Egocentric Early Action Prediction via Multimodal Transformer-Based Dual Action Prediction

Weili Guan, *Member, IEEE*, Xuemeng Song, *Senior Member, IEEE*, Kejie Wang, Haokun Wen, Hongda Ni, Yaowei Wang, *Member, IEEE*, and Xiaojun Chang, *Senior Member, IEEE*

***Abstract*— Egocentric early action prediction, which aims to recognize the on-going action in the video captured in the first-person view as early as possible before the action is fully executed, is a new yet challenging task due to the limited partial video input. Pioneer studies focused on solving this task with LSTMs as the backbone and simply compiling the observed video segment and unobserved video segment into a single vector, which hence suffer from two key limitations: lack the non-sequential relation modeling with the video snippet sequence and the correlation modeling between the observed and unobserved video segment. To address these two limitations, in this paper, we propose a novel multimodal TransfoRmer-based duAl aCtion prEdiction (mTRACE) model for the task of egocentric early action prediction, which consists of two key modules: the early (observed) segment action prediction module and the future (unobserved) segment action prediction module. Both modules take Transformer encoders as the backbone for encoding all the potential relations among the input video snippets, and involve several single-modal and multi-modal classifiers for comprehensive supervision. Different from previous work, each of the two modules outputs two multi-modal feature vectors: one for encoding the current input video segment, and the other one for predicting the missing video segment. For optimization, we design a two-stage training scheme, including the mutual enhancement stage and end-to-end aggregation stage. The former stage alternatively optimizes the two action prediction modules, where the correlation between the observed and unobserved video segment is modeled with a consistency regularizer, while the latter seamlessly aggregates the two modules to fully utilize the capacity of the two modules. Extensive experiments have demonstrated the superiority of our proposed model. We have released the codes and the corresponding parameters to benefit other researchers at https://trace729.wixsite.com/trace.**

***Index Terms*— Egocentric early action prediction, transformer, mutual enhancement.**

I. Introduction

VIDEO action recognition is an essential research problem in computer vision domain, which aims to classify the action of the person in the video segment. Early studies [1], [2], [3] focus on the action recognition for videos recorded in the third-person view. Later, with the rapid development of wearable devices, increasing research efforts have been dedicated to the action recognition [4], [5], [6], [7] for videos recorded in the first-person view, *i.e.,* egocentric action recognition. Moreover, in many real-world scenarios, like autonomous driving [8], [9], we may expect to recognize the action as early as possible. Accordingly, several pioneer studies have paid attention to the new task of egocentric early action prediction [10], [11], [12]. As shown in Figure 1, different from the egocentric action recognition whose input is the fully observed video segment, egocentric early action prediction aims to recognize the on-going action in the video captured in the first-person view as early as possible. In other words, the input for egocentric early action prediction is the partial video segment with incomplete action execution.

Although the pioneer research studies [13], [14], [15], [16] on egocentric early action prediction have achieved promising progress, they suffer from two key limitations.

- **L1: Lack the non-sequential relation modeling among video snippets.** Existing studies mainly utilize LSTMs to encode the video segment, which can capture the temporal sequential relation modeling among video snippets. Nevertheless, LSTMs cannot model the non-sequential relations among video snippets well. For example, there may be some correlations among discrete video snippets at different time steps. Such correlations could be hard to be captured by LSTMs.
- **L2: Lack the correlation modeling between the observed and unobserved video segment.** Although existing studies have incorporated the future (unobserved) video segment to enhance the representation learning of early (observed) video segment in the training phrase, they entangled the clues in both observed and unobserved segments into a single vector, and overlook the correlation modeling between the two video segments. Intuitively,

Manuscript received 5 January 2023; revised 9 February 2023; accepted 21 February 2023. Date of publication 23 February 2023; date of current version 6 September 2023. This work was supported in part by the Shandong Provincial Natural Science Foundation under Grant ZR2022YQ59 and in part by the Australian Research Council (ARC) Discovery Early Career Researcher Award (DECRA) under Grant DE190100626. This article was recommended by Associate Editor J. Wu. *(Corresponding author: Xuemeng Song.)*

Weili Guan is with the Department of Data Science and Artificial Intelligence, Monash University, Clayton, VIC 3800, Australia, and also with the Peng Cheng Laboratory, AI Research Center, Shenzhen 518055, China (e-mail: honeyguan@gmail.com).

Xuemeng Song and Kejie Wang are with the School of Computer Science and Technology, Shandong University, Tsingtao 266000, China (e-mail: sxmustc@gmail.com; kjwang.henry@gmail.com).

Haokun Wen and Hongda Ni are with the School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China (e-mail: whenhaokun@gmail.com; nihongda.hit@gmail.com).

Yaowei Wang is with the Peng Cheng Laboratory, AI Research Center, Shenzhen 518055, China (e-mail: wangyw@pcl.ac.cn).

Xiaojun Chang is with the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: cxj273@gmail.com).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2023.3248271.

Digital Object Identifier 10.1109/TCSVT.2023.3248271

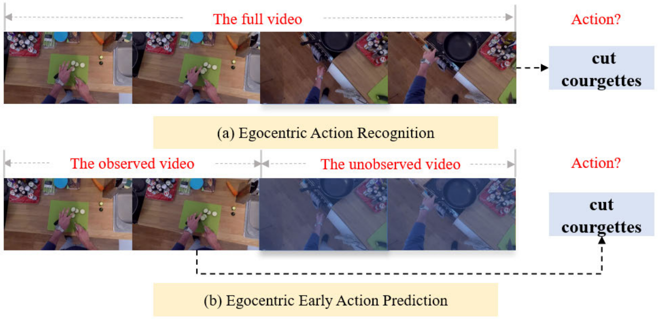


Fig. 1. Comparison between the tasks of egocentric action recognition and egocentric early action prediction.

we can predict the person's following behavior in the unobserved video segment based on the observed video segment, and vice versa.

To address these limitations, we propose a novel multimodal TransfoRmer-based duAl aCtion prEdiction (mTRACE) model for the task of egocentric early action prediction, which consists of two key modules: the early segment action prediction module and the future segment action prediction module. The former aims to predict the person's action based on the early observed video segment, while the latter targets predicting that based on the future unobserved video segment. Both modules share the same architecture. Specially, each module is composed by a Transformer-based video encoder and several single-modal and multi-modal action classifiers. Moreover, the video encoder in each module consists of M Transformer subencoders corresponding to the single-modal feature encoding, as well as two fully connected layers for deriving two multi-modal features for encoding the current input video segment and the missing video segment, respectively. The motivation for using Transformer lies in its powerful capability in the non-sequential relation modeling in many computer vision tasks [17], [18]. For optimization, we design a two-stage training paradigm that optimizes the model through two stages: mutual enhancement stage and end-to-end aggregation stage. The goal of the former stage is to alternatively train the early and future segment action prediction modules by allowing the knowledge transfer between the two modules, while that of the latter stage is to seamlessly aggregate the two modules to promote the final model performance.

Our main contributions can be summarized threefold:

- We present a novel multimodal Transformer-based dual action prediction model for the task of egocentric early action prediction. To the best of our knowledge, we are the first to incorporate Transformer architecture to enhance the performance of egocentric early action prediction.
- We devise a dual action prediction model, where we take into account the coherence of the full video segment and assume that the feature of the future video segment can be inferred based on the early video segment, while that of the early video segment can be also inferred by mining the future video segment.
- We design a two-stage optimization scheme, including the mutual enhancement stage and end-to-end aggregation stage. The former stage allows the knowledge transferring between the two action prediction modules, while the latter one fully utilizes the capacity of the two modules. Extensive experiments have demonstrated the superiority of our proposed model.

II. Related Works

Our work is related to early action prediction and Transformer in vision.

A. Early Action Prediction

According to the manner of the video being recorded, existing early action prediction studies can be generally classified into two groups: early action prediction in the third-person view [3], [11], [12], [19], [20], and that in the first-person view. Previous studies mainly focus on the former group. For example, Kong et al. [12], [20] exploited the abundant sequential context information to enrich the feature representations of the given partial videos in the context of early action prediction. In addition, Wang et al. [10] presented a teacher-student learning framework that distills progressive knowledge from an action recognition network for an early action prediction network. Moreover, Cai et al. [11] resorted to transferring knowledge from full videos to partial videos by a two-stage learning framework, which learns the feature embeddings and action classifier based on the full videos in the first stage and then transfers the knowledge obtained by the first stage to the counterpart (*i.e.,* the feature embedding and action classifier learning based on the partial videos) in the second stage. Later, owing to the rapid development of multi-sensor wearable computing platforms, several research efforts have been dedicated to the egocentric early action prediction, where the input video is recorded in the first-person view. Initially, the egocentric early action prediction task was introduced by Furnari and Farinella [13], and the authors proposed a Rolling-Unrolling LSTM architecture [14], which can predict egocentric actions at multiple temporal scales, to solve this task. Recently, Zheng et al. [15] presented an adversarial knowledge distillation scheme for the task of egocentric early action prediction, which also adopts a teacher network for learning the comprehensive video representation based on the full video segment, and a student network for predicting the action only based on the partial video segment.

Although great success has been achieved by these studies, existing methods have two key limitations: lack the non-sequential relation modeling among video snippets, and lack the correlation modeling between the observed and unobserved video segments. To address them, we propose the mTRACE model, which incorporates the Transformer architecture to enhance the correlation modeling among video snippets, and adopt dual action prediction models to promote the correlation modeling between the observed and unobserved video segments for boosting the egocentric early action prediction performance.

