A Semantic and Motion-Aware Spatiotemporal Transformer Network for Action Detection

Matthew Korban, *Senior Member, IEEE*, Peter Youngs, Scott T. Acton, *Fellow, IEEE*

Abstract—This paper presents a novel spatiotemporal transformer network that introduces several original components to detect actions in untrimmed videos. First, the multi-feature selective semantic attention model calculates the correlations between spatial and motion features to model the spatiotemporal interactions between different action semantics properly. Second, the motion-aware network encodes the locations of action semantics in video frames utilizing the motion-aware 2D positional encoding algorithm. Such a motion-aware mechanism memorizes the dynamic spatiotemporal variations in action frames that current methods cannot exploit. Third, the sequence-based temporal attention model captures the heterogeneous temporal dependencies in action frames. In contrast to standard temporal attention used in natural language processing, primarily aimed at finding similarities between linguistic words, the proposed sequence-based temporal attention is designed to determine both the differences and similarities between video frames that jointly define the meaning of actions. The proposed approach outperforms the state-of-the-art solutions on four spatiotemporal action datasets: AVA 2.2, AVA 2.1, UCF101-24, and EPIC-Kitchens.

Index Terms—Human action detection, transformer network, spatiotemporal attention, action semantics, positional encoding.

1 Introduction

Spatiotemporal action detection aims to localize action class instances in untrimmed videos in both spatial and temporal dimensions [1]. However, such a spatiotemporal action detection faces several challenges, including (1) complex spatiotemporal interactions between action semantics, (2) dynamic spatiotemporal variations in action semantics, and (3) heterogeneous temporal dependencies between action frames. We will explain our proposed solutions to solve the challenges above as follows:

The semantics are the meaningful components of actions that can be categorized into two fundamental types: *persons* and *objects*. [2] is one of the first methods incorporating action semantics into spatiotemporal action detection. However, their method was limited to finding the motion trajectory of localized objects over video frames, which alone might not be enough to model various actions. [3] addressed such a limitation by including persons and objects that better represent human actions. Yet, they did not consider the interactions between semantics, which are crucial parts of actions. To solve this issue, [4] proposed a video action transformer network that captures the correlations between a central person and the surrounding pixels. However, they did not explicitly model the interactions between all the action semantics in video frames. However, many actions are defined based on the spatiotemporal interactions between action semantics. For example, the action "kicking a ball" is characterized by the interaction between a "person" and a moving "ball", in both spatial and temporal domains. These domains are associated with the spatial and motion properties of action semantics, respectively. More examples will be illustrated in Fig. 3. To address this issue, we propose a multi-feature semantic attention model that enables the transformer network to selectively capture the spatiotemporal interactions between action semantics based on the correlations between their motion and spatial features. More details about the multi-feature selective attention model are explained in Section 3.3.2.

The positional encoding of the transformer network extends its capability to encode the order information in the data [5]. As an alternative to the sequential connection in RNN and LSTM, the positional encoding allows the transformer network to process sequential data more effectively and efficiently. The positional encoding was initially proposed to represent 1D temporal order information in linguistic words [6]. It is also used to encode spatial order information in computer vision [7], [8] recently. Two main strategies to incorporate positional information in computer vision have been ordered pixels [7] and patches of pixels [8].

Nevertheless, the current positional encoding strategies have two issues when dealing with video frames: First, they are still based on standard 1D temporal positional encoding designed for linguistic words, which might not work well for images due to their 2D nature. Furthermore, existing positional encoding algorithms are limited to static spatial order, which cannot accurately represent dynamic action semantics. Specifically, the positional information of action semantics changes based on their spatiotemporal variations caused by the dynamic movements of action semantics in videos. These spatiotemporal variations will be illustrated later in Fig. 2. Hence, we propose a motion-aware 2D positional encoding algorithm, which is more effective than the standard methods in modeling the positions of action semantics considering their spatiotemporal variations. The motion-aware 2D positional encoding will be discussed in

- *Matthew Korban and Scott T. Acton are with the Department of Electrical and Computer Engineering, University of Virginia, Charlottesville, VA, 22904.*
  E-mail: {acw6ze@virgnia.edu, acton@virginia.edu}
- *Peter Youngs is with the Department of Curriculum, Instruction and Special Education at the University of Virginia, Charlottesville, VA 22903. E-mail: pay2n@virginia.edu*

Manuscript received.

more detail in Section 3.3.1.

Temporal order information is essential in modeling actions since human action is a sequential process over different times. Human actions are, however, temporally heterogeneous, making modeling the temporal dependencies between action frames challenging. For example, in some actions such as "running", the temporal dependencies between similar and adjacent frames are essential. On the other hand, in many actions such as "jumping", the critical temporal dependencies are between distinctive and non-adjacent frames, so-called keyframes such as "start", "middle", and "end" of the jump.

Traditional sequential methods such as Recurrent Neural Network (RNN) encounter difficulty computing heterogeneous temporal dependencies in actions because of limited temporal receptive fields and bias toward adjacent action frames. A better way to model temporal dependencies in actions is using a transformer network, which can more effectively process non-adjacent frames due to a larger temporal receptive field. However, the standard transformer network was initially designed for natural language processing in which the highest temporal dependencies are between the exact linguistic words, followed by words from similar categories [6]. Hence, while the transformer network can include the temporal dependencies between non-adjacent frames, it is still heavily biased toward similar and often adjacent action frames. Consequently, the transformer network struggles to capture the heterogeneous temporal dependencies between distinctive and non-adjacent frames. Therefore, to resolve this issue, we propose a sequence-based temporal attention model to capture heterogeneous temporal dependencies within the transformer network effectively. More details about the above are discussed in Section 3.3.4.

In summary, the main **contributions** of this paper are as follows:

- A novel spatiotemporal transformer network is proposed that includes several original components to detect actions in untrimmed videos by properly modeling the action semantics, their interactions, and movements. The new design solves the fundamental issues of the standard transformer network in action modeling and understanding. To do this, the transformer network and the attention mechanism are renovated in both spatial and temporal domains.
- We are the first to develop a multi-feature selective semantic attention model to capture the important spatiotemporal interactions between action semantics based on the correlations between their spatial and motion properties. The multi-feature attention handles the issue of the standard self-attention, which is restricted to a single feature space. The proposed transformer also uses a selective attention model, which, by selecting informative inputs, is more effective and efficient than the standard self-attention in modeling action semantics.
- A novel motion-aware 2D positional encoding is introduced that uses a 2D motion memory module to model dynamic spatiotemporal semantic variations in action frames. This overcomes the drawback of the standard 1D positional encoding, which is limited to static spatial positions, making it ineffective in dealing with 2D images and movements in videos.
- A sequence-based temporal attention model is suggested that, along with the sequence-based temporal positional encoding, can effectively capture heterogeneous temporal dependencies in action frames that might exist in distinctive and often non-adjacent frames. This eliminates the limitations of traditional temporal attention, which is biased towards temporal dependencies between similar and commonly adjacent frames in action videos.
- The proposed method outperforms state-of-the-art methods on four public spatiotemporal action benchmarks, AVA (V2.2 and V2.1), UCF101-24, and EPIC-Kitchens.

2 Related Work

There are three relevant topics covered in this section, including methods based on (1) action semantics, (2) multiple features for action modeling, (3) spatiotemporal action detection.

2.1 Action Semantics

To represent action semantics, earlier approaches used handcrafted features, such as shape descriptors extracted from human silhouettes [9] and local binary patterns captured from human parts [10]. With advances in deep learning, action semantics can now be analyzed more effectively. Several studies in the literature used deep networks for action semantics that focus either on persons [11] or objects [12]. In [11], a convolutional feature-based action tubelet detector is presented for modeling persons over different time periods. [12] suggested an object-centric feature alignment mechanism to signify critical objects in action videos. However, various actions in the wild rely on complex interactions between persons and objects. To accommodate this complexity, [13] introduced a transformer network to compute the interactions between persons and objects based on their spatial features. As of yet, no effective strategy has been developed to capture the spatiotemporal relationship between action semantics based on spatial and motion features.

2.2 Multi-feature Networks in Action Modeling

[14] introduced one of the first deep networks that used both RGB images and optical flow fields to extract spatial and motion features for action modeling and recognition. [15] redesigned its architecture, including softmax and pooling layers, in order to improve a multi-feature network. Despite taking advantage of multiple features in [14], [15], they did not suggest any explicit solution to employ multiple features effectively in action recognition. Therefore, [16] proposed directly enhancing the individual and hybrid representations of multiple features via a spatiotemporal pyramid pooling and a fusion mechanism, respectively. [17] improved this multi-feature representation by using region proposals rather than the whole frames suggested in [14], [15], [16], to emphasize more informative parts of

the optical flow and RGB images. [18] further strengthened the joint representation of multiple features by computing the relationship between feature vectors with a self-attention mechanism. Still, calculating the relationship between action semantics based on the correlations between their spatial and motion features remains unresolved.

2.3 Spatiotemporal Action Detection

In one of the earliest spatiotemporal action detection approaches, [1] proposed codebooks in which one codeword represented spatial and temporal information about each action class. A major weakness of [1] is its inability to successfully detect actions in complex backgrounds due to using handcrafted features. This shortcoming was addressed by [3] by developing a deep CNN that could more accurately detect spatiotemporal action instances in complex background environments. [19] enhanced such as a deep network in [3] by detecting multiple spatial action instances per frame with a more efficient single-shot multibox detector. Before now, each action proposal was based on a fixed model, accumulating the error over time. In [20], this issue was tackled by proposing a progressive learning framework that adapts to new relevant action contexts as they arise. As an extension of the previous work, [21] used the track-of-interest alignment technique to cope with large spatial variations of in-the-wild action videos. A more effective strategy to tackle both large spatial and temporal variations is the transformer network, which has increasingly been used for action detection and recognition.

3 METHODOLOGY

3.1 Method Overview

Fig. 1 shows the overview of the proposed pipeline for action detection. Given a sequence of RGB frames, $I^{RGB} = \{I_t^{rgb} \in \mathbb{R}^{H\times W\times 3}, t = 0, 1, ..., \tau\}$, the goal is to find the action class scores, $\hat{Y}$, and the start of the end of action, t_s and t_e, respectively. Here, τ is the length of the action sequence; and H and W indicate the size of the full image (height and width). The suggested pipeline includes two main stages: preprocessing (Section 3.2), in which the input data are prepared, and the spatiotemporal transformer network (Section 3.3), which models and detects the action sequence.

In the prepossessing stage, the spatial action semantics $S^G = \{Z^G, O^G\}$ are detected that includes the geometries of persons $Z^G = \{z_i^g \in \mathbb{R}^{h_i\times w_i\times 3}, i = 0, 1, ..., N\}$ and objects $O^G = \{o_i^g \in \mathbb{R}^{h_i'\times w_i'\times 3}, i = 0, 1, ..., N'\}$, where N, N', o^g, and z^g are the numbers of persons, number of objects, individual detected persons, and detected objects in an action frame, respectively. h_i and w_i indicate the sub-image size of the detected persons; and h_i' and w_i' shows the sub-image size of the detected objects. Note that the spatial action semantics represent the spatial information of persons and objects in RGB images. We used [7] to detect the action semantics, including persons and objects. Apart from the spatial features, motion features also are important properties of action semantics. So, to incorporate the motion features, the optical flow fields $I^{FLOW} = \{I_t^{flow} \in \mathbb{R}^{H\times W\times 3}, i = 0, 1, ..., \tau\}$, are computed using [22], a highly efficient optical flow extraction algorithm. The optical flow fields result from converting 2D optical flow motion vectors to three-channel images to enable computing the multi-feature attention between spatial and motion features (more details are in Section 3.3.2). Then, for each frame, the motion semantics $S^M = \{Z^M, O^M\} \in I_t^{flow}$, including the motions of persons $Z^M = \{z_i^m \in \mathbb{R}^{h_i\times w_i\times 3}, i = 0, 1, ..., N\}$ and objects $O^M = \{o_i^m \in \mathbb{R}^{h_i'\times w_i'\times 3}, i = 0, 1, ..., N'\}$ are enhanced and segmented to make it invariant to camera movements (Section 3.2.1). The bounding boxes B^G, obtained from the semantic detection algorithm [7], are used for motion segmentation. Here $B^G = \{B^Z, B^O\}$, where $B^Z = \{b_i^z \in \mathbb{R}^4, i = 0, 1, ..., N\}$ and $B^O = \{b_i^o \in \mathbb{R}^4, i = 0, 1, ..., N'\}$; and b^p and b^o are the corresponding bounding boxes for each motion semantic.

