]	IN-21K	7,140	62.3
TSM [34]	K400	370	63.3
STM [24]	IN-1K	2,000	64.2
SIFAR-L [12]	K400	576	64.2
blVNet [11]	IN-1K	3,860	65.2
ViViT-L [2]	K400	47,600	65.4
ViT-B [10]	IN-21K	540	61.2
K-centered-TimeSformer [41]	IN-1K	7,140	63.8
Ours+ViT-B	K400	537	65.5
Ours+TimeSformer	K400	572	66.2

the TimeSformer model, the merging process effectively reduces the number of tokens, enabling global self-attention calculations concurrently on time and space dimensions in deep layers. This in turn reduces the model's parameter count by 11.7%. Consequently, the proposed method produces an effective and lightweight video Transformer model.

We compare our method with some recent works on Kinetics-400 in Table 7 and SSv2 in Table 8. Compared works include efficient CNN-based action recognition works [7, 11, 12, 14, 24, 31, 34,

40, 49, 53] and recent efficient video Transformer-based methods [2, 3, 10, 40–42]. To conduct experiments on the SSv2 dataset, we followed the common experimental setup by fine-tuning the model pre-trained on Kinetics-400.

It can be observed that Transformer-based methods generally outperform CNN-based methods. Our work is incorporated with video Transformer backbones [3, 10] and further enhances their accuracy and efficiency for both datasets. Specifically, it achieved a top-1 accuracy that is +0.8% better than that of ViViT-L, with a computational overhead of only 1.2% of its GFLOPs on SSv2. The STTM also outperforms the recent token sampling method [41] by +0.4% and +2.4% of top-1 accuracy for Kinetics-400 and SSv2, respectively, with less computational cost. We hence conclude that STTM is effective in boosting the efficiency and accuracy of video Transformer backbones and across multiple datasets.

4.5 Visualization

To show the effectiveness of STTM, we visualize the token merging results in Figure 4, where the fused tokens, attention heatmaps of TimeSformer [3], and our method are illustrated, respectively. Meta tokens help combine related tokens. Using the Most Orthogonal Select strategy, different meta tokens usually represent various visual cues in the picture. In the first row of each example, we select several dominant objects and show their associated meta tokens. We use colors to denote different meta tokens. The second row shows the STTM merging results of those selected meta tokens. The same colors appearing in different locations across frames are fused into a new token. It can be seen that the proposed method efficiently reduces redundancy by combining tokens that describe similar contents. Note that STTM does not mandatorily fuse tokens on the same object into a single one. Each object hence corresponds to multiple meta tokens depicting different local parts. The third and fourth rows are attention heatmaps of the TimeSformer [3] and TimeSformer + STTM module, respectively. To better highlight the differences between those two heatmaps, we manually mark some regions using a red box, where the proposed method notably outperforms the baseline method.

The visualization of merging results illustrates that STTM effectively recognizes tokens depicting the same object and integrates them. The results on different frames show that token merging is stable and consistent across frames. STTM is also capable of distinguishing different objects with similar semantics. For example, the first example involves two persons. Although conveying similar semantics, they are depicted by different tokens. The merged tokens are also semantically reasonable. For example, different tokens are merged to depict different body parts of the person, including the head, upper body, and lower body.

The comparison of attended regions with the original TimeSformer [3] illustrates a better performance achieved by TimeSformer+STTM. STTM makes the attended area more concentrated on the moving objects. It implies that STTM enforces the model to focus on critical tokens for the action recognition and depresses noisy tokens on cluttered backgrounds. In the first example from the Kinetics-400 dataset, a picture of two people performing Tai chi on a lawn, the baseline model struggles with the presence of the background tree and footpath, as it allocates high attention to those objects. STTM successfully eliminates the noise from the background and enables the model to accurately focus on the moving individuals. In the second example, which depicts the action of "Picking up a spoon," the model demonstrates improved performance with STTM, as it better captures the movements of the hand and the small-sized spoon. Lastly, in the third example, the baseline model gives undue attention to the unimportant part of the background sky, whereas STTM enables the model to concentrate more on the hurdler and the hurdle. The visualization results demonstrate that STTM can enhance the model's focus on the critical regions, leading to higher efficiency.