B. Transformer in Vision

The Transformer is a deep learning model that adopts the self-attention mechanism, adaptively weighting the significance of each part of the input data. It was first proposed in the field of natural language processing (NLP) [21]. Inspired by its exemplary performance on representation learning, it has also been widely used in various computer vision tasks, such as image recognition [22], [23], image captioning [24], [25], object detection [26], [27], and video understanding [17], [18], [28]. For instance, Zhao et al. [27] proposed a Transformer-based vote refinement model to cultivate the voting results and improve the performance of 3D object detection. In addition, Yu et al. [24] introduced the multi-modal Transformer model to deal with the multi-view visual representations, which highly improves the image captioning performance. Besides, Girdhar and Grauman [18] presented an end-to-end attention-based video Transformer for predicting the future actions given previously observed video. Although the Transformer has made outstanding achievements in these tasks, it has never been explored in the task of egocentric early action prediction. Therefore, in this work, we incorporate the Transformer to enhance the performance of egocentric early action prediction.

III. Methodology

In this section, we first formulate the research problem, and then detail the components of our proposed model.

A. Problem Formulation

In this work, we focus on tackling the problem of egocentric early action prediction. It aims to predict the person's action by observing only a few early video frames from the first-person view. Formally, suppose $\mathcal{V} = \{V_1, V_2, \ldots, V_K\}$ is an egocentric video segment consisting of K snippets, and $\mathbf{y} \in \mathbb{R}^N$ denotes the one-hot action label vector of the video segment, where N is the number of total action label classes. In particular, we employ the first k snippets $\mathcal{V}_k = \{V_1, V_2, \ldots, V_k\}$ as the given partial video segment, where $k \in [1, K]$. We use k/K to denote the observation ratio. Each snippet V_k involves M modalities, such as audio signals, visual content, flow content and the object labels. Let $\mathbf{x}_m^j \in \mathbb{R}^{d_0^m}$ denote the feature vector of the j-th snippet regarding the m-th modality, where $j = 1, \cdots, K$. In this work, we aim to learn a model $\mathcal{G}(\cdot)$ to predict the action class for a given partial video segment. Mathematically, this task can be formulated as follows,

$$\hat{\mathbf{y}} = \mathcal{G}(\mathcal{X}_1^k, \mathcal{X}_2^k, \cdots, \mathcal{X}_M^k|\Theta), \tag{1}$$

where $\mathcal{X}_m^k = \{\mathbf{x}_m^1, \mathbf{x}_m^2, \cdots, \mathbf{x}_m^k\}$ denotes the feature sequence of the partial video segment $\mathcal{V}_k$ regarding the m-th modality. Θ is the to-be-learned parameters of the model $\mathcal{G}(\cdot)$ and $\hat{\mathbf{y}} \in \mathbb{R}^N$ is predicted action label vector by the model $\mathcal{G}(\cdot)$.

B. Summary

In this work, we propose a Transformer-based dual action prediction model for egocentric early action prediction, which consists of two key modules: the early segment action prediction module $\mathcal{P}^o$ and future segment action prediction module $\mathcal{P}^u$. The former takes the given partial video segment as the input, while the latter takes the future video segment as the input. These two modules share the same network architecture. One major novelty of our model is that we take into account the coherence of full video segment, and assume that the feature of the future video segment can be inferred based on the early video segment, while that of early video segment can be also inferred by mining the future video segment. Accordingly, each module outputs two multi-modal features: one used for encoding the current input video segment, and the other one used for predicting the missing video segment. Notably, to adapt the Transformer encoder to our context, we adopt the autoregressive pre-training for each Transformer encoder.

For the model optimization, we propose the two-stage training paradigm, which optimizes the model through two stages: mutual enhancement stage and end-to-end aggregation stage. In the former stage, the early and future segment action prediction modules learn from each other, so that both of them can infer the missing part of video and hence gain a comprehensive understanding of the video. In the latter stage, these two modules are seamlessly aggregated to deal with the task of egocentric early action prediction.

C. Network Structure

Next, we describe the two key components of each module: Transformer-based video encoder and action classifiers.

1) Transformer-Based Video Encoder: Since the underlying philosophy for the design of the two modules is similar, we here take the video encoder of the early segment action prediction module $\mathcal{P}^o$ as an example.

Existing methods on egocentric early action prediction mainly adopted LSTMs to encode the sequence of the video snippets. Nevertheless, it is known that LSTMs cannot capture the long-term temporal dependencies well and can only model the sequential relations among video snippets. Beyond that, we resort to Transformer, which has shown great success in many computer vision tasks [22], [24], [26], [29]. Different from LSTMs, Transformer can flexibly model the relationship among different video snippets with the multi-head self-attention mechanism. In particular, we introduce a specific Transformer to encode each modality of the video snippets. Formally, let $\mathcal{T}_m^o$ denote the Transformer encoder of the m-th modality which consists of several identical layers. Each layer has two sub-layers: one corresponds to a multi-head self-attention mechanism and the other one refers to a feed-forward network. Then we feed the feature sequence of the k snippets in the observed video segment regarding the m-th modality $\{\mathbf{x}_m^1, \mathbf{x}_m^2, \ldots, \mathbf{x}_m^k\}$ into it. For each modality, we also introduce a to-be-learned [CLS] token $\mathbf{x}_m^{[\mathrm{CLS}],o}$ to aggregate the global feature of the m-th modality. According to Transformer, we add the positional embedding $\mathbf{p}_m^j \in \mathbb{R}^{d_0^m}$, where $j = 0, \cdots, k$, to the [CLS] token along with each feature vector. Mathematically, we have,

$$\mathbf{z}_{m,k}^o = \mathcal{T}_m^o([\mathbf{x}_m^{[\mathrm{CLS}],o} + \mathbf{p}_m^0; \mathbf{x}_m^1 + \mathbf{p}_m^1; \ldots; \mathbf{x}_m^k + \mathbf{p}_m^k]), \tag{2}$$

where $\mathbf{x}_m^{[\mathrm{CLS}],o} \in \mathbb{R}^{d_0^m}$ denotes the [CLS] token of the m-th modality. $\mathbf{z}_{m,k}^o \in \mathbb{R}^{d_1}$ is the encoded feature of the given early