The proposed spatiotemporal transformer network includes several modules to model the spatial and motion semantics obtained from the preprocessing step and detect the action sequence. The multi-feature selective semantic attention model (Section 3.3.2) captures the critical relationship between action semantics based on the correlations between their spatial and motion features. To handle the spatiotemporal variations in action semantics, the motion memory module (Section 3.3.1) utilizes the semantic motion vectors (S^V) and updates the semantic positional encoding of the transformer network using the 2D horizontal and vertical semantic motion memory offsets $\Delta p_i^x \in \Delta P^X$ and $\Delta p_i^y \in \Delta P^Y, i \in \{0, 1, , , , , N + N'\}$. Here, S^V is computed from the optical flow motion vectors before conversion to images (more details are in Section 3.2.1).

The motion memory module also outputs the 2D semantic motion memory features F_M^X and F_M^Y. The output of the multi-feature selective attention is the multi-head semantic attention, A^H, which represents the most informative selection of correlative patterns between spatial and motion semantics extracted by the heads of the transformer networks. The multi-feature fusion module (Section 3.3.3) combines the motion memory and the multi-feature semantic features and directs the dataflow in multiple layers of the deep network. The output of the multi-feature fusion module is the final set of semantic features, f_t^s, that are captured in different frames to form the sequence of final semantic , $X_s = \{f_t^s, t = 0, 1, ..., \tau\}$. Subsequently, X_s is processed in the sequence-based temporal attention model (Section 3.3.4) to extract the heterogeneous temporal dependencies between different frames, f_t^s. The sequence-based temporal attention values, $\hat{A}$, proceed to the classification and regression stages to detect the action sequence. The implementation details of the proposed pipeline are in Section 4.1. We will discuss the aforementioned components of the proposed method thoroughly in the following sections. Our algorithm summary is shown in Table 1. The extended version of the algorithm summary is in the Supplementary Material.

3.2 Preprocessing

In the proposed pipeline, the preprocessing stage includes semantic detection, optical flow extraction, motion enhancement, and segmentation to prepare the inputs for the transformer network. We exploited [7], a well-established

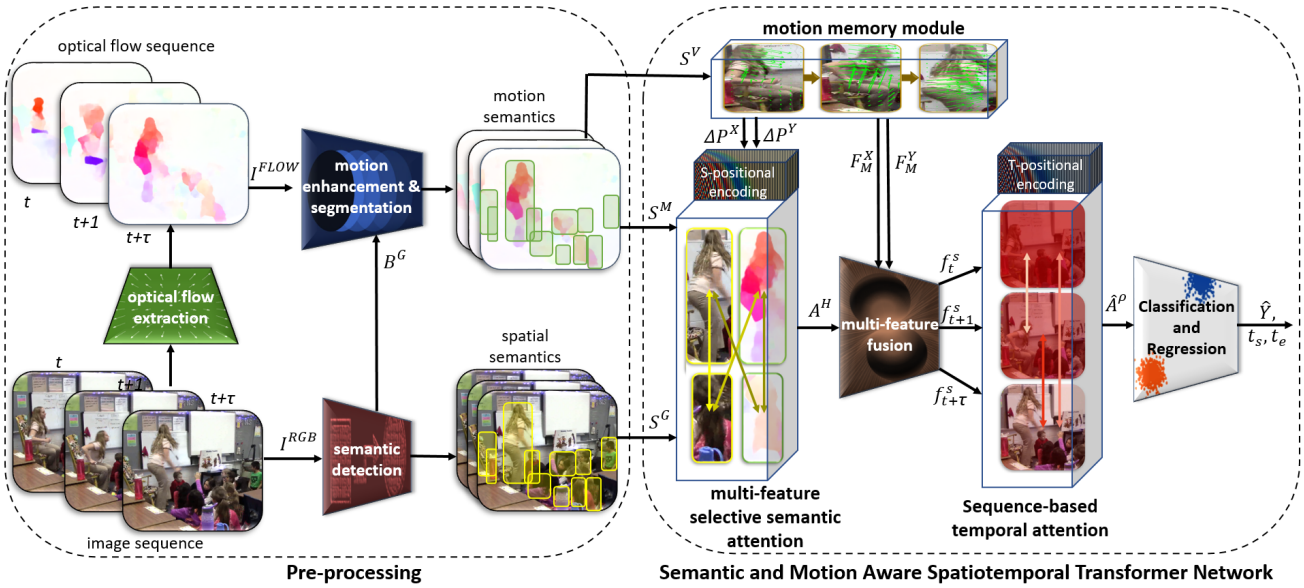


Fig. 1. The pipeline for the proposed method includes the preprocessing stage and the transformer network. Given the sequence of RGB images, first, the spatial semantics and optical flow fields are extracted in the preprocessing stage. The motion enhancement and segmentation algorithm extracts the motion semantics that are invariant to camera movement. The multi-feature selective attention model captures the correlative patterns between spatial and motion semantics. The motion memory module updates the semantic positional encoding (S-positional encoding) of the transformer network and makes it semantically motion-aware. The multi-feature fusion combines the extracted features and directs them to the deep network. The sequence-based temporal attention model captures the heterogeneous temporal dependencies between different times that are then used to detect the action sequence in the classification and regression stage.

TABLE 1
Pipeline algorithm summary in a hierarchical order indicating each phase (in bold), the summary of each phase, and the inputs and outputs of each phase.

Phase/Summary	Inputs	Outputs
Semantic Detection extracts action semantics	I^{RGB}	S^G
Optical Flow Extraction extracts optical flow	I^{RGB}	S^M, S^V
Motion Enhancement improves motion features	B^G, I^V	S^M, S^V
Feature Embedding extract spatiotemporal features	S^G, S^M	X^G, X^M
Motion Memory Module provides motion information	S^V	$\Delta P^X, \Delta P^Y$
MA 2D Positional Encoding for motion-aware transformer	$\Delta P^X, \Delta P^Y$	P_A^X, P_A^Y
MF Semantic Attention for correlations in multi-features	S^G, S^M	A^H
Multi-Feature Fusion combines features in layers	A^H	X_s
SB Temporal Attention computes temporal relations	X_s	$\hat{A}$
Classification and Regression classifies actions and frames	$\hat{A}$	$\hat{Y}, t_s, t_e$

state-of-the-art object/person detection algorithm, to extract spatial (RGB) action semantics, including persons and objects for each frame as $S^G = \{Z_t^G, O_t^G\}$. In this stage, the corresponding bounding boxes for persons and objects are $B_t^G = \{B_t^Z, B_t^O\} \in \mathbb{R}^{(N+N')\times 4}$.

The optical flow fields are estimated from RGB images utilizing [22], a highly efficient state-of-the-art optical flow estimation algorithm. Optical flow is a powerful and popular modality to represent motions in actions. However, optical flow is sensitive to camera movement, a common issue in videos captured in the wild. In other words, the camera movements can remarkably distort the motion information depicted in the optical flow fields. To address this problem, a semantic motion enhancement and segmentation algorithm is developed to effectively use semantic motion vectors in the transformer network.

3.2.1 Semantic Motion Enhancement and Segmentation

Given the semantic bounding boxes $B^G = \{B^Z, B^O\}$ obtained from the semantic detection algorithm, the distorted optical flow fields $I_t^{flow}(x, y)$ (affected by camera movements) and the corresponding motion vectors, $I^V(u, v) \in \mathbb{R}^{H\times W\times 2}$, the goal of the semantic motion enhancement and segmentation algorithm is to extract the enhanced motion semantics including persons and objects $S^M = \{Z^M, O^M\}$; and the corresponding semantic motion vectors, $S^V = \{Z^V, O^V\}$, which are invariant to camera movements. Here, u and v are scalar units that define the motion displacement between the image pixels in the time t, as $(x^{(t)}, y^{(t)})$, and the time $t + \omega$, as $(x^{(t+\omega)}, y^{(t+\omega)})$. The semantic motion enhancement and segmentation algorithm is designed based on the dominant motions of persons in action frames. Consequently, we consider persons as the foreground and the remaining portions of the frame as the background, affected mainly by camera movements. The semantic motion enhancement and segmentation algorithm consists of two steps: motion modeling and motion restoration, which are explained in the Supplementary Material.

3.3 Semantic and Motion-Aware Spatiotemporal Transformer Network

Before delivering the action semantics to the transformer network, they are embedded in the feature space. To do

such, first, the outputs of the prepossessing step, including each spatial action semantic, $s_i^g \in S^G$, are resized as $\mathbb{R}^{h_i \times w_i \times 3} \rightarrow \mathbb{R}^{\hat{h} \times \hat{w} \times 2}$, and each motion semantic, $s_i^m \in S^M$ as $\mathbb{R}^{h_i' \times w_i' \times 3} \rightarrow \mathbb{R}^{\hat{h} \times \hat{w} \times 2}$. Here, $\hat{h}$ and $\hat{w}$ represent the fixed image size. So, the resized spatial and motion semantics are $\hat{S}^G$ and $\hat{S}^M$, respectively. Next, the action semantics are converted to spatial and motion semantic features as $X^G = \{X_Z^G, X_O^G\}$ and $X^M = \{X_Z^M, X_O^M\}$, respectively. This feature embedding is performed using the convolutional layers, $Conv^G$, and $Conv^M$ as:

$$\begin{aligned} X^G &= Conv^G(\hat{S}^G, W^G) : \mathbb{R}^{\hat{N} \times \hat{h} \times \hat{w} \times 3} \rightarrow \mathbb{R}^{\hat{N} \times d_f}, \\ X^M &= Conv^M(\hat{S}^M, W^M) : \mathbb{R}^{\hat{N} \times \hat{h} \times \hat{w} \times 3} \rightarrow \mathbb{R}^{\hat{N} \times d_f}, \end{aligned} \tag{1}$$

where $\hat{N} = N + N'$ is the total number of action semantics, d_f is the size of feature embedding, and W^G and W^M are the kernel weights. The embedded feature set includes X_Z^G and X_O^G, which are the spatial features of persons and objects; and X_Z^M, and X_O^M, which are the motion features of persons and objects, respectively. The aforementioned feature embedding is illustrated in Fig. 4.

3.3.1 Motion-Aware 2D Positional Encoding.

The transformer is a deep network that has recently become popular in action detection [23], [24]. The transformer network has several advantages over traditional temporal networks because of the concurrent processing of inputs. This reduces processing time, extends the temporal receptive field of the network, and prevents vanishing gradients, a common issue encountered during the training phase of transitional sequential networks [6]. The transformer network is able to process inputs concurrently due to the *positional encoding* that adds the order information of the inputs. As an integral part of the transformer network, positional encoding was initially designed for natural language processing, mainly a time-related problem. The standard 1D temporal positional encoding [6] is illustrated as follows.

$$\begin{aligned} P(p_t, 2n) &= sin(\frac{p_t}{\psi^{\frac{2n}{d_m}}}), \\ P(p_t, 2n+1) &= cos(\frac{p_t}{\psi^{\frac{2n}{d_m}}}), \end{aligned} \tag{2}$$

where p_t, n, and d_m are the temporal order of the input, the dimension index of the positional embedding, and the semantic model size, respectively. ψ is a large integer as suggested by the original work [6] to accommodate high-dimensional embedding features.