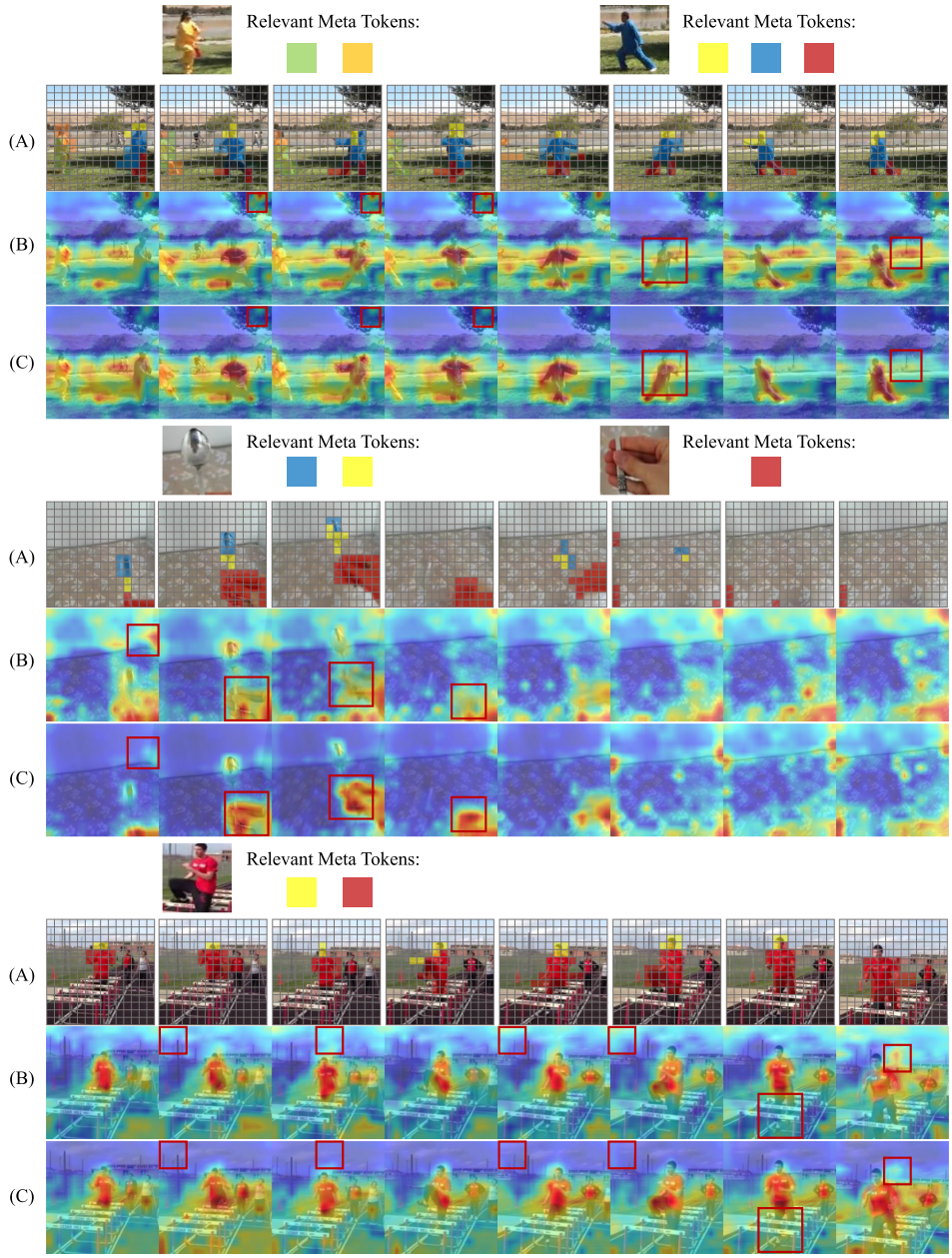

Fig. 4. **Visualization of token merging results.** In each of the three examples, the first row shows several object samples and tokens depicting them, where different colors denote different tokens. The second row (A) is the STTM merging results, where merged tokens are marked with the same color. The (B) and (C) rows are the attention heatmaps of the TimeSformer [3] and TimeSformer+STTM module, respectively. We highlight the attended region of interest with a red box. The visualization is implemented with an attention roll-out method [1].

5 CONCLUSION

This work proposes a novel STTM module to compress tokens of video Transformers into a more compact representation by exploring both appearance and motion cues in videos. STTM extracts appearance and motion features from the original token sequence and merges tokens depicting similar semantics or objects into a more general one. We use the linear combination of positional embeddings as the motion features. Meta tokens are generated to guide the merging of appearance and motion features. STTM outputs a compact set of tokens and hence substantially decreases the number of tokens that need to be processed by each Transformer block and boosts the efficiency. STTM can be applied to different layers of various video Transformers. Visualization to the token merging results and extensive experiments on action recognition demonstrate its promising performance.