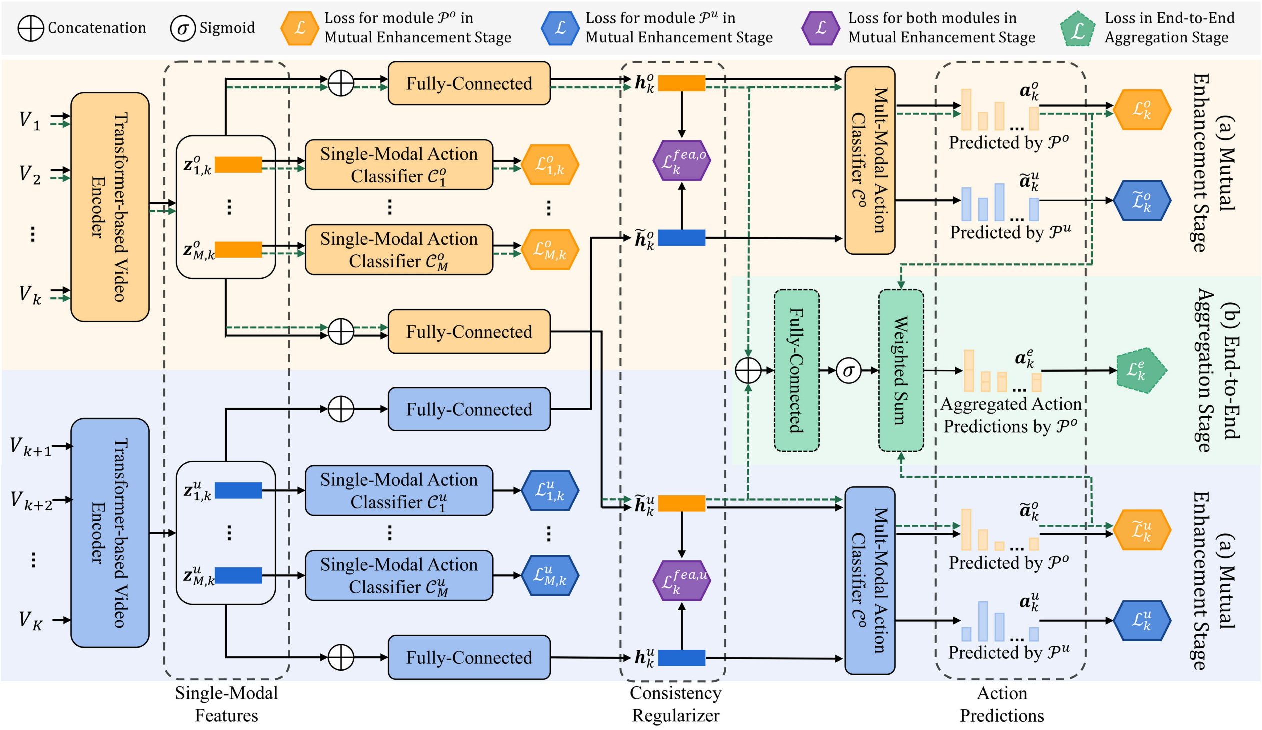

Concatenation
Sigmoid
Loss for module $\mathcal{P}^o$ in Mutual Enhancement Stage
Loss for module $\mathcal{P}^u$ in Mutual Enhancement Stage
Loss for both modules in Mutual Enhancement Stage
Loss in End-to-End Aggregation Stage
V_1
V_2
V_k
Transformer-based Video Encoder
$z^o_{1,k}$
$z^o_{M,k}$
Fully-Connected
Single-Modal Action Classifier $\mathcal{C}^o_1$
Single-Modal Action Classifier $\mathcal{C}^o_M$
$\mathcal{L}^o_{1,k}$
$\mathcal{L}^o_{M,k}$
h^o_k
$\mathcal{L}^{fea,o}_k$
$\tilde{h}^o_k$
Mult-Modal Action Classifer $\mathcal{C}^o$
a^o_k
Predicted by $\mathcal{P}^o$
$\tilde{a}^u_k$
Predicted by $\mathcal{P}^u$
$\mathcal{L}^o_k$
$\tilde{\mathcal{L}}^o_k$
(a) Mutual Enhancement Stage
Fully-Connected
Weighted Sum
a^e_k
Aggregated Action Predictions by $\mathcal{P}^o$
$\mathcal{L}^e_k$
(b) End-to-End Aggregation Stage
V_{k+1}
V_{k+2}
V_K
Transformer-based Video Encoder
$z^u_{1,k}$
$z^u_{M,k}$
Single-Modal Action Classifier $\mathcal{C}^u_1$
Single-Modal Action Classifier $\mathcal{C}^u_M$
$\mathcal{L}^u_{1,k}$
$\mathcal{L}^u_{M,k}$
$\tilde{h}^u_k$
$\mathcal{L}^{fea,u}_k$
h^u_k
Mult-Modal Action Classifer $\mathcal{C}^o$
$\tilde{a}^o_k$
Predicted by $\mathcal{P}^o$
a^u_k
Predicted by $\mathcal{P}^u$
$\tilde{\mathcal{L}}^u_k$
$\mathcal{L}^u_k$
(a) Mutual Enhancement Stage
Single-Modal Features
Consistency Regularizer
Action Predictions

Fig. 2. Illustration of the proposed mTRACE model, which consists of two key modules: the early segment action prediction module and future segment action prediction module. The proposed model is optimized through two stages: mutual enhancement stage and end-to-end aggregation stage.

video segment $\mathcal{V}_k$ regarding the m-th modality, and d_1 is the dimension of the hidden layer.

We then concatenate all the encoded features of the M modalities and use a fully-connected layer to derive the final encoded feature of the given early video segment as follows,

$$\mathbf{h}^o_k = \mathbf{W}^o\mathbf{z}^o_k + \mathbf{b}^o, \tag{3}$$

where $\mathbf{z}^o_k = [\mathbf{z}^o_{1,k}; \cdots; \mathbf{z}^o_{M,k}] \in \mathbb{R}^{M*d_1}$ is the concatenation of all the encoded features. $\mathbf{W}^o$ and $\mathbf{b}^o$ are the to-be-learned parameters of the fully-connected layer. $\mathbf{h}^o_k \in \mathbb{R}^{d_2}$ is the final encoded feature of the early video segment, where d_2 is the final encoded feature dimension.

Previous work on egocentric early action prediction focuses on encoding the given partial video segment into a single vector. Most of them rely on the single vector to capture the information of the observed video segment. Obviously, this manner would miss the information of the future video segment, which undoubtedly benefits the action prediction of the video. Notably, although the future video segment is unavailable in testing, it can be still used in the training phrase. It is worth mentioning some efforts [15] have been dedicated to utilize the future video segment in the training phrase with the knowledge distillation technique. However, they also use a single vector to compile both the observed and unobserved video segments. Different from previous studies, we propose to encode the partial video with two separate features: one used for delivering the content of the observed partial video segment, and the other one used for indicating the content of the unobserved future video segment. In this manner, the features for the two video segments can be disentangled, which could benefit the feature encoding. Therefore, we introduce another fully connected layer parameterized by $\tilde{\mathbf{W}}^o$ and $\tilde{\mathbf{b}}^o$ to predict the feature of the future video segment based on the observed early video segment as follows,

$$\tilde{\mathbf{h}}^u_k = \tilde{\mathbf{W}}^o\mathbf{z}^o_k + \tilde{\mathbf{b}}^o, \tag{4}$$

where $\tilde{\mathbf{h}}^u_k \in \mathbb{R}^{d_2}$ is the predicted future feature for the given partial video segment.

Similar to the structure of $\mathcal{P}^o$, the video encoder of the future video processing module $\mathcal{P}^u$ is composed of M Transformers to encode the M different modalities, and two fully-connected layers for learning the features of the previous video segment and the unobserved future video segment, respectively. Specifically, we feed the future video segment into $\mathcal{P}^u$ and have:

$$\begin{cases} \mathbf{X}^u_m = [\mathbf{x}^{[\text{CLS}],\text{u}}_m + \mathbf{p}^0_m; \mathbf{x}^{k+1}_m + \mathbf{p}^1_m; \ldots; \mathbf{x}^K_m + \mathbf{p}^{K-k}_m] \\ \mathbf{z}^u_{m,k} = \mathcal{T}^u_m(\mathbf{X}^u_m) \\ \mathbf{z}^u_k = [\mathbf{z}^u_{1,k}; \ldots; \mathbf{z}^u_{M,k}], \\ \mathbf{h}^u_k = \mathbf{W}^u\mathbf{z}^u_k + \mathbf{b}^u, \\ \tilde{\mathbf{h}}^o_k = \tilde{\mathbf{W}}^u\mathbf{z}^u_k + \tilde{\mathbf{b}}^u, \end{cases} \tag{5}$$

where $\mathcal{T}^u_m$ is the Transformer for encoding the m-th modality of the unobserved video segment, $\mathbf{X}^u_m$ is the input to $\mathcal{T}^u_m$, $\mathbf{x}^{[\text{CLS}],\text{u}}_m \in \mathbb{R}^{d^m_0}$ denotes the [CLS] token of the m-th modality, used for aggregating the global feature of this modality, $\mathbf{z}^u_{m,k} \in \mathbb{R}^{d_1}$ is the encoded feature of the modality m for the unobserved segments. $\mathbf{W}^u, \mathbf{b}^u, \tilde{\mathbf{W}}^u$ and $\tilde{\mathbf{b}}^u$ are the to-be-learned parameters of two fully-connected layers. $\mathbf{h}^u_k \in \mathbb{R}^{d_2}$ and $\tilde{\mathbf{h}}^o_k \in \mathbb{R}^{d_2}$ are the encoded feature of the future unobserved video segment and predicted feature of the previous observed video segment, respectively.

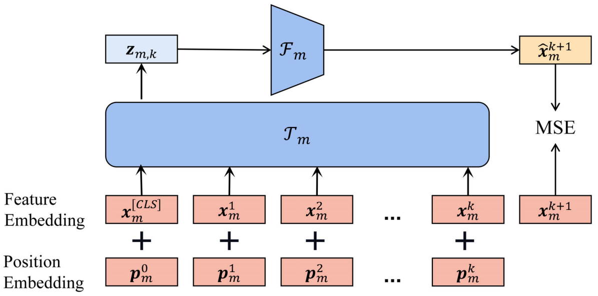


Fig. 3. Pre-training for the Transformer encoder regarding the m-th modality.

Pre-Training: To adapt the Transformer encoder to our egocentric early action prediction task and further improve the model performance, we introduce the autoregressive pre-training [18], [30] to particularly improve the feature encoding capability of the Transformer encoders in our two modules. Figure 3 illustrates the pre-training paradigm for the Transformer encoder regarding the m-th modality. Specifically, for each Transformer encoder $\mathcal{T}_m, m = 1, \cdots, M$, we expect it can predict the feature of m-th modality for the next snippet based on the previous k snippets of the video sequence. Hence the Transformer encoder can fully understand the context of the video snippet sequence.

Mathematically, the pre-training process can be formulated as follows,

$$\mathbf{z}_{m,k} = \mathcal{T}_m([\mathbf{x}_m^{[\mathrm{CLS}]} + \mathbf{p}_m^0; \mathbf{x}_m^1 + \mathbf{p}_m^1; \ldots; \mathbf{x}_m^k + \mathbf{p}_m^k]), \quad (6)$$

where $\mathbf{x}_m^{[\mathrm{CLS}]} \in \mathbb{R}^{d_0^m}$ is the [CLS] token feature of the modality m, used to aggregate the information of the entire video sequence. $\mathbf{x}_m^i \in \mathbb{R}^{d_0^m}$, $i \in [1, k]$ is the feature of the i-th video snippet with respect to the m-th modality. $\mathbf{p}_m^i \in \mathbb{R}^{d_0^m}$, $i \in [0, k]$ is the positional embedding. $\mathbf{z}_{m,k} \in \mathbb{R}^{d_1}$ is the encoded feature for the input partial video segment (*i.e.,* the first k snippets).

We then use a fully-connected layer $\mathcal{F}_m : \mathbb{R}^{d_1} \mapsto \mathbb{R}^{d_0^m}$ to predict the feature of the next snippet (*i.e.,* $\mathbf{x}_m^{k+1}$) based on the encoded feature $\mathbf{z}_{m,k}$ of the previous k snippets. Finally, we adopt the L2 norm loss function for optimization as follows,

$$\begin{cases} \hat{\mathbf{x}}_m^{k+1} = \mathbf{W}_m^{pre}\mathbf{z}_{m,k} + \mathbf{b}_m^{pre}, \quad m = 1, \cdots, M \\ \mathcal{L}_m^{pre} = \sum_{k=1}^{K-1} \|\hat{\mathbf{x}}_m^{k+1} - \mathbf{x}_m^{k+1}\|^2, \end{cases} \quad (7)$$

where $\mathbf{W}_m^{pre}$ and $\mathbf{b}_m^{pre}$ are parameters of the fully-connected layer. $\hat{\mathbf{x}}_m^{k+1}$ is the predicted feature of the next snippet.