The positional encoding was later applied to spatial problems, such as object detection [7], and spatiotemporal problems, such as action detection [24], without effective adaptation to these areas. Two main issues are (1) the 1D nature of the standard positional encoding, which leaves it less effective in handling 2D images; and (2) its incapability to handle spatiotemporal variations in spatiotemporal inputs such as action sequences. The proposed positional encoding has two characteristics that address the aforementioned issues. Firstly, the proposed positional encoding is "2D", making it more adept in dealing with 2D images. Secondly, and more importantly, the suggested positional encoding is motion-aware, which makes it more effective in handling spatiotemporal variations in action sequences.

Currently, the most common method for encoding spatial positions in images is to segment them into patches [8], [25], [26]. In this regard, the current methods treat 2D images as 1D inputs and assign the positional labels to each patch following the standard temporal positional encoding indicated in (2). Nevertheless, the current methods employ fixed patches that cannot accommodate spatiotemporal variations in video frames. Fig. 2 shows examples of such spatiotemporal variations and compares the proposed motion-aware positional encoding to the standard patch-based approach in dealing with this issue. In this example, the image is divided into 18 labeled between p_1 to p_{18}. Here, the red and green basketball players change their positions when moving from time t (in Fig. 2 (a)) to $t + \omega$ (in Fig. 2 (b)). So, now, the green player, as a defender, switches his position with the red player, on offense at position p_{11}. Consequently, when spatiotemporal changes occur in action semantics, the transformer network cannot model the semantic movements at different action frames when using the standard patch-based positional encoding. By contrast, the proposed motion-aware 2D positional encoding memorizes the position changes of each action semantics at different times and adaptively updates their positional information within the transformer network as shown in Fig. 2 (c). We later also numerically compare the proposed motion-aware 2D positional encoding to the standard one in Section 4.4.1 and Table 7. The motion-aware 2D positional encoding is formulated as follows:

$$\begin{aligned} P_A^X(p_i^x, 2n) &= sin(\frac{p_i^x + \Delta p_i^x}{\psi^{\frac{2n}{d_m}}}), \\ P_A^X(p_i^x, 2n+1) &= cos(\frac{p_i^x + \Delta p_i^x}{\psi^{\frac{2n}{d_m}}}), \end{aligned} \tag{3}$$

$$\begin{aligned} P_A^Y(p_i^y, 2n) &= sin(\frac{p_i^y + \Delta p_i^y}{\psi^{\frac{2n}{d_m}}}), \\ P_A^Y(p_i^y, 2n+1) &= cos(\frac{p_i^y + \Delta p_i^y}{\psi^{\frac{2n}{d_m}}}), \end{aligned} \tag{4}$$

where p_i^x and p_i^y are initial horizontal and vertical positions for each action semantics $s_i \in \{S^G, S^M\}$, respectively that are obtained by the standard patch-base approach [8]. $\Delta p_i^x \in \Delta P^X$, and $\Delta p_i^y \in \Delta P^Y$, $i = \{0, 1, ..., \hat{N}\}$ are horizontal and vertical semantic motion memory offsets for each action semantics. ΔP^X and ΔP^Y include the motion offsets for all the action semantics. ω is the duration of the motion. In this work, we define the index of positional embedding as $n \in \{0, 1, ..., d_f/2\}$.

Δp_i^x and Δp_i^y adaptively update the positions of action semantics according to their motions during the action sequence. A set of concatenated embedded 2D positional encoding $P_A = \{P_A^X, P_A^Y\}$ is delivered to the transformer network. To do such, the embedded positional encoding is added to the inputs as:

$$\hat{X}^G = P_A + X^G, \quad \hat{X}^M = P_A + X^M, \tag{5}$$

where $\hat{X}^G = \{\hat{X}_Z^G, \hat{X}_O^G\}$ and $\hat{X}^M = \{\hat{X}_Z^M, \hat{X}_O^M\}$ are positional encoded spatial and motion semantic features (for

persons and objects), respectively. Two new components of the motion-aware positional encoding, Δp_i^x, and Δp_i^y are computed using the motion memory module. Details of this computation are found below.

Motion Memory Module. The input to the motion memory module is the sequence of semantic motion vectors as $S_{seq}^V = \{S_t^V, t = 0, 1, ..., \tau\}$. The outputs are (1) the semantic motion memory offsets, ΔP^X and ΔP^Y, that are used for our motion-aware 2D positional encoding; and (2) 2D semantic motion memory features, $F_M = \{F_M^X, F_M^Y\}$ utilized in the multi-feature fusion stage to enrich the feature representation. The motion memory module is activated when there is a notable change in movement. This makes the pipeline more efficient and reduces the noise caused by random movements. The motion memory module memorizes the changes in semantic motion vectors using the motion memory network as follows:

$$\begin{aligned} update: \quad & z_t = \sigma(W^z x_t + U^z h_{t-\hat{\omega}}), \\ reset: \quad & r_t = \sigma(W^r x_t + U^r h_{t-\hat{\omega}}), \\ current: \quad & n_t = tanh(W^h x_t + r_t \odot U^h h_{t-\hat{\omega}}), \\ output: \quad & h_t = z_t \odot h_{t-\hat{\omega}} + (1-z_t) \odot n_t, \end{aligned} \tag{6}$$

where x_t is the input, $\odot$ represents the element-wise multiplication, and σ is a sigmoid function. z_t, r_t, n_t, and h_t are the update gate, reset gate, current gate, and output, respectively. W^z, W^r, and $W^h \in \mathbb{R}^{\hat{N}\hat{h}\hat{w} \times d_g}$; and U^z, U^r, and $U^h \in \mathbb{R}^{d_g \times d_g}$ are parameter matrices, where d_g is the state size. The above formulation is inspired by the learning steps of the gated recurrent unit [27]. Moreover, $\hat{\omega}$ is a dilated temporal value that represents the time when there exists significant motion in the action sequence based on a threshold value, T_h. Given the inputs that are horizontal and vertical components of the semantic motion vectors, S_X^V, and S_Y^V, the memorized horizontal and vertical outputs of the networks are h^X and h^Y, respectively. The motion memory networks memorize the movements of action semantics during action sequences. As a result, the motion memory networks update the positional encoding, making it motion-aware. The final outputs of the motion memory networks are computed by using an averaging pooling layer (*AvPool*) as:

$$\begin{aligned} \Delta P^X &= AvPool(h^X) : \mathbb{R}^{\hat{N} \times d_g} \rightarrow \mathbb{R}^{\hat{N}}, \\ \Delta P^Y &= AvPool(h^Y) : \mathbb{R}^{\hat{N} \times d_g} \rightarrow \mathbb{R}^{\hat{N}}, \end{aligned} \tag{7}$$

The motion memory module also outputs the semantic motion memory features, $F_M = \{F_M^X, F_M^Y\}$, as:

$$F_M^X = Conv^X(h_Z^X, W^X), \quad F_M^X = Conv^Y(h_Z^X, W^Y). \tag{8}$$

Note that $Conv^X$ and $Conv^Y : \mathbb{R}^{N \times d_g} \rightarrow \mathbb{R}^{N \times d_f}$ and W^X and W^Y are the kernel weights. Only the part of the network outputs corresponding to persons as h_Z^X, and $h_Z^X \in \mathbb{R}^{N \times d_g}$ are selected in the feature selection. This is due to the fact that in action sequences, persons' movements are more critical than objects' movements. F_M is used in the multi-feature fusion module to improve the feature representation of actions.

3.3.2 Multi-feature Selective Semantic Attention

Based on their spatial and motion features, many actions are characterized by spatiotemporal interactions between their semantics, including persons and objects. Transformer networks are designed to capture the interactions between different inputs using an attention mechanism [6]. Therefore, we propose a multi-feature selective attention model to extract such interactions between action semantics. There are three differences between the proposed multi-feature attention and the standard "cross-attention" mechanism. First, the proposed attention is selective, meaning only informative queries are considered. Second, as opposed to the existing methods that include the whole frame/input, the suggested strategy focuses only on action semantics. Finally, four multi-feature attention types are introduced to enrich the action feature representation.

In our view, four types of multi-feature semantic interactions occur in human actions, which are illustrated in Fig. 3. The first scenario happens when all the interacting action semantics are stationary, such as the "person" and "cake/candle" in the action "blowing a candle" in Fig. 3 (a). As a result, the spatial-to-spatial attention, A^{GG}, represents such a stationary-only interaction between action semantics. In the second case, the entire action semantics are in motion, such as the "person" and the "jet ski" in the action "jet skiing" illustrated in Fig. 3 (b). Consequently, the motion-to-motion attention, A^{MM}, serves such motion-only interactions between action semantics. As for the third scenario, the spatial-to-motion attention, A^{GM}, provides the interaction between the moving semantics, the "person pitching the ball", and the stationary one, the "person waiting for the ball" in Fig. 3 (c). Finally, the motion-to-spatial attention, A^{MG}, represents the interactions between moving and stationary semantics, in this example, basketball "player" and "net", respectively (shown in Fig. 3 (d)). Among the above multi-feature attentions types, A^{GG} and A^{MM} are intra-feature attention, while A^{GM} and A^{MG} are inter-feature attention. In the aforementioned examples, the action semantics are highlighted by bounding boxes. The motion semantics are visualized by the green motion vectors, which are obtained from optical flow fields.

The architecture of the multi-feature selective semantic attention model is illustrated in Fig. 4. The attention mechanism is to find the correlations between the *query* (Q) and *keys* (K) and then map them to *values* (V). The four suggested attention types to represent the correlations between spatial and motion semantic features are shown in Fig. 4 and are formulated as:

$$\begin{aligned} A^{GM} &= Softmax(\frac{Q^G (K^M)^T}{\sqrt{d_h}}) V^M, \\ A^{MG} &= Softmax(\frac{Q^M (K^G)^T}{\sqrt{d_h}}) V^G, \\ A^{GG} &= Softmax(\frac{Q^M (K^G)^T}{\sqrt{d_h}}) V^G, \\ A^{MM} &= Softmax(\frac{Q^M (K^M)^T}{\sqrt{d_h}}) V^M, \end{aligned} \tag{9}$$

The multi-feature semantic queries, keys, and values are illustrated as follows:

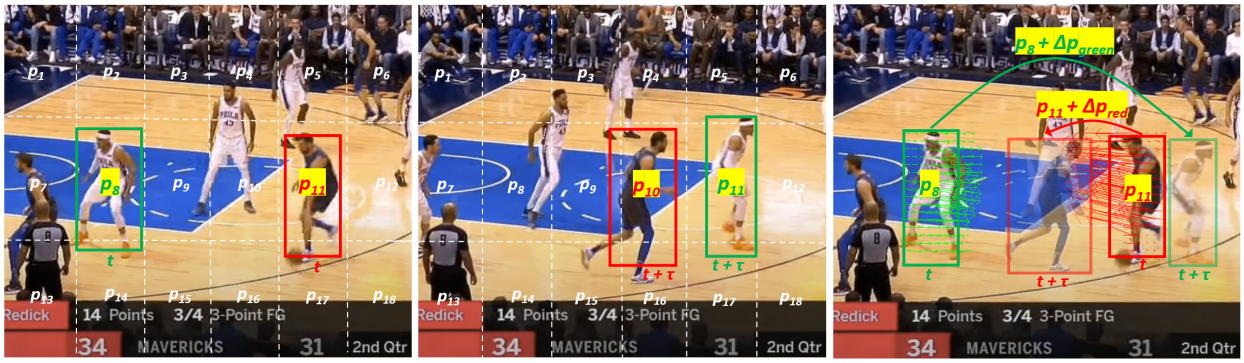


Fig. 2. The motion-aware positional encoding, (c), compared to the standard one, (a) and (b), in dealing with spatiotemporal action semantic variations: two basketball players, red and green, change their positions from time t, (a) to $t+\tau$, (b). So the red player (p_{11} as the offensive player) switched his position to the green player (now p_{11} the defender). The proposed motion-aware positional encoding, (c), can memorize the position changes of two basketball players using the motion memory offsets, Δp_{green} and Δp_{red}. The green and red arrows show the motion vectors obtained from the optical flow fields.

