Jointly-Learnt Networks for Future Action Anticipation via Self-Knowledge Distillation and Cycle Consistency

Md Moniruzzaman, Student Member, IEEE, Zhaozheng Yin, Senior Member, IEEE, Zhihai He, Fellow, IEEE, Ming C Leu, Member, IEEE, and Ruwen Qin, Member, IEEE

Abstract—Future action anticipation aims to infer future actions from the observation of a small set of past video frames. In this paper, we propose a novel Jointly-learnt Action Anticipation Network (J-AAN) via Self-Knowledge Distillation (Self-KD) and cycle consistency for future action anticipation. In contrast to the current state-of-the-art methods which anticipate the future actions either directly or recursively, our proposed J-AAN anticipates the future actions jointly in both direct and recursive ways. However, when dealing with future action anticipation, one important challenge to address is the future's uncertainty since multiple action sequences may come from or be followed by the same action. Training an action anticipation model with one-hot-encoded hard labels that assign zero probabilities to incorrect yet semantically similar actions may not handle the uncertain future. To address this challenge, we design a Self-KD mechanism to train our J-AAN, where the J-AAN gradually distills its own knowledge during the training to soften the hard labels to model the uncertainty on future action anticipation. Furthermore, we design a forward and backward action anticipation framework with our proposed J-AAN based on a cyclic consistency constraint. The forward J-AAN anticipates the future actions from the observed past actions, and the backward J-AAN verifies the anticipation of the forward J-AAN by anticipating the past actions from the anticipated future actions. The proposed method outperforms all the latest state-of-the-art action anticipation methods on the Breakfast, 50Salads, and EPIC-Kitchens-55 datasets. This project will be publicly available on https://github.com/MoniruzzamanMd/J-AAN.

Index Terms—Future Action Anticipation, Self-Knowledge Distillation, Cycle Consistency.

I. Introduction

VIDEO analysis algorithms have been achieving tremendous progress on automatic video-based object action understanding in the past few years, including action recognition [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], temporal action localization [13], [14], [15], [16], [17], [18], [19], spatio-temporal action detection [20], [21], [22]. Most of these tasks analyze the entire video, such as the action recognition task that infers an action category from a video containing the complete action.

Early action recognition [23], [24], [25], [26], [27], [28] has been investigated to infer the label of an action from the small early portion of the video of that action. In contrast to early action recognition, action anticipation, which predicts what will happen in the future based on a short period of video observation, infers the future actions that may not appear in the early portion of the video. Similar to the literature, we use the term of action anticipation in this paper to indicate that a future action is anticipated to occur based on the past observation, without using the term of action prediction which can be defined as the inference of the action category of a video. Action anticipation is an important and challenging problem in computer vision, owing to its application in many areas such as autonomous driving and human-robot collaboration.

Recently, some works [29], [30], [31], [32], [33] investigated the anticipation of the next action immediately following the current one, but such short-term near-future anticipation is not sufficient for many real-world applications. For instance, anticipating a future traffic accident from dashboard cameras ahead of time is very valuable for autonomous driving, and by anticipating a manufacturing worker's future actions in advance, collaborative robots can have sufficient time to grasp materials and move to the correct location to assist humans.

In this work, we focus on the challenging problem of anticipating future actions and their corresponding duration in a longer time horizon from the observation of the past few frames, which may include multiple sequential actions, as illustrated in Fig. 1. However, anticipating the future actions and their duration in a longer time horizon is challenging due to the uncertainty of the future, and the weaker cause-effect correlation between the observed past and the future actions that are far-away from the observed actions. The above challenges lead to a research question: how to obtain an action anticipation algorithm that can model the uncertain future and accurately anticipate the actions and their corresponding duration in both the near and far future?

Key observations: To address the above research question, we have the following three key observations:

Observation 1: The future actions are jointly anticipable in both direct and recursive ways. The direct anticipation directly anticipates all the future actions and their corresponding duration in a single step, while the recursive anticipation anticipates

M. Moniruzzaman is with the Department of Computer Science, Stony Brook University, New York, NY 11794 (e-mail: mmoniruzzama@cs.stonybrook.edu)

Z. Yin is with the Department of Computer Science and Department of Biomedical Informatics, Stony Brook University, New York, NY 11794 (e-mail: zyin@cs.stonybrook.edu)

Z. He is with the Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO 65211 (e-mail: hezhi@missouri.edu)

M. C. Leu is with the Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology, Rolla, MO 65401 (e-mail: mleu@mst.edu)

R. Qin is with the Department of Civil Engineering, Stony Brook University, New York, NY 11794 (e-mail: ruwen.qin@stonybrook.edu)

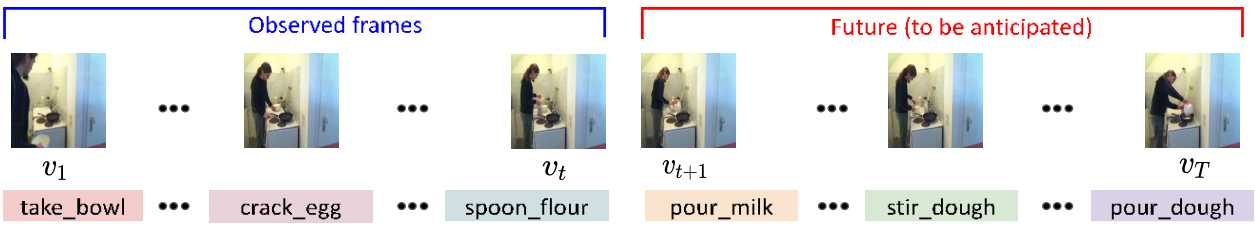


Fig. 1. An example video of future action anticipation in the longer time horizon, where the future may contain multiple sequential actions.