The learned parameters of these M Transformer encoders will be used for the parameter initialization. Specifically, for each modality m, the Transformer encoder $\mathcal{T}_m^o$ in the early segment action prediction module and $\mathcal{T}_m^u$ in the future segment action prediction module are both initialized with parameters of $\mathcal{T}_m$ to enhance their encoding capabilities.

2) Action Classifier: In order to ensure the features extracted by $\mathcal{P}^o$ can contain discriminant information for action classification, we introduce a classifier $\mathcal{C}^o$ with the cross-entropy loss function as follows,

$$\begin{cases} \mathbf{a}_k^o = \mathbf{W}^o\mathbf{h}_k^o + \mathbf{b}^o, \\ \mathcal{L}_k^o = \mathrm{CE}(\mathbf{a}_k^o, \mathbf{y}), \end{cases} \quad (8)$$

where $\mathbf{W}^o$ and $\mathbf{b}^o$ are the to-be-learned parameters of the classifier. $\mathbf{a}_k^o \in \mathbb{R}^N$ is the predicted action label vector based on the learned final feature of the observed video segment.

Similarly, for the future segment action prediction module $\mathcal{P}^u$, we also introduce a classifier $\mathcal{C}^u$ as follows,

$$\begin{cases} \mathbf{a}_k^u = \mathbf{W}^u\mathbf{h}_k^u + \mathbf{b}^u, \\ \mathcal{L}_k^u = \mathrm{CE}(\mathbf{a}_k^u, \mathbf{y}), \end{cases} \quad (9)$$

where $\mathbf{W}^u$ and $\mathbf{b}^u$ are the to-be-learned parameters of the classifier. $\mathbf{a}_k^u \in \mathbb{R}^N$ is the predicted action label vector based on the learned final feature of the future video segment.

As to supervise the feature prediction of the future video segment by $\mathcal{P}^o$, we feed the predicted feature of the future video segment into the action classifier of $\mathcal{P}^u$, and use cross-entropy loss for optimization. Conversely, we also feed the predicted feature of the previous video segment by the future segment action prediction module into the action classifier of early segment action prediction module for optimization. Formally, we have,

$$\begin{cases} \tilde{\mathbf{a}}_k^u = \mathbf{W}^u\tilde{\mathbf{h}}_k^u + \mathbf{b}^u, \\ \tilde{\mathbf{a}}_k^o = \mathbf{W}^o\tilde{\mathbf{h}}_k^o + \mathbf{b}^o, \\ \tilde{\mathcal{L}}_k^u = \mathrm{CE}(\tilde{\mathbf{a}}_k^u, \mathbf{y}), \\ \tilde{\mathcal{L}}_k^o = \mathrm{CE}(\tilde{\mathbf{a}}_k^o, \mathbf{y}). \end{cases} \quad (10)$$

Moreover, to enhance the discriminative feature learning of each Transformer-based encoder in each module, we also introduce a classifier $\mathcal{C}_m^o$ for each modality. For the early segment action prediction module, utilizing the cross-entropy loss, we have,

$$\begin{cases} \mathbf{a}_{m,k}^o = \mathbf{W}_m^o\mathbf{z}_{m,k}^o + \mathbf{b}_m^o, \\ \mathcal{L}_{m,k}^o = \mathrm{CE}(\mathbf{a}_{m,k}^o, \mathbf{y}), \end{cases} \quad (11)$$

where $\mathbf{a}_{m,k}^o \in \mathbb{R}^N$ is the predicted action label vector of the early video segment based on its m-th modality. $\mathbf{W}_m^o$ and $\mathbf{b}_m^o$ are the to-be-learned parameters of the action classifier for the m-th modality.

Similarly, for the future segment action prediction module, we introduce M classifiers (denoted as $\mathcal{C}_m^u$) and loss function,

$$\begin{cases} \mathbf{a}_{m,k}^u = \mathbf{W}_m^u\mathbf{z}_{m,k}^u + \mathbf{b}_m^u, \\ \mathcal{L}_{m,k}^u = \mathrm{CE}(\mathbf{a}_{m,k}^u, \mathbf{y}), \end{cases} \quad (12)$$

where $\mathbf{a}_{m,k}^u \in \mathbb{R}^N$ is the predicted action label vector of the future video segment based on its m-th modality. $\mathbf{W}_m^u$ and $\mathbf{b}_m^u$ are the to-be-learned parameters.

D. Two-Stage Training

As aforementioned, we train our model with two stages: mutual enhancement stage and end-to-end aggregation stage. In the former stage, the two modules $\mathcal{P}^o$ and $\mathcal{P}^u$ learn from each other for gaining predictive ability iteratively. In the latter stage, we aim to fully integrate the prediction capability of the two modules.

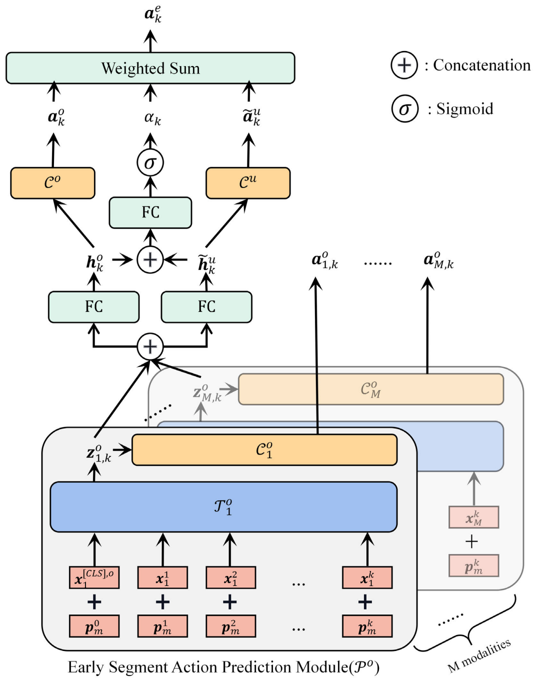


Fig. 4. Illustration of the end-to-end aggregation stage, where the early segment action prediction module and the future segment action prediction module are seamlessly aggregated for the final action prediction.

1) Mutual Enhancement Stage: As aforementioned, $\mathcal{P}^o$ outputs not only the observed feature but also the predicted unobserved feature, while $\mathcal{P}^u$ outputs not only the future video feature but also the predicted feature of the previous early video segment. Intuitively, the learned features of these two modules can be mutually used for guiding each other. Specifically, $\mathbf{h}_k^o$ learned by $\mathcal{P}^o$ based on the input early observed video segment can be used for guiding the learning of $\tilde{\mathbf{h}}_k^o$ predicted by $\mathcal{P}^u$ based on the future video segment. Conversely, $\mathbf{h}_k^u$ derived by $\mathcal{P}^u$ can be used for supervising the learning of $\tilde{\mathbf{h}}_k^u$ predicted by $\mathcal{P}^o$. Thus, we introduce the consistency regularizer to allow the two modules to share knowledge to each other and gain better action prediction capability. The regularizer can be written as follows,

$$\mathcal{L}_k^{fea} = \begin{cases} \|\mathbf{h}_k^o - \tilde{\mathbf{h}}_k^o\|^2 + \|\mathbf{h}_k^u - \tilde{\mathbf{h}}_k^u\|^2, & 0 < k < K, \\ 0, & otherwise. \end{cases} \tag{13}$$

Intuitively, only when both early video segment and future video segment are valid (*i.e.*, $0 < k < K$), we can conduct the consistency regularization. Otherwise, the consistency regularizer will be removed.

Ultimately, we obtain the final loss function of the early segment action prediction module $\mathcal{P}^o$ in the mutual enhancement stage as follows,

$$\mathcal{L}_{mut}^o = \sum_{k=1}^{K}(\frac{1}{M}\sum_{m=1}^{M}\mathcal{L}_{m,k}^o + \mathcal{L}_k^o + \tilde{\mathcal{L}}_k^u + w^{fea}\mathcal{L}_k^{fea}), \tag{14}$$

where w^{fea} is the hyper-parameter.

Algorithm 1 The Two-Stage Training Procedure of Our Model

Input: Training set Ω, hyper-parameter w^{fea}.
Output: Parameters Θ^o in P^o, parameters Θ^u in P^u, weight $\mathbf{W}^e$ and bias $\mathbf{b}^e$.

```
Initialize $\mathcal{T}_m^o$ and $\mathcal{T}_m^u$ with the pretrained parameters for each modality $m$.
repeat
    Sample minibatch from $\Omega$.
    Freeze the parameters $\Theta^u$ and update the parameters $\Theta^o$ according to $\mathcal{L}_{mut}^o$ in Eqn.(14).
    Freeze the parameters $\Theta^o$ and update the parameters $\Theta^u$ according to $\mathcal{L}_{mut}^u$ in Eqn.(15).
until Convergence
Freeze the parameters in $\mathcal{T}_m^o$ for each modality $m$.
repeat
    Sample minibatch from $\Omega$.
    Update the parameters $\Theta^o$, $\mathbf{W}^e$ and $\mathbf{b}^e$ according to $\mathcal{L}_{agg}$ in Eqn.(19).
until Convergence
```

Similarly, we can obtain the total loss of future segment action prediction module $\mathcal{P}^u$ as follows,

$$\mathcal{L}_{mut}^u = \sum_{k=0}^{K-1}(\frac{1}{M}\sum_{m=1}^{M}\mathcal{L}_{m,k}^u + \mathcal{L}_k^u + \tilde{\mathcal{L}}_k^o + w^{fea}\mathcal{L}_k^{fea}). \tag{15}$$

Notably, inspired by the mutual learning framework, these two modules are optimized alternatively. For each batch, we first freeze $\mathcal{P}^u$ and only train the early segment action prediction module $\mathcal{P}^o$ with loss $\mathcal{L}_{mut}^o$. Thereafter, we freeze $\mathcal{P}^o$ and only train the future segment action prediction module $\mathcal{P}^u$ with loss $\mathcal{L}_{mut}^u$. Ultimately, by iterative learning from each other, both two modules can gain the discriminative feature of the full video segment, although their inputs are incomplete.