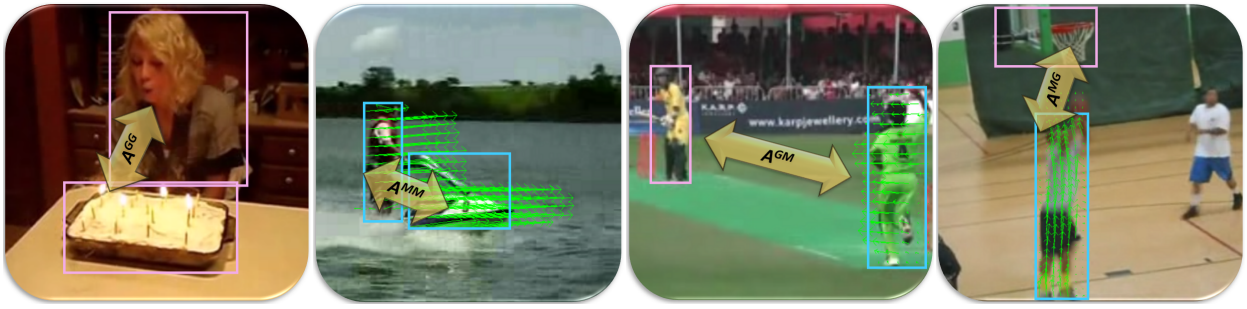


Fig. 3. Some examples of our multi-feature attention types that capture the spatiotemporal semantic interactions in action samples. (a): *spatial-to-spatial attention* between the "sitting person" and the "stationary cake" ; (b): *motion-to-motion attention* between the "moving person" and the "moving jet ski" ; (c): *spatial-to-motion attention* between the "stationary waiting player" and the "moving pitching player"; (d): *motion-to-spatial attention* between the "moving jumping player" and the "stationary net". The green arrows show the motion vectors obtained from the optical flow fields. The action samples are collected from the UCF101 dataset [28].

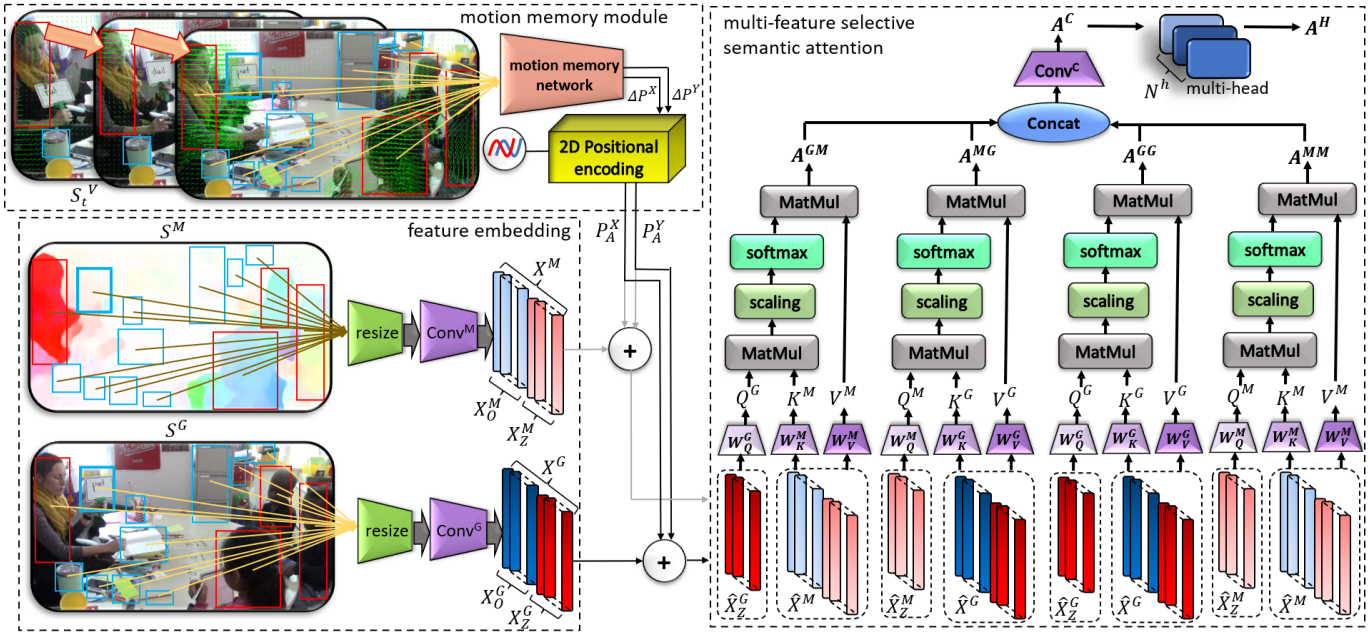


Fig. 4. The proposed transformer network includes several modules to capture the multi-feature semantic features. The feature embedding converts the motion and spatial semantics to features. The motion memory module memorizes the semantic position changes and includes them as the motion-aware positional encoding in the semantic multi-feature extraction. The multi-feature selective attention represents the correlations between persons with other persons and the most relevant objects. In this action example, "teacher using an instructional tool", these correlations represent the interactions between the "teacher" and the 'students", and the"teacher" and relevant objects such as the "handheld whiteboard."

$$\begin{aligned} Q^G = \hat{X}_Z^G W_Q^G,\ K^G = \hat{X}^G W_K^G,\ V^G = \hat{X}^G W_V^G, \\ Q^M = \hat{X}_Z^M W_Q^M,\ K^M = \hat{X}^M W_K^M,\ V^M = \hat{X}^M W_V^M, \end{aligned} \tag{10}$$

where $W_Q^G, W_Q^M \in \mathbb{R}^{N\times d_f \times d_h}$, and $W_K^G, W_K^M, W_V^G, W_V^M \in \mathbb{R}^{\hat{N}\times d_f \times d_h}$ are projecting weights. $d_h = df/N^h$ is the size of attention head, N^h is number of attention heads, and T is a transpose operation.

The compound semantic attention head, A^C, is defined by combining the four multi-feature attention types and then applying a convolutional operation as:

$$\begin{aligned} A^{C'} = Concat^C(A^{GM}, A^{MG}, A^{GG}, A^{MM}), \\ A^C = Conv^C(A^{C'}, W^C), \end{aligned} \tag{11}$$

where $Conv^C : \mathbb{R}^{4\times N\times d_h} \rightarrow \mathbb{R}^{N\times d_h}$, $Concat^C$ is a concatenation operator that stacks the inputs as $Concat^C : 4\times\mathbb{R}^{N\times d_h} \rightarrow \mathbb{R}^{4\times N\times d_h}$, W^C is the kernel weights, and $A^{C'}$ is the concatenated head. To extract features more effectively through different heads of the transformer network, the multi-head semantic attention, $A^H \in \mathbb{R}^{N\times d_f}$ is computed by concatenating each of the N^h compound semantic attention heads as:

$$A^H = Concat^H(A_i^C, i = \{0, 1, ..., N^h\}), \tag{12}$$

where $Concat^H : N^h \times \mathbb{R}^{N\times d_h} \rightarrow \mathbb{R}^{N\times N^h \times d_h} = \mathbb{R}^{N\times d_f}$.

Selective Attention. As shown in Fig. 4 and (10), in the proposed attention model, the queries and keys/values are defined differently. The queries are only persons, $\hat{X}_Z$, while the keys/values include both persons and objects, $\hat{X} = \{\hat{X}_Z, \hat{X}_O\}$. However, in the standard self-attention model, queries and keys/values are derived from the same source. We argue that the proposed selective attention model is more effective and efficient than the standard self-attention in modeling action semantics. In particular, the selective attention model computes the interactive correlations between persons-to-persons and persons-to-objects, which are essential to a variety of actions. Therefore, we exclude the computation of correlations between objects-to-objects, which are often irrelevant to actions, thereby reducing the expressive power of action features. Specifically, among a large number of background objects in action videos, only a few are relevant to the action. An example is shown in Fig. 4, where the most relevant object is the "Handheld whiteboard" in the action "teacher using an instructional tool". By contrast, the "teacher" is correlated with all the other persons, "students", in this activity. We will later numerically compare the performance of the selective attention model to the standard one in Section 4.4.3 and Table 9. Moreover, by excluding the objects from the queries, the selective attention is now $\mathbb{R}^{N\times d_h}$, that compared to the standard self-attention $\mathbb{R}^{(N+N')\times d_h}$, requires fewer number of computations.

3.3.3 Multi-feature Fusion

The proposed transformer network consists of multiple layers with a defined relationship between the successive layers $l-1$ and l, as illustrated as follows:

$$\begin{aligned} \hat{B}_O^l = MFSSA(B_O^{l-1}) + B_O^{l-1}, \quad l \in \{2, ..., L\}, \\ B_O^l = MLP(Norm(\hat{B}^l)) + \hat{B}^l, \quad l \in \{2, ..., L\}, \end{aligned} \tag{13}$$

where $MFSSA$ is the multi-feature selective semantic attention, B_O is the layer output, $\hat{B}_O$ is the intermediate layer output, L is the number of layers, and $Norm$ is a normalization layer.

For the first layer, the relation between the inputs and output is shown as follows:

$$\hat{B_O^1} = MFSSA(\hat{X}^G, \hat{X}^M) + F^{M.A}, \tag{14}$$

where $F^{M.A} \in \mathbb{R}^{N\times d_h}$ is our motion-aware features defined as:

$$F^{M.A} = Conv^A(Concat^A(X_Z^G, X_Z^M, F_M^X, F_M^Y), W^A). \tag{15}$$

where $Conv^A : \mathbb{R}^{4\times N\times d_f} \rightarrow \mathbb{R}^{N\times d_f}$, $Concat^A : 4\times\mathbb{R}^{N\times d_f} \rightarrow \mathbb{R}^{4\times N\times d_f}$, and W^A are the kernel weights. Here, $MFSSA(\hat{X}^G, \hat{X}^M) = A^H$. To properly update the projecting weights for the keys and values in the layers $l \geq 2$ during the training, the selected inputs for the multi-feature keys and values are converted to their original size of the first layer as $\mathbb{R}^{\hat{N}\times d_f}$. The final output of the multi-feature fusion for frame t is the final semantic features, f_t^s is illustrated as:

$$f_t^s = Conv^F(B_O^L, W^O) : \mathbb{R}^{N\times d_f} \rightarrow \mathbb{R}^{d_f}, \tag{16}$$

where W^O are the kernel weights. The sequence of f_t^s is used as the multi-feature representation of semantics for each action frame in the sequence-based temporal attention model.

3.3.4 Sequence-based Temporal Attention

Given the sequence of final semantic features for different frames, $X_s = \{f_t^s, t = 0, 1, ..., \tau\}$, the goal of the sequence-based temporal attention model, $\hat{A} \in \mathbb{R}^{\tau\times d_f}$ is to compute the temporal dependencies between semantic features in different frames $t \in \tau$, effectively. Temporal attention is capable of modeling long sequences without discriminating against older data, which makes it suitable for processing long, untrimmed action sequences. Nevertheless, the standard temporal attention was designed for neural language processing, specifically for modeling linguistic words [6]. Accordingly, the standard temporal attention tends to assign the highest attention to words themselves and followed by words from similar categories. [29]. However, such a homogeneous attention model is unsuitable for modeling heterogeneous temporal dependencies between action frames. There are many actions in which the temporal dependencies between distinctive frames, so-called keyframes, provide essential information. For example, in the action sequence "triple jump" the temporal dependencies between the three distinctive steps "hop", "step", and "jump" define the meaning of the action. The standard homogeneous temporal attention, however, mainly focuses on the temporal dependencies between similar and often adjacent frames,

discarding essential temporal dependencies between distinctive frames. To illustrate the above, we will show some examples in Section 4.4.4, and in Fig. 5.