REFERENCES

[1] Samira Abnar and Willem Zuidema. 2020. Quantifying attention flow in transformers. *arXiv preprint arXiv:2005.00928* (2020).

[2] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. 2021. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6836–6846.

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. 2021. Is space-time attention all you need for video understanding? In *International Conference on Machine Learning*, PMLR, 813–824.

[4] Daniel Bolya, Cheng-Yang Fu, Xiaoliang Dai, Peizhao Zhang, Christoph Feichtenhofer, and Judy Hoffman. 2022. Token merging: Your vit but faster. *arXiv preprint arXiv:2210.09461* (2022).

[5] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers. In *European Conference on Computer Vision*. Springer, 213–229.

[6] Jiawei Chen and Chiu Man Ho. 2022. MM-ViT: Multi-modal video transformer for compressed video action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 1910–1921.

[7] Yunpeng Chen, Haoqi Fan, Bing Xu, Zhicheng Yan, Yannis Kalantidis, Marcus Rohrbach, Shuicheng Yan, and Jiashi Feng. 2019. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3435–3444.

[8] Zhen Chen, Ming Yang, and Shiliang Zhang. 2023. Complementary coarse-to-fine matching for video object segmentation. *ACM Transactions on Multimedia Computing, Communications and Applications (TOMM'23)* 19, 6 (2023), 1–21.

[9] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Uszkoreit Jakob, and Houlsby Neil. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[11] Quanfu Fan, Chun-Fu Richard Chen, Hilde Kuehne, Marco Pistoia, and David Cox. 2019. More is less: learning efficient video representations by big-little network and depthwise temporal aggregation. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. 2264–2273.

[12] Quanfu Fan and Rameswar Panda. 2021. An image classifier can suffice for video understanding. *arXiv preprint arXiv:2106.14104* 2, (2021).

[13] Christoph Feichtenhofer. 2020. X3D: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 203–213.

[14] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. 2019. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6202–6211.

[15] Zhanzhou Feng and Shiliang Zhang. 2023. Efficient vision transformer via token merger. *IEEE Transactions on Image Processing: A Publication of the IEEE Signal Processing Society* 32, (2023), 4156–4169.

[16] Zhanzhou Feng and Shiliang Zhang. 2023. Evolved part masking for self-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10386–10395.

[17] Shreyank N. Gowda, Marcus Rohrbach, and Laura Sevilla-Lara. 2021. Smart frame selection for action recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 35. 1451–1459.

[18] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, Florian Hoppe, Christian Thurau, Ingo Bax, and Roland Memisevic. 2017. The "something something" video database for learning and evaluating visual common sense. In *Proceedings of the IEEE International Conference on Computer Vision*. 5842–5850.

[19] Benjamin Graham, Alaaeldin El-Nouby, Hugo Touvron, Pierre Stock, Armand Joulin, Hervé Jégou, and Matthijs Douze. 2021. LeViT: A vision transformer in ConvNet's clothing for faster inference. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 12259–12269.

[20] Tengda Han, Weidi Xie, and Andrew Zisserman. 2022. Turbo training with token dropout. *arXiv preprint arXiv:2210.04889* (2022).

[21] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2021. Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377* (2021).

[22] Huaibo Huang, Xiaoqiang Zhou, Jie Cao, Ran He, and Tieniu Tan. 2023. Vision transformer with super token sampling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 22690–22699.

[23] Ziyuan Huang, Shiwei Zhang, Liang Pan, Zhiwu Qing, Mingqian Tang, Ziwei Liu, and Marcelo H. Ang Jr. 2022. TAda! Temporally-adaptive convolutions for video understanding. In *International Conference on Learning Representations*.

[24] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. 2019. STM: Spatiotemporal and motion encoding for action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2000–2009.

[25] Zi-Hang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Yujun Shi, Xiaojie Jin, Anran Wang, and Jiashi Feng. 2021. All tokens matter: Token labeling for training better vision transformers. *Advances in Neural Information Processing Systems* 34, (2021), 18590–18602.

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

[27] Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. 2021. Movinets: Mobile video networks for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16020–16030.

[28] Bruno Korbar, Du Tran, and Lorenzo Torresani. 2019. SCSampler: Sampling salient clips from video for efficient action recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 6232–6242.