2) End-to-End Aggregation Stage: In the former stage, the two modules $\mathcal{P}^o$ and $\mathcal{P}^u$ iteratively learn from each other for improving their action prediction abilities, respectively. Specifically, the early segment action prediction module should be good at predicting the person's action based on the given partial video segment, while the future segment action prediction module should be skilled in predicting the person's action based on the unobserved future video segment. Although the unobserved future video segment is unavailable in the testing phase, we can predict its feature $\tilde{\mathbf{h}}_k^u$ based on $\mathcal{P}^o$. Accordingly, the future segment action prediction module can enhance the final action prediction in the testing phase by providing the action prediction based on the predicted feature of the unobserved future video segment output by $\mathcal{P}^o$. Therefore, in the end-to-end aggregation stage, we aim to fully integrate the prediction capability of the two modules.

In addition, we believe that early observed video and future unobserved video segments have different levels of importance for the action prediction. We hence add a fully-connected layer with the sigmoid activation function to learn the trade-off weights between the early action prediction and future action prediction results. In particular, the trade-off weight for the given partial video segment can be obtained as follows,

$$\alpha_k = \text{sigmoid}(\mathbf{W}^e[\mathbf{h}_k^o; \tilde{\mathbf{h}}_k^u] + \mathbf{b}^e), \tag{16}$$

where $\mathbf{W}^e$ and $\mathbf{b}^e$ are the to-be-learned parameters, and α_k represents the importance of early action prediction.

We then can get the final action prediction result for the given partial video segment as follows,

$$\mathbf{a}_k^e = \alpha_k \mathbf{a}_k^o + (1 - \alpha_k)\tilde{\mathbf{a}}_k^u, \tag{17}$$

where $\mathbf{a}_k^e \in \mathbb{R}^N$ denotes the final action prediction.

In this stage, we also utilize the cross entropy loss for optimization as follows,

$$\mathcal{L}_k^e = \text{CE}(\mathbf{a}_k^e, \mathbf{y}). \tag{18}$$

Similar to the mutual enhancement stage, we also consider the classification loss of each single modality to improve the discriminative feature learning. Finally, the loss function of the end-to-end aggregation stage can be written as follows,

$$\mathcal{L}_{agg} = \sum_{k=1}^{K}(\mathcal{L}_k^e + \frac{1}{M}\sum_{m=1}^{M}\mathcal{L}_{m,k}^o). \tag{19}$$

It is worth noting that in this training stage, we will fix the backbone parameters of the action prediction module, but only optimize the aggregation parameters. In this manner, the knowledge obtained from the previous training stage can be well retained. Algorithm 1 summarizes the two-stage training procedure of our model.

E. Testing

For testing, to enhance the action prediction, apart from the final action prediction result $\mathbf{a}_k^e$ in Eqn. (17), we also take into account the action prediction result based on each single modality (*i.e.,* $\mathbf{a}_{m,k}^o$). Specifically, we generate the final action prediction result by the linear fusion as follows,

$$\begin{cases} \mathbf{a}_k = \gamma \mathbf{a}_k^e + \sum_{m=1}^{M} \gamma_m \mathbf{a}_{m,k}^o, \\ \gamma + \sum_{m=1}^{M} \gamma_m = 1, \end{cases} \tag{20}$$

where $\mathbf{a}_k \in \mathbb{R}^N$ is the final action prediction result for testing. γ and γ_m $(m = 1, \cdots, M)$ are the trade-off hyper-parameters.

IV. Experiments

In this section, we conducted extensive experiments over the two real-world datasets EPIC-Kitchens55 [31] and EGTEA Gaze+ [32] by answering the following research questions.

- **RQ1**: Does our model outperform existing methods?
- **RQ2**: How does each component affect our model?
- **RQ3**: How sensitive is our model to the key hyperparameters?
- **RQ4**: What is the intuitive performance of our method?

TABLE I

Three Dataset Splits of EGTEA Gaze+

Dataset	Training sets	Testing sets
EGTEA Gaze+ S1	8, 299	2, 022
EGTEA Gaze+ S2	8, 299	2, 022
EGTEA Gaze+ S3	8, 230	2, 021

A. Experimental Settings

1) Datasets: To evaluate our proposed method, we utilized two large-scale egocentric video benchmarks: EPIC-Kitchens55 [31] and EGTEA Gaze+ [32]. 1) *EPIC-Kitchens55*. Videos of EPIC-Kitchens55 dataset are recorded by 32 participants conducting their non-scripted daily activities in their kitchen environments. It consists of 39, 596 video segments annotated by 2, 513 action labels in total. As the testing set of EPIC-Kitchens55 is unavailable, similar to [13] and [15], we re-split the public training dataset of 28, 472 video segments, into two parts: 23, 493 video segments from 232 videos for training and 4, 979 video segments from the other 40 videos for testing. And 2) *EGTEA Gaze+*. As to the EGTEA Gaze+ [32] dataset, it consists of 10, 328 video segments annotated by 106 action labels. Specifically, EGTEA Gaze+ [32], [33] provides three different dataset splits, as shown in Table I. Each split was randomly sampled from the whole dataset, where 80% of the samples per action is used for training and the rest for testing. For both datasets, we uniformly sampled $K = 8$ video snippets from each action segment. To validate the model performance, we adopted the top-1 accuracy as the evaluation metrics for the following observation ratios (*i.e.,* k/K): 12.5%, 25%, 37.5%, 50%, 62.5%, 75%, 87.5%, and 100%.

2) Modality Features: We employed four different modalities in the EPIC-Kitchens55 dataset: visual content, flow content, object labels, and audio signals. For fair comparison, we adopted the same features with the work [15]. In particular, we utilized the features of visual content, flow content, and objected labels provided by [13], the dimensions of which are 1, 024, 1, 024, and 352, respectively. Meanwhile, we used the features of audio modality provided by [16], the dimension of which is 1, 024. The four modalities are arranged in the order of "*[visual modality, flow modality, object modality, audio modality]*" as the input of our model. As for the EGTEA Gaze+ dataset, it only provides three modalities of videos, including the visual content, the flow content, and the object labels. Therefore, for this dataset, the input of our model involves three modalities arranged in the order of "*[visual modality, flow modality, object modality]*" as the input of our model. We directly adopted the features released by [15]. The dimensions of the visual and flow content features are both 1, 024, while that of the object label feature is 352.

3) Implementation Details: For encoding the input sequence of each modality in both datasets, we utilized a Transformer encoder with 2 layers and 8 attention heads. The hidden layer dimension of the Transformer encoder in both datasets is $d_1 = 768$. The dimension of the multi-modal fused feature in EPIC-Kitchens55 is $d_2 = 1024$, while that in

TABLE II

PERFORMANCE COMPARISON AMONG DIFFERENT METHODS IN TERMS OF TOP-1 ACCURACY (%) OVER THE EPIC-KITCHENS55 DATASET WITH DIFFERENT OBSERVATION RATIOS. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

Observation ratio	12.5%	25%	37.5%	50%	62.5%	75%	87.5%	100%	avg.
LSTM-late	25.48	29.51	31.54	32.58	33.57	34.45	34.65	34.35	32.02
LSTM-early	24.96	28.18	30.01	31.52	32.92	33.11	33.45	33.17	30.91
RULSTM (Furnari et al. 2022) [13]	24.48	27.63	29.44	30.93	32.16	33.09	33.63	34.07	30.68
RULSTM-audio (Kazakos et al. 2022) [16]	25.46	28.92	31.09	32.90	34.61	35.38	35.78	36.12	32.53
AANet (Kong et al. 2022) [12]	21.38	25.14	28.92	31.09	32.30	33.63	34.11	34.21	30.10
PTSN (Wang et al. 2022) [10]	25.82	29.02	30.95	32.64	34.05	34.73	35.14	34.69	32.13
AKT (Cai et al. 2022) [11]	24.66	27.35	27.19	30.23	31.94	33.61	34.21	34.96	30.52
AKD (Zheng et al. 2022) [15]	27.65	31.26	33.07	34.98	35.90	36.99	37.45	37.55	34.36
mTRACE	**28.74**	**33.00**	**36.02**	**37.33**	**38.35**	**39.08**	**39.26**	**39.66**	**36.43**