On the other hand, the proposed sequence-based temporal attention model can compute the heterogeneous temporal dependencies between action frames without discriminating on the basis of their similarity. It is because the sequence-based temporal attention model focuses on the temporal dependencies that are relevant for actions, rather than individual frames themselves. Some conceptual poof is provided in the supplementary material. We will later numerically compare the sequence-based temporal attention to the standard one in Section 4.4.4 and Table 10.

Some researchers suggested using a probabilistic approach to enhance the standard attention mechanism. [30] proposed exploiting a mixture model whose parameters are updated following the observed queries. [31] presented a mixture of Gaussian keys as an effective replacement for redundant heads in a transformer network. These methods, however, are based on Expectation Maximization (EM), which has several issues, especially when used in a deep learning framework. This includes high computational cost, sensitivity to initialization, model complexity, and slow convergence [32]. On the other hand, our sequence-based temporal attention is simple yet effective and does not require any complex and iterative optimization algorithms such as EM.

The standard temporal attention [6] is shown as:

$$A(Q_\Gamma, K_\Gamma, V_\Gamma) = Softmax(\frac{Q_\Gamma K_\Gamma^T}{\sqrt{d_f}})V_\Gamma, \tag{17}$$

where the temporal queries, keys, and values are Q_Γ, K_Γ, and $V_\Gamma \in \mathbb{R}^{\tau \times d_f}$; and τ is the number of frames. $A^{corr}(Q_\Gamma, K_\Gamma) = Q_\Gamma K_\Gamma^T \in \mathbb{R}^{\tau \times \tau}$ is the temporal attention correlation matrix that indicates the frame-by-frame temporal dependencies. In the standard approach, $Q_\Gamma = (X_s + P)W_Q$, $K_\Gamma = (X_s + P)W_K$, and $V_\Gamma = (X_s + P)W_V$, where W_Q, W_K, and $W_Q \in \mathbb{R}^{\tau \times d_f}$ are the temporal projecting weights. P is the temporal positional encoding following (2) that is added to the input sequence, X_s to include the temporal order information. So, let's start with the definition of the standard temporal attention correlation matrix, $A^{corr}(Q_\Gamma, V_\Gamma)$, that only focuses on the relationship between individual inputs (queries and keys):

$$A^{corr}(Q_\Gamma, K_\Gamma) = (X_s + P)W_Q(X_s + P)^T W_K^T, \tag{18}$$

The above can be rewritten as:

$$A^{corr}(Q_\Gamma, K_\Gamma) = \overbrace{X_s W_Q X^T W_K^T}^{(a)} + \overbrace{P W_Q P^T W_K^T}^{(b)} + \underbrace{X_s W_Q P^T W_K^T}_{(c)} + \underbrace{P W_Q X_s^T W_K^T}_{(d)}, \tag{19}$$

We argue that in the above, the expressions (c) and (d) do not contain useful information. This is because, during the learning process, they do not lead to meaningful gradient updates since they are the results of multiplying the input sequence and the position information. So, a more efficient temporal attention correlation matrix $\tilde{A}^{corr}(Q_\Gamma, K_\Gamma)$ can be shown as follows:

$$\tilde{A}^{corr}(Q_\Gamma, K_\Gamma) = X_s W_Q X^T W_K^T + P W_Q P^T W_K^T, \tag{20}$$

The above can be rewritten as:

$$\tilde{A}^{corr}(Q_\Gamma, K_\Gamma) = Q_\Gamma K_\Gamma^T + P_Q P_K^T. \tag{21}$$

Consequently, the following defines the efficient temporal attention:

$$\tilde{A} = Softmax(\frac{\tilde{A}^{corr}}{\sqrt{d_f}})V_\Gamma \tag{22}$$

Until now, the temporal dependencies between Q_Γ and K_Γ are calculated between individual frames without including their relation to the action sequence. To resolve this issue, the sequence-based temporal attention correlation matrix, $\hat{A}^{corr}$, revises (21) to provide the temporal dependencies between the inputs, Q_Γ and K_Γ with respect to the distribution of the action sequence, X_s, as shown as:

$$\begin{aligned} \hat{A}^{corr}(Q_\Gamma, K_\Gamma; X_s) = I_\tau - \\ (Q_\Gamma - K_\Gamma)S_X^{-1}(Q_\Gamma - K_\Gamma)^T/N_X - \\ (P_Q - P_K)S_X^{-1}(P_Q - P_K)^T/N_P, \end{aligned} \tag{23}$$

where $S_X \in \mathbb{R}^{d_f \times d_f}$ is the covariance matrix of the distribution X_s, and $I_\tau \in \mathbb{R}^{\tau \times \tau}$ is matrix of ones. N_X and N_P are normalizing terms. The above also presents a more effective sequence-based temporal positional encoding than the original P_Q and P_K. (23) is based on the intuition of the Mahalanobis distance that is explained in Supplementary Material. Finally, the sequence-based temporal attention is computed as follows:

$$\hat{A} = Softmax(\frac{\hat{A}^{corr}}{\sqrt{d_f}})V_\Gamma, \tag{24}$$

3.3.5 Classification and Regression

The final stage of the proposed pipeline is classification and regression. The multi-label classification layer includes two frame-level and sequence-level predictions shown as follows:

$$\begin{aligned} \hat{Y}^S &= Softmax(Conv^s(\hat{A}^P, W^s)), \\ \hat{Y}^F &= Softmax(Conv^f(\hat{A}^P, W^f)), \end{aligned} \tag{25}$$

where $\hat{Y}^S$, and $\hat{Y}^F$ are the sequence and frame prediction scores, respectively. $Conv^s : \mathbb{R}^{\tau \times df} \rightarrow \mathbb{R}^{C_l}$, $Conv^f : \mathbb{R}^{\tau \times df} \rightarrow \mathbb{R}^{\tau \times C_l}$, and W^s and W^f are the kernel weights. Note that C_l is the number of classes. The set of class scores can be defined as $\hat{Y} = \{\hat{Y}^S, \hat{Y}^F\}$. Finally, the regression layer calculates the start and end of the action sequence, t_s, and t_e, for a given class c so that the selected time interval satisfies: $(t_s, t_e; c) = \underset{t}{\text{argmax}}(\frac{1}{\tau}\sum_t(\hat{Y}^F_{t_s,t_e}) + \hat{Y}^S_{t_s,t_e})$.

The loss function of the proposed network is shown as follows:

$$L = -\sum_{t=1}^{\tau}\sum_{c=1}^{C_l} y_t^{(c)} log\hat{y}_t^{(c)} - \alpha\sum_{c=1}^{C_l} Y^{(c)} log\hat{Y}^{(c)}, \quad (26)$$

where y and $\hat{y}$ are the ground truth and predicted values for each frame t of class c at time t. Y and $\hat{Y}$ are the ground truth and predicted values for each action sequence, respectively. Note that α is a loss adjustment parameter.

4 Experimental Results

4.1 Implementation Details

The implementation details of the proposed pipeline are summarized in Table 2. In this table, all the hyperparameters mentioned in the paper are described. The Model Zoo with the DETR architecture [7] and R101 backbone is used for object/person detection to balance efficiency and performance. The object/person detection threshold is 0.5. The optical flow is extracted by the FastFlowNet [22] fine-tuned on FlyingThings3D [33] with using 320×448 patches during data augmentation. The learnable semantic and temporal projecting weights W_K^G, W_K^M, W_V^G, W_V^M, W_Q, W_K, and W_V are initialized with random values. For the EPIC-Kitchens [34] dataset, only the classification layer is used. The network weights are initialized from the Kinetics pre-trained model. The focal loss [35] was used during the training.

S^V are required to update our motion-aware positional encoding, which is also 2D. On the other hand, S^M helps calculate multi-feature attention between motion and spatial features that are also derived from images. Moreover, the motion feature embedding in our pipeline is computed by a convolutional layer requiring images as the inputs.

All the experiments are conducted using PyTorch 1.7 on a server PC with dual Nvidia RTX 3090 GPUs (24GB VRAM), AMD Ryzen Threadripper 3990X 64-Core Processor, and 256GB of RAM.

TABLE 2
Implementation details of the proposed pipeline with corresponding sections.

Parameter	Value	Section
Semantic detection confidence threshold	0.5	3.2.1
Number of GMM distributions (K)	16	3.2.1
Number of action semantic ($\hat{N}$)	10	3.3
Semantic image size ($\hat{h} \times \hat{w}$)	128×128	3.3
Size of feature embedding (d_f)	2048	3.3
Positional encoding integer (ψ)	$1e^4$	3.3.1
Duration of motion (τ)	32	3.3.1
Size of motion network state (d_g)	1024	3.3.1
Motion memory module threshold (T_h)	2.35	3.3.1
Number of attention heads (N^h)	4	3.3.2
Size of attention head (d_h)	512	3.3.2
Number of multi-feature layers (L)	6	3.3.3
Input normalizing term (N_X)	$1e^3$	3.3.4
Position normalizing term (N_P)	$2e^3$	3.3.4
Loss adjustment parameter (α)	2.4	3.3.5
Optimizer	SGD	3.3.5
Number of training epochs	100	3.3.5
Number of videos per batch	16	3.3.5
Learning rate	$1e^{-4}$	3.3.5
Learning rate decay (every 30 epochs)	0.1	3.3.5
Weight decay	$1e^{-6}$	3.3.5
Multi-label classification threshold	0.5	3.3.5

4.2 Datasets

The proposed pipeline is evaluated using four public spatiotemporal action datasets, AVA (version 2.1 and 2.2) [36], UCF101-24 [28], and EPIC-Kitchens [34]. Short descriptions of these datasets are given in the following.

AVA dataset consists of 15-minute video segments of 80 action classes. The AVA dataset is suitable for evaluating our proposed pipeline as it covers a notable variety of interactions between action semantics, including person-to-person and person-to-object interactions. Both AVA versions (v2.1 and v2.2) include 235 videos for training and 64 videos for validation. The annotations for bounding boxes and class labels are provided for every second of the video segments. AVA 2.2 revises the label annotation by adding more frame-level labels. Following the instruction of the benchmark, [36] and the previous studies [13], [37], [38], we used 60 action classes with the measurement metrics of *mean Average Precision (mAP)* with an IoU threshold of 0.5.

UCF101-24 is a subset of the UCF101 dataset [28] that includes 3'207 untrimmed videos from 24 action classes. As a part of THUMOS Challenge 2015, this dataset includes the annotation for bounding boxes and frames. It covers a variety of spatial and temporal action instances in each video, which is well-suited for our experiments. For evaluation, an mAP with an IoU threshold of 0.5 is used on the first split of the data following the previous work [13], [37].

EPIC-Kitchens comprises 55 hours of daily activities, primarily focusing on the interaction between persons and objects. This dataset features the most diverse objects among spatiotemporal action benchmarks, making it suitable to test the effectiveness of the proposed pipeline in modeling action semantics. Following a standard setting [37], [39], we used 22'675 videos for training and 5'886 videos for testing. According to the literature [40], [41], [42], the *top-1* accuracy metric is adopted to evaluate the suggested method on the EPIC-Kitchens dataset on "action", "noun", and "verb" classes. A *top-1* accuracy is obtained by comparing the highest predicted results with the ground truth.

4.3 Comparative Results

The comparative results of four public benchmarks are as follows.