[29] Wenxu Li, Gang Pan, Chen Wang, Zhen Xing, and Zhenjun Han. 2022. From coarse to fine: Hierarchical structure-aware video summarization. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 18, 1s (2022), 1–16.

[30] Xinyu Li, Yanyi Zhang, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. 2021. VidTr: Video transformer without convolutions. *arXiv e-prints* (2021), arXiv–2104.

[31] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. 2020. TEA: Temporal excitation and aggregation for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 909–918.

[32] Zhaowen Li, Zhiyang Chen, Fan Yang, Wei Li, Yousong Zhu, Chaoyang Zhao, Rui Deng, Liwei Wu, Rui Zhao, Ming Tang, and Wang Jinqiao. 2021. Mst: Masked self-supervised transformer for visual representation. *Advances in Neural Information Processing Systems* 34, (2021), 13165–13176.

[33] Youwei Liang, Chongjian Ge, Zhan Tong, Yibing Song, Jue Wang, and Pengtao Xie. 2022. Not all patches are what you need: Expediting vision transformers via token reorganizations. *arXiv preprint arXiv:2202.07800* (2022).

[34] Ji Lin, Chuang Gan, and Song Han. 2019. TSM: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7083–7093.

[35] Yun Liu, Guolei Sun, Yu Qiu, Le Zhang, Ajad Chhatkuli, and Luc Van Gool. 2021. Transformer in convolutional neural networks. *CoRR* abs/2106.03180 (2021). arXiv:2106.03180 https://arxiv.org/abs/2106.03180

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

[37] Sifan Long, Zhen Zhao, Jimin Pi, Shengsheng Wang, and Jingdong Wang. 2023. Beyond attentive tokens: Incorporating token importance and diversity for efficient vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10334–10343.

[38] Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. arXiv preprint arXiv:1711.05101 (2017).

[39] Dmitrii Marin, Jen-Hao Rick Chang, Anurag Ranjan, Anish Prabhu, Mohammad Rastegari, and Oncel Tuzel. 2021. Token pooling in vision transformers. *arXiv preprint arXiv:2110.03860* (2021).

[40] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. 2021. Video transformer network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 3163–3172.

[41] Seong Hyeon Park, Jihoon Tack, Byeongho Heo, Jung-Woo Ha, and Jinwoo Shin. 2022. K-centered patch sampling for efficient video recognition. In *European Conference on Computer Vision*. Springer, 160–176.

[42] Mandela Patrick, Dylan Campbell, Yuki Asano, Ishan Misra, Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F. Henriques. 2021. Keeping your eye on the ball: Trajectory attention in video transformers. *Advances in Neural Information Processing Systems* 34 (2021), 12493–12506.

[43] Yongming Rao, Wenliang Zhao, Benlin Liu, Jiwen Lu, Jie Zhou, and Cho-Jui Hsieh. 2021. DynamicViT: Efficient vision transformers with dynamic token sparsification. *Advances in Neural Information Processing Systems* 34 (2021), 13937–13949.

[44] Xiangbo Shu, Jinhui Tang, Guo-Jun Qi, Wei Liu, and Jian Yang. 2019. Hierarchical long short-term concurrent memory for human interaction recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, 3 (2019), 1110–1118.

[45] Xiangbo Shu, Liyan Zhang, Guo-Jun Qi, Wei Liu, and Jinhui Tang. 2021. Spatiotemporal co-attention recurrent neural networks for human-skeleton motion prediction. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 6 (2021), 3300–3315.

[46] Hao Tang, Lei Ding, Songsong Wu, Bin Ren, Nicu Sebe, and Paolo Rota. 2023. Deep unsupervised key frame extraction for efficient video classification. *ACM Transactions on Multimedia Computing, Communications and Applications (TOMM)* 19, 3 (2023), 1–17.

[47] Jinhui Tang, Xiangbo Shu, Rui Yan, and Liyan Zhang. 2019. Coherence constrained graph LSTM for group activity recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 2 (2019), 636–647.

[48] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. 2021. Training data-efficient image transformers & distillation through attention. In *International Conference on Machine Learning*. PMLR, 10347–10357.

[49] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. 2018. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6450–6459.

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, ?ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 6000–6010.

[51] Junke Wang, Xitong Yang, Hengduo Li, Li Liu, Zuxuan Wu, and Yu-Gang Jiang. 2022. Efficient video transformers with spatial-temporal token selection. In *European Conference on Computer Vision*. Springer, 69–86.

[52] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. 2021. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 568–578.