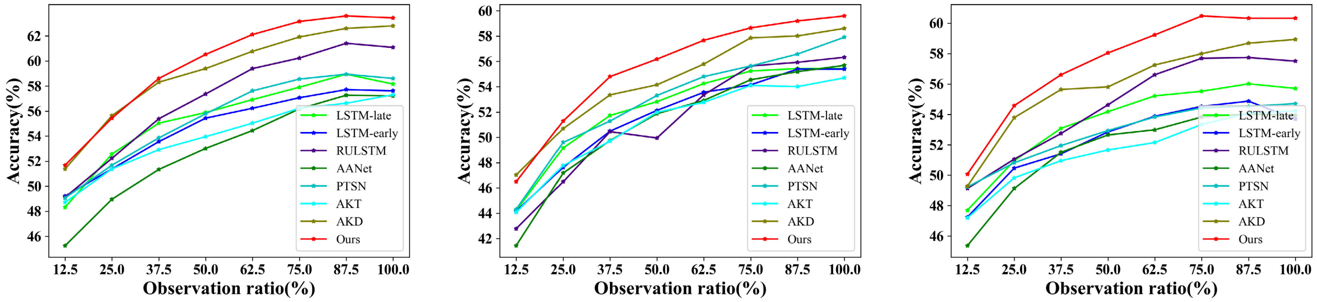
EGTEA Gaze+ is 512. For both datasets, in the pre-training stage, the parameters and to-be-learned tokens of transformer were randomly initialized. The number of pre-training epochs was set to 50, the dropout rate was set to 0.1, and the batch size was set to 128. We employed the Adam optimizer and made the learning rate first linearly increased from the initial 0.0001 to 0.001 by using the warmup strategy in the first 10 epochs and then gradually decayed to 1e-6 by the end. In the mutual enhancement stage, we trained 30 epochs in total. The weight of the consistency regularization is set to $w^{fea} = 5$. For EPIC-Kitchens55, we employed the Adam optimizer with an initial learning rate of 0.0001, which multiplies 0.5 at the 15-th and 20-th epochs, respectively. As for EGTEA Gaze+, the initial learning rate is set to 0.0001, which multiplies 0.5 at the 9-th and 12-th epochs, respectively. The batch size for EPIC-Kitchens55 and EGTEA Gaze+ is set to 256 and 128, respectively. The dropout rate of the model for EPIC-Kitchens55 and EGTEA Gaze+ is set to 0.3 and 0.15, respectively. In the end-to-end aggregation stage, for both datasets, we froze all the parameters of the Transformer encoder, and the initial integration weight of the fully-connected layer is set to 0. For EPIC-Kitchens55, we trained 80 epochs in total. The dropout rate in this stage is set to 0.95 and the batch size is set to 1024. For EGTEA Gaze+, the number of training epochs is set to 20, the dropout rate is set to 0.9, and the batch size is set to 1024. In this stage, we also employed Adam optimizer for both datasets. The learning rate for EPIC-Kitchens55 and EGTEA Gaze+ was gradually improved to 0.0004 and 0.0007, respectively, by using the warmup strategy in the first four epochs and maintaining till the end. Ultimately, for EPIC-Kitchens55, the trade-off hyperparameter for the multi-modal action prediction in Eqn. (20) is set to $\gamma = 0.48$, while that for the single-modal action predictions (*i.e.,* $\{\gamma_m\}$) are set to $\{0.02, 0.10, 0.18, 0.22\}$, respectively. Pertaining to EGTEA Gaze+, the trade-off hyperparameter $\gamma = 0.6$, while that for the single-modal action predictions are set to $\{0.12, 0.27, 0.01\}$, respectively.

B. On Model Comparison (RQ1)

To validate the effectiveness of our proposed methods, we chose the following baselines for comparison.

- **LSTM-late.** This baseline utilizes a one-layer LSTM to encode each modality and then fuses all the prediction results as the final output.
- **LSTM-early.** Similar to LSTM-late, this method adopts the early fusion strategy to concatenate the feature vectors of all modalities.
- **RULSTM.** This method utilizes a rolling-unrolling LSTM architecture to solve the egocentric action anticipation task, which jointly explores three modalities (*i.e.*, flow contents, visual contents and object labels) of the video segment.
- **RULSTM-audio.** As for a fair comparison, we design this baseline by extending RULSTM to incorporate the audio signal modality.
- **PTSN.** This baseline, devised for early action prediction from the third-person view, uses a teacher-student learning block for distilling progressive knowledge from teacher to student, where the teacher is an action recognition model and the student is the to-be-learned early action prediction model.
- **AANet.** This is an adversarial action prediction framework, which aims to learn the representative and discriminative features to enhance the performance of early action prediction from the third-person view.
- **AKT.** This baseline employs the knowledge transfer learning from fully observed videos to boost the third-person view early action prediction performance.
- **AKD.** This is a multi-modal adversarial knowledge distillation framework for egocentric early action prediction, which involves a teacher network to learn a comprehensive video representation based on the multi-modal full video segment, and a student network to predict the action only based on the partial video segment.

Table II and Figure 5 show the performance comparison among different methods in terms of Top-1 accuracy (%) over the EPIC-Kitchens55 dataset and EGTEA Gaze++ dataset with different observation ratios, respectively. As can be seen, our mTRACE model almost consistently outperform all the baseline methods on all observation ratios, which demonstrates the effectiveness of the proposed Transformer-based dual action prediction model for the egocentric early action prediction. Specifically, on average, our mTRACE model exceeds the best baseline (*i.e.,* AKD) by 2.07% regarding the Top-1 accuracy on the EPIC-Kitchens55 dataset. This may be due to three reasons. 1) All the baseline methods adopted LSTM as the backbone for encoding the video segment, while our mTRACE model utilizes the Transformer encoder, which can model the non-sequential correlations among the discrete snip-



(a) EGTEA Gaze+ S1 **(b) EGTEA Gaze+ S2** **(c) EGTEA Gaze+ S3**

Fig. 5. Performance comparison among different methods with different observation ratios on the (a) EGTEA Gaze+ S1, (b) EGTEA Gaze+ S2 and (c) EGTEA Gaze+ S3, respectively.

TABLE III

ABLATION STUDY RESULTS OVER THE EPIC-KITCHENS55 DATASET

Observation ratio	12.5%	25%	37.5%	50%	62.5%	75%	87.5%	100%	Avg.
w/o Pre-training	28.64	32.46	35.04	36.56	37.41	38.48	38.74	39.20	35.82
w/o Single-Modal Classification	27.84	31.86	34.43	35.96	36.89	37.83	38.19	37.17	35.02
w/o Consistency Regularizer	28.76	32.32	34.69	36.20	36.93	37.47	37.85	38.21	35.31
w/o Mutual Enhancement Stage	27.80	31.62	33.71	35.94	36.77	37.01	37.43	37.57	34.73
w/o End-to-end Aggregation Stage	**28.78**	32.68	35.66	37.13	37.95	38.46	38.98	39.02	36.08
mTRACE	28.74	**33.00**	**36.02**	**37.33**	**38.35**	**39.08**	**39.26**	**39.66**	**36.43**

pets in the video segment. 2) Different from all the baselines, our mTRACE model incorporates the correlation modeling between the observed video segment and unobserved video segment with two separate multi-modal features. 3) Beyond all the baseline methods, we optimize the model by two stages, where the two action prediction modules (*i.e.,* early segment action prediction module and future segment action prediction module) can mutually share knowledge to each other, and fully aggregated in the testing phase.

In addition, we also compared the training time of our mTRACE model and the best baseline AKD. For EPIC-Kitchens55, the training time cost of the pre-training stage, mutual enhancement stage, and end-to-end aggregation stage of our model with an A100 GPU are 50 minutes, 120 minutes, and 160 minutes, respectively, while AKD takes 160 minutes and 100 minutes for the pre-training and training under the same conditions, respectively. For EGTEA Gaze++, the training time cost of the pre-training stage, mutual enhancement stage, and end-to-end aggregation stage of our model are 20 minutes, 40 minutes, and 60 minutes, respectively, while AKD takes 90 minutes and 40 minutes for the pre-training and training, respectively. Overall, the training time cost of our model and the best baseline is comparable.

C. On Ablation Study (RQ2)

We conducted the ablation study on our model with the following model derivatives. 1) **w/o Pre-training**: To verify the effectiveness of autoregressive pre-training, we randomly initialized all the Transformer parameters and then adopted the two-stage optimization training. 2) **w/o Single-Modal Classification**: To explore the function of the single-modal classifications, we removed the losses of single-modal classifications in both training stages, and eliminated the single-modal action prediction results in the testing phase. 3) **w/o Consistency Regularizer**: To justify the effect of $\tilde{\mathbf{h}}_k^u$, we removed the feature consistency regularization from the mutual learning enhancement stage. 4) **w/o Mutual Enhancement Stage**: To study the impact of the mutual enhancement stage, we removed the mutual enhancement stage and directly carried out the end-to-end aggregation training stage. In this derivative, to improve the modal capacity, the parameters of the Transformer were not frozen in the training stage. And 5) **w/o End-to-end Aggregation Stage**: Similarly, to investigate the effect of end-to-end aggregation stage, we directly removed the end-to-end aggregation training stage.

Table III summarizes the ablation study results over the EPIC-Kitchens55 dataset. It can be seen that our methods consistently outperforms all the derivatives, which demonstrates the effectiveness of each component in our proposed model. Specifically, we had the following four detailed observations. 1) The performance of w/o Pre-training and w/o Single-Modal Classification drop largely, which indicates that the autoregressive pre-training does improve the feature encoding capability of the Transformer encoder in the dual action prediction modules, and the single-modal classification is indeed effective in enhancing the discriminative feature learning of our model. 2) w/o Consistency Regularizer performs much worse than our model, which validates that the consistency regularizer is useful in allowing the two action prediction modules to share knowledge with each other and hence gaining better action prediction performance. 3) Both w/o Mutual Enhancement Stage and w/o End-to-end Aggregation Stage perform inferior to our method, which indicates that it is essential to design the two-stage optimization paradigm.