4.3.1 Results on AVA 2.2

Table 3 illustrates the comparison between our method, SMAST, and the state-of-the-art approaches on the AVA 2.2 dataset. In [43], a video-masked autoencoder using pre-trained models achieved the highest transfer learning performance when using Kinetic-700. Overall, our proposed pipeline (SMAST) outperforms the state-of-the-art approaches on the AVA 2.2 dataset. Specifically, with the mAP@0.5 of **40.2%**, the suggested method surpasses the best current benchmark [43] by 0.9.

4.3.2 Results on AVA 2.1

Table 4 compares The proposed method, SMAST, to the state-of-the-art strategies on the AVA 2.1 dataset. In conclusion, the proposed SMAST surpassed the state-of-the-art methods previously used on the AVA 2.1 dataset with the mAP@0.5 of **33.1%**, exceeding the highest existing benchmark [49] by 1.1%.

TABLE 3
Comparison of our proposed method (SMAST) with the state-of-the-art strategies on the AVA 2.2 dataset.

Team	Method	Pub/Year	mAP@0.5 (%)
Tang et al. [37]	AIA	ECCV 2020	34.4
Feichten al. [44]	MoCo	CVPR 2021	20.3
Fan et al. [45]	MVT	ICCV 2021	27.3
Feichten al. [46]	X3D	CVPR 2022	27.4
Chen et al. [47]	WOO	ICCV 2021	28.3
Wu et al. [48]	TLVU	CVPR 2021	31.0
Pan et al. [13]	ACARNet	CVPR 2021	31.7
Liu et al. [42]	ORVT	CVPR 2022	26.6
Zhao et al. [49]	Tuber	CVPR 2022	33.6
Wu et al. [41]	MeMViT	CVPR 2022	35.4
Wei et al. [50]	MaskFeat	CVPR 2022	38.8
Tong et al. [43]	VideoMAE	NIPS 2022	39.3
Faure et al. [51]	HIT	WCACV 2023	32.6
Korban et al.	**SMAST**	- - - - -	**40.2**

TABLE 4
Comparison of our suggested method (SMAST) with the state-of-the-art approaches on the AVA 2.1 dataset.

Team	Method	Pub/Year	mAP@0.5 (%)
Sun et al. [52]	ACRN	ECCV 2018	17.4
Girdhar et al. [4]	VT	CVPR 2019	27.6
Wu et al. [38]	LTFB	CVPR 2019	27.7
Stroud et al. [53]	D3D	CVPR 2020	23.0
Tang et al. [37]	AIA	ECCV 2020	31.2
Wu et al. [48]	TLVU	CVPR 2021	27.8
Chen et al. [47]	WOO	ICCV 2021	28.0
Pan et al. [13]	ACARNet	CVPR 2021	30.0
Shah et al. [54]	PGA	CVPR 2022	28.4
Zhao et al. [49]	TubeR	CVPR 2022	32.0
Korban et al.	**SMAST**	- - - - -	**33.1**

4.3.3 Results on UCF101-24

Table 5 illustrates the comparative results on the UCF101-24 benchmark. SMAST, outperformed the other strategies on the UCF101-24 benchmark with the mAP@0.5 of **85.5%**, exceeding the highest performance [51] by 0.7.

TABLE 5
Comparison of our pipeline (SMAST) with the state-of-the-art methods on the UCF101-24 dataset.

Team	Method	Pub/Year	mAP@0.5 (%)
Song et al. [55]	Tacnet	CVPR 2019	72.1
Pramono et al. [56]	HSAN	CVPR 2019	73.7
Yang et al. [20]	STEP	CVPR 2019	75.0
Zhang et al. [57]	ASM	CVPR 2019	77.9
Tang et al. [37]	AIA	ECCV 2020	78.8
Li et al. [58]	MOC	ECCV 2020	78.0
Su et al. [17]	PCSC	PAMI 2020	79.2
Liu et al [59]	ACDnet	PRL 2021	70.9
Pan et al. [13]	ACARNet	CVPR 2021	84.3
Kumar et al. [60]	ESSL	CVPR 2022	69.9
Li et al. [61]	DSRM	SIVP 2022	81.2
Zhao et al. [49]	TubeR	CVPR 2022	81.3
Faure et al. [51]	HIT	WCACV 2023	84.8
Korban et al.	**SMAST**	- - - - -	**85.5**

4.3.4 Results on EPIC-Kitchens

6 shows the comparison between the proposed pipeline, SMAST, and state-of-the-art methods on the EPIC-Kitchens benchmark. Our proposed method outperformed the existing approaches with the top-1 accuracy for "action", "verb", and "noun" of **50.9%**, **70.1%**, and **64.8%**, respectively.

TABLE 6
Comparative results on the EPIC-Kitchens dataset showing top-1 accuracy (%) for different classes of action, noun, and verb.

Team	Method	Pub/Year	action	verb	noun
Tang et al. [37]	AIA	ECCV 2020	27.7	60.0	37.2
Nagrani et al. [62]	MBT	NIPS 2021	43.4	64.8	58.0
Arnab et al. [63]	Vivit	ICCV 2021	44.0	66.4	56.8
Patrick et al. [64]	Mformer	NIPS 2021	44.5	67.0	58.5
Kondrat et al. [65]	Movinets	CVPR 2021	47.7	72.2	57.3
Liu et al. [42]	ORVT	CVPR 2022	45.7	68.4	58.7
Wu et al. [41]	MeMViT	CVPR 2022	48.4	71.4	60.3
Girdhar et al. [40]	Omnivore	CVPR 2022	49.9	69.5	61.7
Yan et al. [66]	MVT	CVPR 2022	50.5	69.9	63.9
Korban et al.	**SMAST**	- - - - -	**50.9**	**70.1**	**64.8**

4.4 Ablation Study

In this section, we present an ablation study that evaluates the impact of each module in the proposed action detection solution.

4.4.1 Motion-Aware 2D Positional Encoding and Motion Enhancement

Table 7 indicates the impact of the proposed motion-aware (MA) 2D positional encoding and the proposed motion enhancement algorithm on the overall action detection performance. Our findings show that the MA 2D positional encoding is ineffective without the motion enhancement algorithm. Specifically, it only boosted the overall performance by 0.1 % (from 30.9 % to 31.0%). The reason is that, without motion enhancement, the motion awareness of 2D positional encoding is impacted by the camera movement, a common issue in action videos captured in the wild. Notably, the camera movement causes incorrect extraction of motion vectors, resulting in imprecise calculations of the motion memory offsets, ΔP^X and ΔP^Y, in the 2D positional encoding. In contrast, when the motion enhancement algorithm is used, the MA 2D positional encoding increases the overall performance by 2.2% (from 30.9% to 33.1%). The 1D positional encoding is computed following [8].

TABLE 7
The impact of the motion-aware (MA) 2D positional encoding and motion enhancement algorithm on the overall action detection performance. The evaluation is conducted on the AVA 2.1 benchmark.

Scenario	mAP@0.5
Standard 1D positional encoding	30.9
MA 2D positional encoding (**no** motion enhancement)	31.0
MA 2D positional encoding (**with** motion enhancement)	**33.1**

4.4.2 Multi-Feature Semantic Attention

Table 8 illustrates the impact of various multi-feature attention types, and their combinations, on the overall action detection performance (tested on the AVA 2.1 dataset). When a single attention type is used, the highest performance is achieved by the intra-feature attention types, including A^{GG} with a mAP@0.5 of 26.5% followed by A^{MM}, with a mAP@0.5 of 25.6%. The inter-feature attention types, A^{MG}, and A^{GM}, alone did not result in competitive performance. However, combined with A^{GG} and A^{MM}, the inter-feature attention types, A^{MG}, and A^{GM}, boosted the overall action detection performance. The results show that although

many actions can still be effectively modeled independently, many of them are dependent on the interaction between spatial and motion features. The highest performance is obtained when all the multi-feature attention types, A^{GG}, A^{MM}, A^{GM}, and A^{MG} are used, yielding a mAP@0.5 of 33.1%.

TABLE 8
The impact of different multi-feature (MF) attention types, A^{GG} (spatial-to-spatial), A^{MM} (motion-to-motion), A^{GM} (spatial-to-motion), and A^{MG} (motion-to-spatial) on the overall action detection performance. Different attention types represent standard one-stream and two-stream baselines and our Single, double, combined, and full MF attentions.

Attention type	mAP@0.5 (%)
A^{GG} (one-stream RGB baseline)	26.5
A^{MM} (one-stream FLLOW baseline)	25.6
A^{GM} (single MF attention)	24.7
A^{MG} (single MF attention)	24.4
A^{GG}+ A^{MM} (two-stream RGB+FLOW baseline)	28.0
A^{GM}+ A^{MG} (double MF attention)	27.1
A^{GG}+ A^{MM}+ A^{GM} (combined MF attention)	31.8
A^{GG}+ A^{MM}+ A^{MG} (combined MF attention)	30.9
$\boldsymbol{A^{GG} + A^{MM} + A^{GM} + A^{MG}}$ (full model)	**33.1**

4.4.3 Selective Attention

Table 9 compares various forms of self-attention and the proposed selective attention model characterized by two types of action semantics, persons (Z) and objects (O). The lowest performance is due to the self-attention mechanism that defined queries and keys as objects (mAP@0.5 of 24.3%). In contrast, the maximum performance among the self-attention types is achieved with the queries and keys specified as persons + objects ($Z + O$) with a mAP@0.5 of 31.8%. Overall, the proposed selective attention model with the queries defined as persons (Z) and the keys as persons + objects ($Z + O$) yields a mAP@0.5 of 33.1%.

TABLE 9
The comparison between different self-attention types and the proposed selective attention model based on two action semantics, persons (Z), and objects (O). Here the self-attention follows the standard attention mechanism in the baseline [6].

Attention category	Inputs (Semantics)	mAP@0.5 (%)
Self-attention	Query (O) , Key (O)	24.3
Self-attention	Query (Z) , Key (Z)	27.5
Self-attention	Query ($Z + O$) , Key ($Z + O$)	31.8
Selective attention	**Query** ($\boldsymbol{Z}$) , **Key** ($\boldsymbol{Z + O}$)	**33.1**

4.4.4 Sequence-based Temporal Attention

Table 10 compares the suggested sequence-based temporal attention with the efficient and standard ones on the AVA 2.1 dataset. In this table, the standard temporal positional encoding (TPE) was presented in (18), the efficient temporal attention (ETPE) in (21), and the sequence-based temporal positional encoding (PTPE) in (23). Moreover, A, $\tilde{A}$, and $\hat{A}$ were shown in (17), (22), and (24), respectively.

The results illustrated that the efficient temporal attention ($\tilde{A}$) + ETPE with a mAP@0.5 of 30.9% slightly performed better than the standard temporal attention (A) + TPE with a mAP@0.5 of 30.4%. Using sequence-based models for both temporal attention and positional encoding, $\hat{A}$ + PTPE, yielded the maximum performance with a mAP@0.5 of 33.1.

TABLE 10
The comparison between standard temporal attention, efficient temporal attention, and the proposed sequence-based temporal attention. Different types of positional encodings are the standard temporal positional encoding (TPE), efficient temporal positional encoding (ETPE), and the proposed sequence-based temporal positional encoding (PTPE).