[53] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. 2018. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7794–7803.

[54] Xiaohan Wang, Linchao Zhu, Fei Wu, and Yi Yang. 2023. A differentiable parallel sampler for efficient video classification. *ACM Transactions on Multimedia Computing, Communications and Applications (TOMM)* 19, 3 (2023), 1–18.

[55] Zejia Weng, Zuxuan Wu, Hengduo Li, Jingjing Chen, and Yu-Gang Jiang. 2023. HCMS: Hierarchical and conditional modality selection for efficient video recognition. *ACM Transactions on Multimedia Computing, Communications and Applications* 20, 2 (2023), 1–18.

[56] Chao-Yuan Wu, Ross Girshick, Kaiming He, Christoph Feichtenhofer, and Philipp Krahenbuhl. 2020. A multigrid method for efficiently training video models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 153–162.

[57] Chao-Yuan Wu, Manzil Zaheer, Hexiang Hu, R. Manmatha, Alexander J. Smola, and Philipp Krähenbühl. 2018. Compressed video action recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6026–6035.

[58] Zuxuan Wu, Caiming Xiong, Chih-Yao Ma, Richard Socher, and Larry S. Davis. 2019. Adaframe: Adaptive frame selection for fast video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 1278–1287.

[59] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems* 34, (2021), 12077–12090.

[60] Binqian Xu and Xiangbo Shu. 2023. Pyramid self-attention polymerization learning for semi-supervised skeleton-based action recognition. *arXiv preprint arXiv:2302.02327* (2023).

[61] Binqian Xu, Xiangbo Shu, Jiachao Zhang, Guangzhao Dai, and Yan Song. 2023. Spatiotemporal decouple-and-squeeze contrastive learning for semisupervised skeleton-based action recognition. *IEEE Transactions on Neural Networks and Learning Systems* (2023), 1–14. DOI: https://doi.org/10.1109/TNNLS.2023.3247103

[62] Haotian Xu, Xiaobo Jin, Qiufeng Wang, Amir Hussain, and Kaizhu Huang. 2022. Exploiting attention-consistency loss for spatial-temporal stream action recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 18, 2s (2022), 1–15.

[63] Ruihan Xu, Haokui Zhang, Wenze Hu, Shiliang Zhang, and Xiaoyu Wang. 2023. ParCNetV2: Oversized kernel with enhanced attention. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5752–5762.

[64] Chenhongyi Yang, Jiarui Xu, Shalini De Mello, Elliot J. Crowley, and Xiaolong Wang. 2022. GPViT: A high resolution non-hierarchical vision transformer with group propagation. *arXiv preprint arXiv:2212.06795* (2022).

[65] Wang Zeng, Sheng Jin, Wentao Liu, Chen Qian, Ping Luo, Wanli Ouyang, and Xiaogang Wang. 2022. Not all tokens are equal: Human-centric visual analysis via token clustering transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 11101–11111.

[66] Hao Zhang, Yanbin Hao, and Chong-Wah Ngo. 2021. Token shift transformer for video classification. In *Proceedings of the 29th ACM International Conference on Multimedia*. 917–925.

[67] Yanyi Zhang, Xinyu Li, Chunhui Liu, Bing Shuai, Yi Zhu, Biagio Brattoli, Hao Chen, Ivan Marsic, and Joseph Tighe. 2021. VidTr: Video transformer without convolutions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV '21)*. 13577–13587.

[68] Mengyi Zhao, Hao Tang, Pan Xie, Shuling Dai, Nicu Sebe, and Wei Wang. 2023. Bidirectional transformer gan for long-term human motion prediction. *ACM Transactions on Multimedia Computing, Communications and Applications* 19, 5 (2023), 1–19.

[69] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, and Zhang Li. 2021. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 6881–6890.

[70] Dandan Zhu, Xuan Shao, Qiangqiang Zhou, Xiongkuo Min, Guangtao Zhai, and Xiaokang Yang. 2023. A novel lightweight audio-visual saliency model for videos. *ACM Transactions on Multimedia Computing, Communications and Applications (TOMM)* 19, 4 (2023), 1–22.

[71] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. 2020. Deformable DETR: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159* (2020).

[72] Mohammadreza Zolfaghari, Kamaljeet Singh, and Thomas Brox. 2018. Eco: Efficient convolutional network for online video understanding. In *Proceedings of the European Conference on Computer Vision (ECCV '18)*. 695–712.

Received 28 May 2023; revised 19 September 2023; accepted 8 November 2023