D. On Sensitivity Analysis (RQ3)

Besides, we studied the sensitivity of our model pertaining to the number of heads and the number of layers of the Transformer encoder, as well as the trade-off hyperparameter

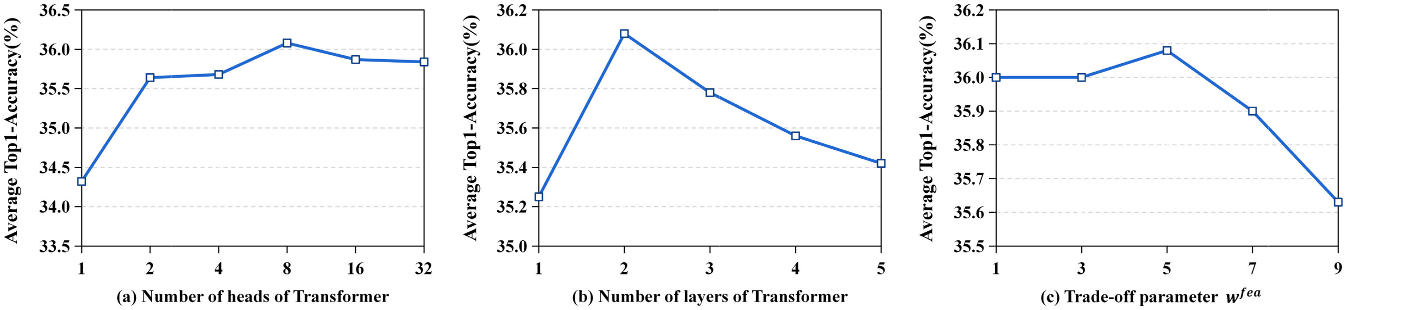


Fig. 6. Performance of our model with different numbers of heads and layers of the Transformer encoder, as well as different trade-off weight values of w^{fea} for the consistency regularizer.

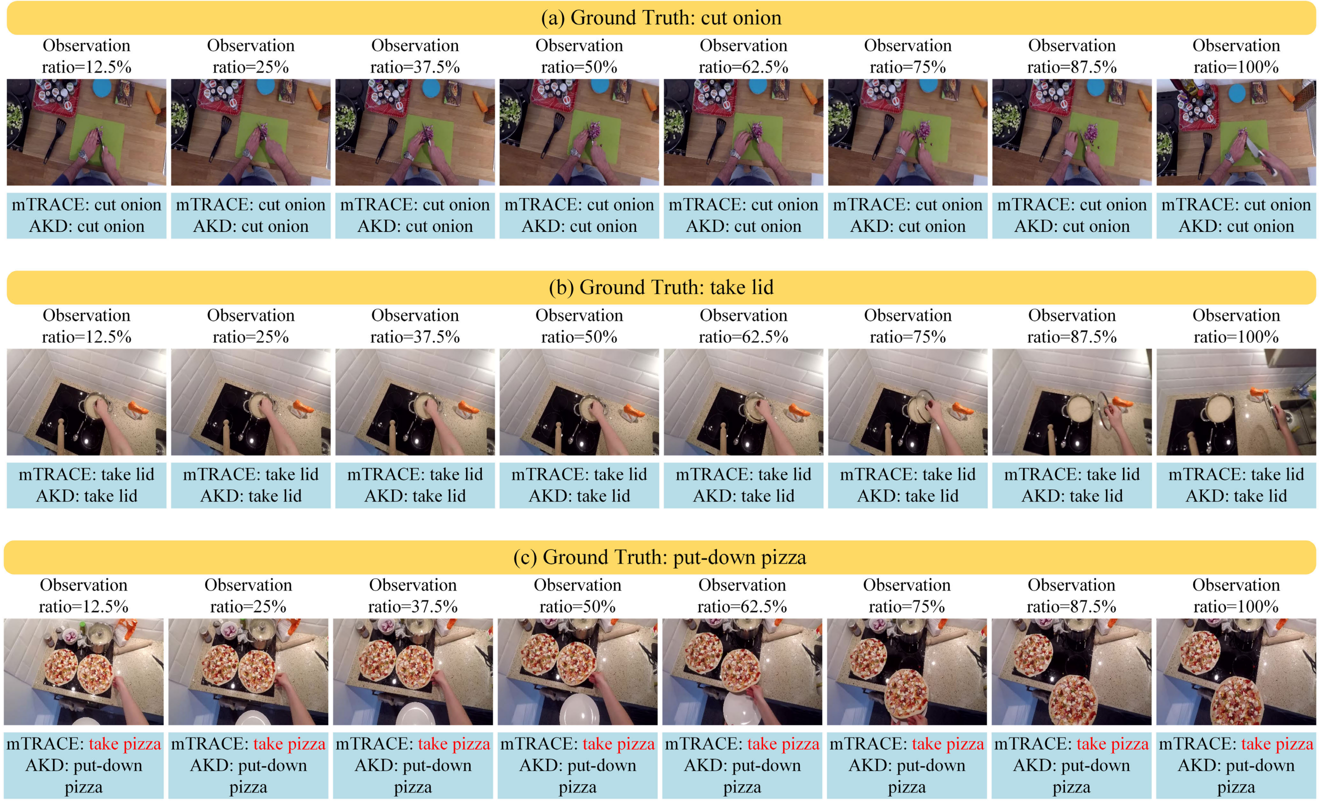


Fig. 7. Action predictions of our model and the best baseline AKD for three testing examples in the EPIC-Kitchens55 dataset, where all observation ratios have been tested for comprehensive comparison. For a clear illustration, we present the frames of the corresponding observation ratios for all the examples. Incorrect predictions have been highlighted in red.

for the consistency regularizer in Eqn. (14) and Eqn. (15) with the EPIC-Kitchens55 dataset.

1) On the Number of Heads of Transformer Encoder: We varied the number of heads in the transformer model in the range of {1, 2, 4, 8, 16, 32}. Figure 6(a) shows the performance of our model with different number of heads in terms of the average top-1 accuracy. As can be seen, with the increase of the number of heads used in the Transformer encoder, the performance of our model increases first and then decreases. In the end, our model achieves the optimal performance when the number of heads is 8. When the number of heads is larger than 8, the model's performance slightly drops. This may be due to too many heads may lead to the overfitting issue.

2) On the Number of Layers of Transformer Encoder: To learn the impact of the number of layers used in the Transformer encoder, we changed the number of layers from 1 to 5 with the step of 1. As can be seen from Figure 6(b), our model performs best when the number of layers in Transformer encoder is 2. One likely reason is also that too many layers can lead to the overfitting problem.

3) On the Trade-off Hyperparameter for Consistency Regularizer w^{fea}: To explore the effect of the trade-off hyperparameter in controlling the balance between the cross-entropy losses and the consistency regularizer in the optimization of the mutual enhancement stage, we tuned the trade-off hyperparameter around the optimal value 5. In particular, we varied it from 1 to 9 with the step of 2. As can be seen from

Figure 6(c), the performance of our model first goes better with the increase of the hyperparameter, and then goes worse with the continue increase. This is reasonble as the too small w^{fea} cannot guarantee the knowledge transferring between the too modules, while too large w^{fea} may downgrade the discriminative feature learning towards action prediction.

E. On Case Study (RQ4)

To gain the intuitive understanding of our mTRACE model for the task of egocentric early action prediction, we reported the action predictions of our model and the best baseline AKD on three testing cases in the EPIC-Kitchens55 dataset, where all observation ratios have been tested for comprehensive comparison. As we can see from Fig 7(a), the person is cutting an onion in the video segment. As "cutting onion" is an action that is easy to recognize, both our model and AKD given the correct predictions at all observation ratios. This indicates that both our model and AKD can predict correctly in the relatively simple and obvious case. For the more complex and difficult case in Fig 7(b), where the to-be-recognized action "take lid" is highly similar to "put lid", our model can correctly predict the on-going action "take lid" by observing only 12.5% snippets, while the baseline method AKD has to observe 37.5% snippets. This confirms the effectiveness of our Transformer-based dual action prediction model in modeling the correlations between the observed and unobserved video segments. Fig 7(c) shows a failure case of our model. As can be seen, compared with AKD, our model fails to predict the action labels at all observation ratios. By analyzing the video content, we noticed that for this case, our predicted ongoing action "take pizza" is correct, while the ground truth label "put-down pizza" should be accidentally annotated and is a noisy label. This also reflects the capability of our mTRACE model in solving the task of egocentric early action prediction.

V. Conclusion and Future Work

In this paper, we present a novel multimodal TransfoRmer-based duAl aCtion prEdiction (mTRACE) model for the task of egocentric early action prediction, which involves two key modules: the early (observed) segment action prediction module and the future (unobserved) segment action prediction module. Both modules adopt the Transformer encoders to encode the video segment. To model the coherence of the full video segment, we introduce a consistency regularizer between the two modules. Moreover, to boost the performance, we design a two-stage optimization scheme, including the mutual enhancement stage and end-to-end aggregation stage. Extensive experiments on two public datasets have demonstrated the superiority of our proposed model, and the benefit of incorporating Transformer encoder, consistency regularizer, pre-training as well as the two-stage training. Currently, in this work, we directly adopted existing autoregresssive pre-training technique to pre-train our model. In the future, we plan to investigate more advanced pre-training scheme to further promote the model performance.

References

[1] Z. Yang, Y. Li, J. Yang, and J. Luo, "Action recognition with spatio-temporal visual attention on skeleton image sequences," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 8, pp. 2405–2415, Aug. 2019.

[2] J. Weng, X. Jiang, W.-L. Zheng, and J. Yuan, "Early action recognition with category exclusion using policy-based reinforcement learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 12, pp. 4626–4638, Dec. 2020.

[3] S. Li, K. Li, and Y. Fu, "Early recognition of 3D human actions," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 14, no. 1, p. 20, 2018.