Modules	mAP@0.5 (%)
Standard temporal attention (A) + TPE (baseline [6])	30.4
Efficient temporal attention ($\tilde{A}$) + ETPE	30.9
sequence-based temporal attention ($\hat{A}$) + ETPE	32.5
sequence-based temporal attention ($\hat{\boldsymbol{A}}$) + PTPE	**33.1**

Fig. 5 compares the proposed sequence-based temporal attention correlation and the standard one on an action sequence, "triple jump", with 10 sampled frames. Note that A^{corr} and $\hat{A}^{corr}$ were explained in (18) and (21), respectively. Here, A^{corr}_{i-j} and $\hat{A}^{corr}_{i-j}$ are the temporal correlation between the frame $i \in \tau$ and $j \in \tau$, Investigation of Fig. 5 more closely indicates that the standard temporal attention model leans toward producing greater values to similar frames, such as $t = 1$ and $t = 2$ or $t = 4$ and $t = 5$. On the other hand, The standard temporal attention sets lower values to distinctive frames, such as $t = 8$ and $t = 9$ or $t = 9$ and $t = 10$. By contrast, the proposed sequence-based temporal attention does not discriminate among frames based on similarity. In other words, $\hat{A}^{corr}$ can more effectively provide the temporal relationship between distinctive frames, such as the start, middle, and end of the jump at $t = 8$, $t = 9$, and $t = 10$ in the sequence of "triple jump". In many actions, such distinctive frames, so-called key-frames yield crucial temporal information. Furthermore, the proposed sequence-based temporal attention, $\hat{A}^{corr}$, set less importance on the temporal dependencies between non-critical frames that are less relevant to the action, such as the running sequence at $t = 1$ and $t = 2$.

4.4.5 Enhancement of the current frameworks

We integrated the proposed selective multi-feature attention (MFA) into several state-of-the-art spatiotemporal action detection frameworks based on the transformer network. This includes the Tublet attention in TubeR [49], actor-context attention in ACARNet [13], and interaction attention in AIA [37]. The results for the AVA 2.1 dataset are shown in Table 11. The outcomes indicated that all the methods improved when the new MFA was used, with the highest enhancement of 1.2% for AIA.

TABLE 11
The results of enhancing the state-of-the-art spatiotemporal action detection frameworks using the proposed selective multi-feature attention (MFA).

Method	mAP@0.5% (baseline)	**mAP@0.5% (MFA)**
AIA [37]	31.2	**32.4**
ACARNet [13]	30.0	**30.1**
TubR [49]	32.0	**32.5**

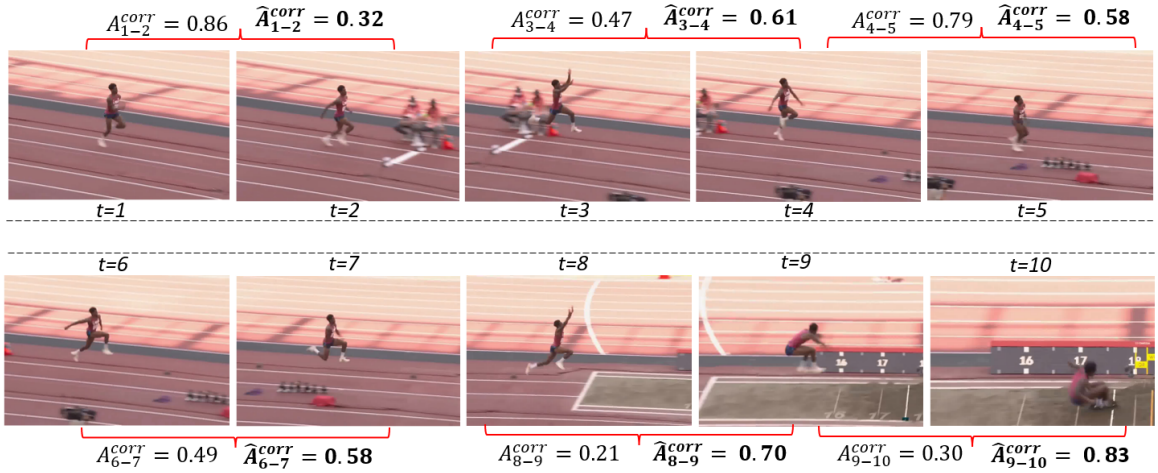


Fig. 5. An example of action "triple jump" illustrates the comparison between the proposed sequence-based temporal attention correlation, $\hat{A}^{corr}$, and the standard temporal attention correlation, A^{corr}. Here, A^{corr}_{i-j} and $\hat{A}^{corr}_{i-j}$ represents the temporal correlation between the frame i and j. The standard temporal attention tends to give higher values to similar frames, such as $t = 1$ and $t = 2$ or $t = 4$ and $t = 5$, and lower values to distinctive frames, such as $t = 8$ and $t = 9$ or $t = 9$ and $t = 10$. In contrast, the proposed temporal attention does not discriminate against frames based on their similarities. Hence, $\hat{A}^{corr}$ is more effective than A^{corr} in representing the temporal dependencies between distinctive frames such as the start, middle, and end of the jump at $t = 8$, $t = 9$, and $t = 10$ that are critical in defining the meaning of the action. Additionally, $\hat{A}^{corr}$ places a lower priority on the relationships between less critical frames, such as the running sequence frames, $t = 1$ and $t = 2$, which is less relevant to the action "triple jump".

4.4.6 Error Analysis and Failure Cases

Fig. 6 (and Fig. 1 and Fig. 2 in the Supplementary Material) show some examples of success and failure cases from three sequences selected from the validation set of the AVA dataset. The true positives (TP), false positives (FP), and false negatives (FN) are the correct, incorrect, and missed action class predictions. Some **limitations** of the current pipeline that contributed to failure cases are (1) *2D restriction of videos* is a cause of error. For example, the action "walking along" was incorrectly detected (FP) as "talk to" or "listen to" due to the lack of 3D perception of face orientations. (2) *Occlusion* between persons and objects can distract the network from detecting the correction relations between action semantics. For example, a background object occluded with persons gave the wrong impression (FP) that they "give" objects to others. On the other hand, the occlusion can also cause an object to be hidden, preventing it from detecting (FN) the action class label "carry/hold" an object. (3) *The similarity between different classes* also has contributed to failure cases. For instance, the action "sit" could be confused with ""bend/bow".

TABLE 12
Efficiency analysis of different pipeline modules

Module	time (ms)
Action semantic detection	401
Optical Flow extraction	233
Motion Memory Module	106
Multi-feature semantic attention	160
Temporal Attention	41
Classification and Regression	35
Total pre-processing	634
Total network modules	342
Total runtime	976

TABLE 13
Efficiency analysis of different multi-feature attention (MFA) modules and their combinations

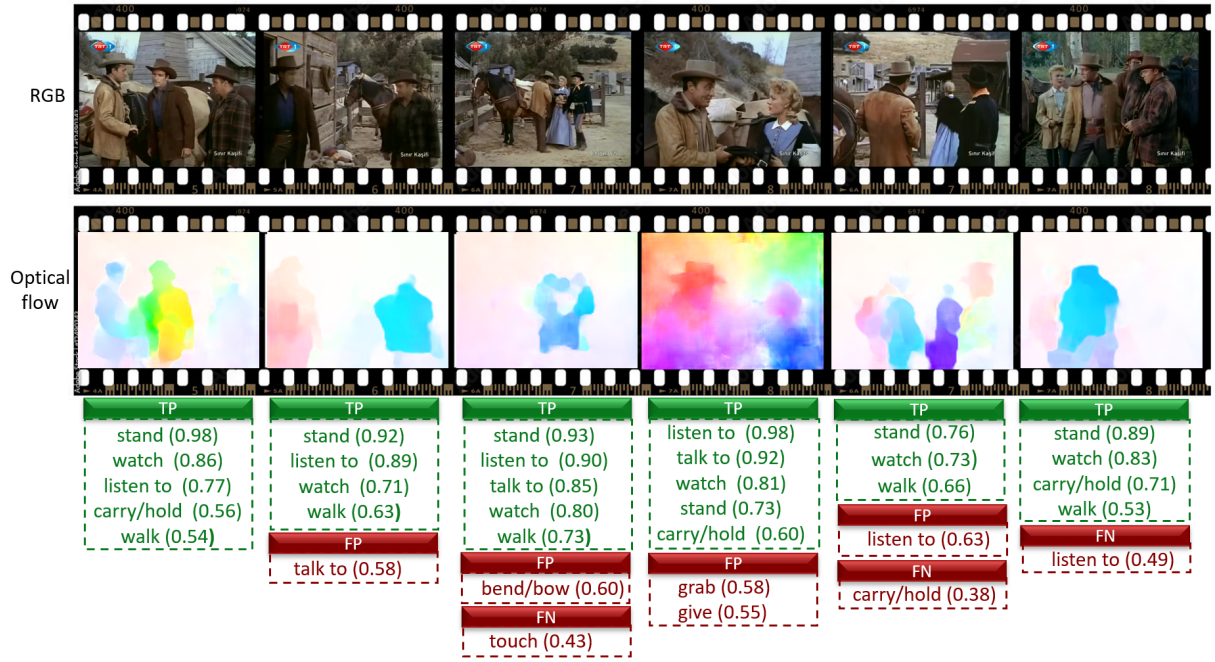
MFA Module	time (ms)
A^{GG}	101
A^{MM}	108
$A^{GM} + A^{MG}$	143
$A^{GG} + A^{MM} + A^{GM} + A^{MG}$	160

4.4.7 Computational Efficiency Analysis

Table 12 shows the efficiency analysis of different pipeline modules, which is based on running our algorithm on 30 frames sampled from 5 seconds of an action video. Furthermore, Table 13 indicates the efficiency analysis of different combinations of multi-feature attention types.

5 Conclusions and Future Work

This paper presents a novel spatiotemporal transformer network to model action semantics and their interactions effectively. In terms of contributed components, we introduce, first, a motion-aware 2D positional encoding that, in contrast to the standard one, can handle spatiotemporal variations in videos, especially geared toward action semantics. Second, a multi-feature attention model captures complicated multi-feature interactions between action semantics based on their spatial and motion properties. This model is accompanied by a special-purpose selective attention mechanism, which is more effective and efficient than the standard self-attention mechanism. Third, the sequence-based temporal attention model effectively captures heterogeneous temporal dependencies between action frames. In contrast to the traditional temporal attention mechanism that focuses on the relationship between individual action frames, the sequence-based temporal attention model prioritizes the temporal dependencies between frames based on their contribution to the action sequence. The pro-



video id: "7T5G0CmwTPo"

Fig. 6. Examples of success and failure cases on a sequence of the AVA validation set (video id: "7T5G0CmwTPo"). The results show the true positive (TP), false positive (FP), and false negative (FN) predicted classes with their confidence scores.

posed pipeline outperformed the state-of-the-art methods on four spatiotemporal action benchmarks, AVA 2.2, AVA 2.1, UCF101-24, and EPIC-Kitchens.

The current pipeline detects action semantics with a pre-trained network which is not end-to-end. A seamless learning process within the transformer network that encodes action semantics directly may be more desirable. The same criticism could also be levied toward our current method of encoding optical flow fields. As possible future work, we suggest extracting or simulating the optical flow fields end-to-end within the deep network.

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 20004.

References

[1] Antonios Oikonomopoulos, Ioannis Patras, and Maja Pantic. Spatiotemporal localization and categorization of human actions in unsegmented image sequences. *IEEE transactions on Image Processing*, 20(4):1126–1140, 2010.

[2] Du Tran and Junsong Yuan. Max-margin structured output regression for spatio-temporal action localization. *Advances in neural information processing systems*, 25, 2012.

[3] Philippe Weinzaepfel, Zaid Harchaoui, and Cordelia Schmid. Learning to track for spatio-temporal action localization. In *Proceedings of the IEEE international conference on computer vision*, pages 3164–3172, 2015.

[4] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 244–253, 2019.

[5] Bin Li, Jian Tian, Zhongfei Zhang, Hailin Feng, and Xi Li. Multitask non-autoregressive model for human motion prediction. *IEEE Transactions on Image Processing*, 30:2562–2574, 2020.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[7] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020.

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *2020 International Conference on Learning Representations (ICLR)*, 2020.

[9] Kai Guo, Prakash Ishwar, and Janusz Konrad. Action recognition from video using feature covariance matrices. *IEEE Transactions on Image Processing*, 22(6):2479–2494, 2013.

[10] Fahad Shahbaz Khan, Joost Van De Weijer, Rao Muhammad Anwer, Michael Felsberg, and Carlo Gatta. Semantic pyramids for gender and action recognition. *IEEE transactions on image processing*, 23(8):3633–3645, 2014.

[11] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid. Action tubelet detector for spatio-temporal action localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4405–4413, 2017.

[12] Xiaohan Wang, Linchao Zhu, Yu Wu, and Yi Yang. Symbiotic attention for egocentric action recognition with object-centric alignment. *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[13] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. Actor-context-actor relation network for spatio-temporal action localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 464–474, 2021.

[14] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014.

[15] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1933–1941, 2016.

[16] Xuanhan Wang, Lianli Gao, Peng Wang, Xiaoshuai Sun, and Xianglong Liu. Two-stream 3-d convnet fusion for action recognition

in videos with arbitrary size and length. *IEEE Transactions on Multimedia*, 20(3):634–644, 2017.

[17] Rui Su, Dong Xu, Luping Zhou, and Wanli Ouyang. Progressive cross-stream cooperation in spatial and temporal domain for action localization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(12):4477–4490, 2020.

[18] Weiming Hu, Haowei Liu, Yang Du, Chunfeng Yuan, Bing Li, and Stephen Maybank. Interaction-aware spatio-temporal pyramid attention networks for action classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10):7010–7028, 2022.

[19] Gurkirt Singh, Suman Saha, Michael Sapienza, Philip HS Torr, and Fabio Cuzzolin. Online real-time multiple spatiotemporal action localisation and prediction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3637–3646, 2017.

[20] Xitong Yang, Xiaodong Yang, Ming-Yu Liu, Fanyi Xiao, Larry S Davis, and Jan Kautz. Step: Spatio-temporal progressive learning for video action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 264–272, 2019.

[21] Gurkirt Singh, Vasileios Choutas, Suman Saha, Fisher Yu, and Luc Van Gool. Spatio-temporal action detection under large motion. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 6009–6018, 2023.

[22] Lingtong Kong, Chunhua Shen, and Jie Yang. Fastflownet: A lightweight network for fast optical flow estimation. In *2021 IEEE International Conference on Robotics and Automation (ICRA)*, pages 10310–10316. IEEE, 2021.

[23] Mingze Xu, Yuanjun Xiong, Hao Chen, Xinyu Li, Wei Xia, Zhuowen Tu, and Stefano Soatto. Long short-term transformer for online action detection. *Advances in Neural Information Processing Systems*, 34:1086–1099, 2021.

[24] Xiaolong Liu, Qimeng Wang, Yao Hu, Xu Tang, Shiwei Zhang, Song Bai, and Xiang Bai. End-to-end temporal action detection with transformer. *IEEE Transactions on Image Processing*, 31:5427–5441, 2022.

[25] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *ACM computing surveys (CSUR)*, 54(10s):1–41, 2022.

[26] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. *Advances in Neural Information Processing Systems*, 34:15908–15919, 2021.

[27] Amulya Arun Ballakur and Arti Arya. Empirical evaluation of gated recurrent neural network architectures in aviation delay prediction. In *2020 5th International Conference on Computing, Communication and Security (ICCCS)*, pages 1–7. IEEE, 2020.

[28] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012.

[29] Katikapalli Subramanyam Kalyan, Ajit Rajasekharan, and Sivanesan Sangeetha. Ammu: a survey of transformer-based biomedical pretrained language models. *Journal of biomedical informatics*, page 103982, 2021.

[30] Prasad Gabbur, Manjot Bilkhu, and Javier Movellan. Probabilistic attention for interactive segmentation. *Advances in Neural Information Processing Systems*, 34:4448–4460, 2021.

[31] Tam Minh Nguyen, Tan Minh Nguyen, Dung DD Le, Duy Khuong Nguyen, Viet-Anh Tran, Richard Baraniuk, Nhat Ho, and Stanley Osher. Improving transformers with probabilistic attention keys. In *International Conference on Machine Learning*, pages 16595–16621. PMLR, 2022.

[32] Wael Abd-Almageed, Aly El-Osery, and Christopher E Smith. Non-parametric expectation maximization: a learning automata approach. In *SMC'03 Conference Proceedings. 2003 IEEE International Conference on Systems, Man and Cybernetics. Conference Theme-System Security and Assurance (Cat. No. 03CH37483)*, volume 3, pages 2996–3001. IEEE, 2003.

[33] Nikolaus Mayer, Eddy Ilg, Philip Hausser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4040–4048, 2016.

[34] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, Toby Perrett, Will Price, et al. Scaling egocentric vision: The epic-kitchens dataset. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 720–736, 2018.

[35] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.

[36] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al. Ava: A video dataset of spatio-temporally localized atomic visual actions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6047–6056, 2018.

[37] Jiajun Tang, Jin Xia, Xinzhi Mu, Bo Pang, and Cewu Lu. Asynchronous interaction aggregation for action detection. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*, pages 71–87. Springer, 2020.

[38] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krahenbuhl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 284–293, 2019.

[39] Fabien Baradel, Natalia Neverova, Christian Wolf, Julien Mille, and Greg Mori. Object level visual reasoning in videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 105–121, 2018.

[40] Rohit Girdhar, Mannat Singh, Nikhila Ravi, Laurens van der Maaten, Armand Joulin, and Ishan Misra. Omnivore: A single model for many visual modalities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16102–16112, 2022.

[41] Chao-Yuan Wu, Yanghao Li, Karttikeya Mangalam, Haoqi Fan, Bo Xiong, Jitendra Malik, and Christoph Feichtenhofer. Memvit: Memory-augmented multiscale vision transformer for efficient long-term video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13587–13597, 2022.

[42] Roei Herzig, Elad Ben-Avraham, Karttikeya Mangalam, Amir Bar, Gal Chechik, Anna Rohrbach, Trevor Darrell, and Amir Globerson. Object-region video transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3148–3159, 2022.

[43] Zhan Tong, Yibing Song, Jue Wang, and Limin Wang. VideoMAE: Masked autoencoders are data-efficient learners for self-supervised video pre-training. In *Advances in Neural Information Processing Systems*, 2022.

[44] Christoph Feichtenhofer, Haoqi Fan, Bo Xiong, Ross Girshick, and Kaiming He. A large-scale study on unsupervised spatiotemporal representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3299–3309, 2021.

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

[46] Christoph Feichtenhofer. X3d: Expanding architectures for efficient video recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 203–213, 2020.

[47] Shoufa Chen, Peize Sun, Enze Xie, Chongjian Ge, Jiannan Wu, Lan Ma, Jiajun Shen, and Ping Luo. Watch only once: An end-to-end video action detection framework. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8178–8187, 2021.

[48] Chao-Yuan Wu and Philipp Krahenbuhl. Towards long-form video understanding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1884–1894, 2021.

[49] Jiaojiao Zhao, Yanyi Zhang, Xinyu Li, Hao Chen, Bing Shuai, Mingze Xu, Chunhui Liu, Kaustav Kundu, Yuanjun Xiong, Davide Modolo, et al. Tuber: Tubelet transformer for video action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13598–13607, 2022.

[50] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14668–14678, 2022.

[51] Gueter Josmy Faure, Min-Hung Chen, and Shang-Hong Lai. Holistic interaction transformer network for action detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3340–3350, 2023.

[52] Chen Sun, Abhinav Shrivastava, Carl Vondrick, Kevin Murphy, Rahul Sukthankar, and Cordelia Schmid. Actor-centric relation

network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 318–334, 2018.
[53] Jonathan Stroud, David Ross, Chen Sun, Jia Deng, and Rahul Sukthankar. D3d: Distilled 3d networks for video action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 625–634, 2020.
[54] Anshul Shah, Shlok Mishra, Ankan Bansal, Jun-Cheng Chen, Rama Chellappa, and Abhinav Shrivastava. Pose and joint-aware action recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3850–3860, 2022.
[55] Lin Song, Shiwei Zhang, Gang Yu, and Hongbin Sun. Tacnet: Transition-aware context network for spatio-temporal action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11987–11995, 2019.
[56] Rizard Renanda Adhi Pramono, Yie-Tarng Chen, and Wen-Hsien Fang. Hierarchical self-attention network for action localization in videos. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 61–70, 2019.
[57] Yubo Zhang, Pavel Tokmakov, Martial Hebert, and Cordelia Schmid. A structured model for action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9975–9984, 2019.
[58] Yixuan Li, Zixu Wang, Limin Wang, and Gangshan Wu. Actions as moving points. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*, pages 68–84. Springer, 2020.
[59] Yu Liu, Fan Yang, and Dominique Ginhac. Acdnet: An action detection network for real-time edge computing based on flow-guided feature approximation and memory aggregation. *Pattern Recognition Letters*, 145:118–126, 2021.
[60] Akash Kumar and Yogesh Singh Rawat. End-to-end semi-supervised learning for video action detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14700–14710, 2022.
[61] Hui Li, Wenjun Hu, Ying Zang, and Shuguang Zhao. Action recognition based on attention mechanism and depthwise separable residual module. *Signal, Image and Video Processing*, pages 1–9, 2022.
[62] Arsha Nagrani, Shan Yang, Anurag Arnab, Aren Jansen, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion. *Advances in Neural Information Processing Systems*, 34:14200–14213, 2021.
[63] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6836–6846, 2021.
[64] Mandela Patrick, Dylan Campbell, Yuki M. Asano, Ishan Misra Florian Metze, Christoph Feichtenhofer, Andrea Vedaldi, and João F. Henriques. Keeping your eye on the ball: Trajectory attention in video transformers. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.
[65] Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. Movinets: Mobile video networks for efficient video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16020–16030, 2021.
[66] Shen Yan, Xuehan Xiong, Anurag Arnab, Zhichao Lu, Mi Zhang, Chen Sun, and Cordelia Schmid. Multiview transformers for video recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3333–3343, 2022.

Matthew Korban (Senior Member, IEEE) received his BSc and MSc degrees in Electrical Engineering from the University of Guilan, where he worked on sign language recognition in video. He received his Ph.D. in Computer Engineering from Louisiana State University. Matthew has had leading roles in several groundbreaking projects, such as LSU CAVE2, one the world's largest and most advanced Cave2 VR. He has published several publications in top-tier computer vision conferences and journals such as ECCV and Pattern Recognition. He is currently a Postdoc Research Associate at the University of Virginia, working with Prof. Scott T. Acton. His research interest includes Human Action Recognition, Early Action Recognition, Motion Synthesis, and Human Geometric Modeling.

Peter Youngs is a professor in the Department of Curriculum, Instruction and Special Education at the University of Virginia. He studies how neural networks can be used to automatically classify instructional activities in videos of elementary mathematics and reading instruction. He currently serves as co-editor of American Educational Research Journal.

Scott T. Acton is a Professor and Chair of Electrical and Computer Engineering at the University of Virginia. He is also appointed in Biomedical Engineering. For the last three years, he was Program Director in the Computer and Information Sciences and Engineering directorate of the National Science Foundation. He received the M.S. and Ph.D. degrees at the University of Texas at Austin, and he received his B.S. degree at Virginia Tech. Professor Acton is a Fellow of the IEEE "for contributions to biomedical image analysis." Professor Acton's laboratory at UVA is called VIVA - Virginia Image and Video Analysis. They specialize in biological/biomedical image analysis problems. The research emphases of VIVA include image analysis in neuroscience, tracking, segmentation, representation, retrieval, classification and enhancement. Recent theoretical interests include machine learning, active contours, partial differential equation methods, scale space methods, and graph signal processing. Professor Acton has over 300 publications in the image analysis area including the books Biomedical Image Analysis: Tracking and Biomedical Image Analysis: Segmentation. He was the 2018 Co-Chair of the IEEE International Symposium on Biomedical Imaging. Professor Acton was Editor-in-Chief of the IEEE Transactions on Image Processing (2014-2018)