[4] Y. Tang, Z. Wang, J. Lu, J. Feng, and J. Zhou, "Multi-stream deep neural networks for RGB-D egocentric action recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 10, pp. 3001–3015, Oct. 2019.

[5] M. Wang, C. Luo, B. Ni, J. Yuan, J. Wang, and S. Yan, "First-person daily activity recognition with manipulated object proposals and non-linear feature fusion," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2946–2955, Oct. 2017.

[6] X. Wang, L. Zhu, H. Wang, and Y. Yang, "Interactive prototype learning for egocentric action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8148–8157.

[7] T. Liu, K.-M. Lam, R. Zhao, and J. Kong, "Enhanced attention tracking with multi-branch network for egocentric activity recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 3587–3602, Jun. 2022.

[8] W. M. Alvarez, F. M. Moreno, O. Sipele, N. Smirnov, and C. Olaverri-Monreal, "Autonomous driving: Framework for pedestrian intention estimation in a real world scenario," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Oct. 2020, pp. 39–44.

[9] A. Rasouli, I. Kotseruba, and J. K. Tsotsos, "Pedestrian action anticipation using contextual feature fusion in stacked rnns," in *Proc. 30th Brit. Mach. Vis. Conf. (BMVC)*. London, U.K.: BMVA Press, 2019, p. 171.

[10] X. Wang, J.-F. Hu, J.-H. Lai, J. Zhang, and W.-S. Zheng, "Progressive teacher-student learning for early action prediction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3556–3565.

[11] Y. Cai, H. Li, J. Hu, and W. Zheng, "Action knowledge transfer for action prediction with partial videos," in *Proc. AAAI Conf. Artif. Intell. (AAAI)*. Palo Alto, CA, USA: AAAI Press, 2019, pp. 8118–8125.

[12] Y. Kong, Z. Tao, and Y. Fu, "Adversarial action prediction networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 3, pp. 539–553, Mar. 2020.

[13] A. Furnari and G. Farinella, "What would you expect? Anticipating egocentric actions with rolling-unrolling LSTMs and modality attention," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6251–6260.

[14] A. Furnari and G. M. Farinella, "Rolling-unrolling LSTMs for action anticipation from first-person video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 11, pp. 4021–4036, Nov. 2021.

[15] N. Zheng, X. Song, T. Su, W. Liu, Y. Yan, and L. Nie, "Egocentric early action prediction via adversarial knowledge distillation," *ACM Trans. Multim. Comput. Commun. Appl.*, vol. 1, no. 1, pp. 1–16, 2022.

[16] E. Kazakos, A. Nagrani, A. Zisserman, and D. Damen, "EPIC-fusion: Audio-visual temporal binding for egocentric action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5491–5500.

[17] H. Fan et al., "Multiscale vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6804–6815.

[18] R. Girdhar and K. Grauman, "Anticipative video transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13485–13495.

[19] M. Huang et al., "Multifeature selection for 3D human action recognition," *ACM Trans. Multim. Comput. Commun. Appl.*, vol. 14, no. 2, p. 45, 2018.

[20] Y. Kong, Z. Tao, and Y. Fu, "Deep sequential context networks for action prediction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3662–3670.

[21] A. Vaswani et al., "Attention is all you need," in *Proc. Neural Inf. Process. Syst. (NeurIPS)*. Cambridge, MA, USA: MIT Press, 2017, pp. 5998–6008.

[22] A. Dosovitskiy et al., "An image is worth 16 × 16 words: Transformers for image recognition at scale," in *Proc. 9th Int. Conf. Learn. Represent. (ICLR)*, 2021, pp. 1–21.

[23] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. Jégou, "Training data-efficient image transformers & distillation through attention," in *Proc. 34th Int. Conf. Mach. Learn. (ICML)*, 2021, pp. 10347–10357.

[24] J. Yu, J. Li, Z. Yu, and Q. Huang, "Multimodal transformer with multi-view visual representation for image captioning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 12, pp. 4467–4480, Dec. 2020.

[25] S. Cao, G. An, Z. Zheng, and Z. Wang, "Vision-enhanced and consensus-aware transformer for image captioning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 10, pp. 7005–7018, Oct. 2022.

[26] Z. Liu, Y. Tan, Q. He, and Y. Xiao, "SwinNet: Swin transformer drives edge-aware RGB-D and RGB-T salient object detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 7, pp. 4486–4497, Jul. 2022.

[27] L. Zhao, J. Guo, D. Xu, and L. Sheng, "Transformer3D-Det: Improving 3D object detection by vote refinement," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 12, pp. 4735–4746, Dec. 2021.

[28] D. Roy and B. Fernando, "Action anticipation using pairwise human-object interactions and transformers," *IEEE Trans. Image Process.*, vol. 30, pp. 8116–8129, 2021.

[29] Z. Yuan, X. Song, L. Bai, Z. Wang, and W. Ouyang, "Temporal-channel transformer for 3D LiDAR-based video object detection for autonomous driving," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2068–2078, Apr. 2022.

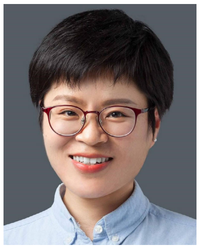
[30] C. Vondrick, H. Pirsiavash, and A. Torralba, "Anticipating visual representations from unlabeled video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 98–106.

[31] D. Damen et al., "Scaling egocentric vision: The dataset," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2018, pp. 753–771.

[32] Y. Li, M. Liu, and J. M. Rehg, "In the eye of beholder: Joint learning of gaze and actions in first person video," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Cham, Switzerland: Springer, 2018, pp. 639–655.

[33] Y. Li, M. Liu, and M. J. Rehg, "In the eye of beholder: Gaze and actions in first person video," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jan. 15, 2021, doi: 10.1109/TPAMI.2021.3051319.

Weili Guan (Member, IEEE) received the master's degree from the National University of Singapore. She is currently pursuing the Ph.D. degree with the Faculty of Information Technology, Monash University (Clayton Campus), Australia. She is also an Intern with the Peng Cheng Laboratory. After her master's degree, she joined Hewlett Packard Enterprise, Singapore, as a Software Engineer, where she has worked for around five years. She has published many papers at the first-tier conferences and journals, such as ACM MM, SIGIR, and IEEE TRANSACTIONS ON IMAGE PROCESSING. Her research interests are multimedia computing and information retrieval.



Xuemeng Song (Senior Member, IEEE) received the B.E. degree from the University of Science and Technology of China, in 2012, and the Ph.D. degree from the School of Computing, National University of Singapore, in 2016. She is currently an Associate Professor with Shandong University, China. She has published several papers in the top venues, such as ACM SIGIR, ACM MM, and *ACM TOIS*. Her research interests include information retrieval and social network analysis. She has served as a reviewer for many top conferences and journals. She is also an AE of *IET Image Processing*.

Kejie Wang is currently pursuing the B.Eng. degree in computer science with Shandong University. His research interests include multimedia computing.

Haokun Wen received the B.E. degree from the Ocean University of China, in 2019, and the master's degree from the School of Computer Science and Technology, Shandong University, in 2022. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen). He has published several papers in top venues, such as ACM SIGIR, ACM MM, IEEE TRANSACTIONS ON IMAGE PROCESSING, and *ACM TOMM*. His research interests include multimedia computing and information retrieval.

Hongda Ni is currently pursuing the B.Eng. degree in computer science with the Harbin Institute of Technology (Shenzhen). His research interests include multimedia computing and natural language processing.

Yaowei Wang (Member, IEEE) received the Ph.D. degree in computer science from the University of Chinese Academy of Sciences in 2005. He has worked with the Department of Electronics Engineering, Beijing Institute of Technology, from 2005 to 2019. Currently, he is a Professor with the Peng Cheng Laboratory, Shenzhen, China. He has coauthored more than 120 technical articles in international journals and conferences, including IEEE TRANSACTIONS ON IMAGE PROCESSING, CVPR, and ICCV. His research interests include machine learning, and multimedia content analysis and understanding. He serves as a member for CIE, CCF, and CSIG. He was a recipient of the second prize of the National Technology Invention in 2017 and the first prize of the CIE Technology Invention in 2015. He serves as the Chair for the IEEE Digital Retina Systems Working Group. He has promoted the digital retina technology, made efforts to establish system standards for Digital Retina. He has trained a vision model named "Pengcheng Dasheng" with one billion parameters, achieving an over 10% performance gain in the detection and recognition task in more than 20 application scenarios. He led the development of the first digital retina verification systems, which has been applied to the urban traffic management field of over 30 large and medium-sized cities in China.

Xiaojun Chang (Senior Member, IEEE) is a Professor with the Faculty of Engineering and Information Technology, Australian Artificial Intelligence Institute, University of Technology Sydney (UTS). He is also the Director of the ReLER Laboratory. He is also an Honorary Professor with the School of Computing Technologies, RMIT University, Australia, where he was an Associate Professor with the School of Computing Technologies, before joining UTS. After graduation, he subsequently worked as a Post-Doctoral Research Fellow with the School of Computer Science, Carnegie Mellon University and a Lecturer and a Senior Lecturer with the Faculty of Information Technology, Monash University, Australia. He has focused his research on exploring multiple signals (visual, acoustic, and textual) for automatic content analysis in unconstrained or surveillance videos. His team has won multiple prizes from international grand challenges, which hosted competitive teams from MIT, University of Maryland, Facebook AI Research (FAIR), and Baidu VIS; and aim to advance visual understanding using deep learning. For example, he won the first place in the TrecVID 2019–Activity Extended Video (ActEV) Challenge, which was held by the National Institute of Standards and Technology, USA.