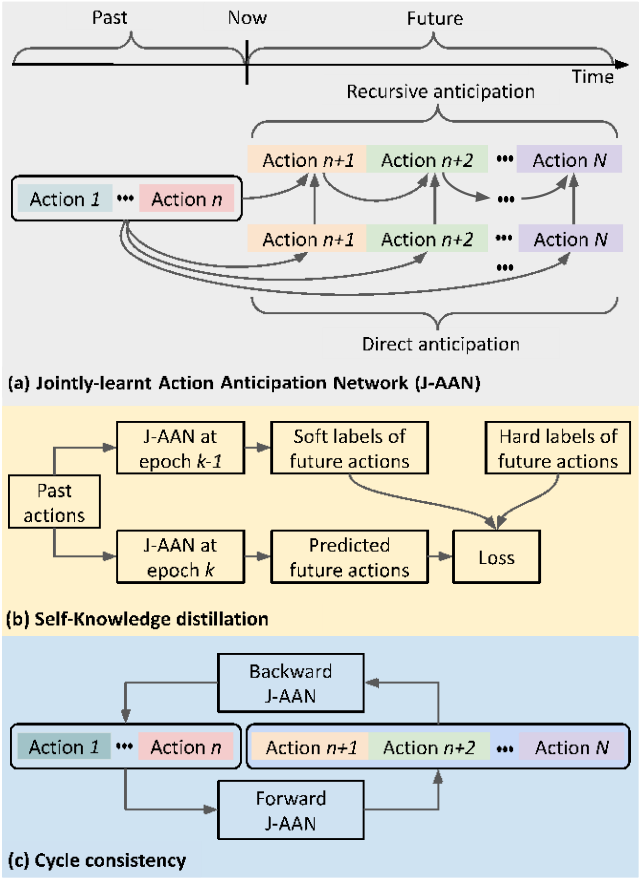


Fig. 2. Illustration of our three main ideas: (a) Jointly-learnt Action Anticipation Network (J-AAN) that anticipates the future actions from the observed past actions in both direct and recursive ways; (b) Self-knowledge distillation mechanism to train the J-AAN, where the J-AAN gradually distills its own knowledge during training; and (c) Forward and backward J-AANs with cycle consistency, where the backward J-AAN evaluates how well the forward J-AAN anticipates the future actions by anticipating the past actions from the anticipated future actions.

the future actions and their corresponding duration one-by-one in a recursive way. In joint learning, the recursive anticipation can take the advantage of direct anticipation (since the direct anticipation anticipates all the future actions in a single step) for better future action anticipation.

Observation 2: In action anticipation, multiple action sequences may follow the same action. For example, both actions "pour coffee" and "spoon powder" may happen after a person takes a cup from the cupboard. In such cases, training an action anticipation model with hard labels (i.e., the one-hot encoded labels of the ground-truth future actions that assign zero probabilities to incorrect yet semantically similar actions) may not handle the future's uncertainty. One possible solution to deal with this challenge is to train the action anticipation model with soft labels (i.e., the labels that reduce the most confident value of the one-hot-vector and assign a small amount of probability mass to semantically similar actions).

Observation 3: Actions are not only forward anticipable but also backward anticipable, considering the relation among sequential actions.

Our proposal: The above observations motivate us to design the following approaches to address the research question:

(1) The observation 1 leads us to design a novel Jointly-learnt Action Anticipation Network (J-AAN) that anticipates the future actions and their corresponding duration in both direct and recursive ways (Fig. 2(a)) from the observed past actions. Since the direct anticipation directly anticipates all the actions in a single step, the recursive anticipation utilizes them as an additional information in its one-by-one recursive anticipation approach to improve the anticipation performance.

(2) The observation 2 motivates us to design a Self-Knowledge Distillation (Self-KD) mechanism in the J-AAN, where the J-AAN progressively distills its own knowledge during the training to soften the hard labels of the ground-truth future actions and handle the uncertainty on future action anticipation. As shown in Fig. 2(b), J-AAN at epoch k is trained with target labels that are generated from the combination of the hard labels of the ground-truth future actions and the anticipated future actions (soft labels) of the previous epoch. In this Self-KD, the J-AANs at epoch $k-1$ and k are teacher and student anticipation models, respectively. The student model at epoch k will be the teacher model for epoch $k+1$, which gradually utilizes its own knowledge for softening the hard ground-truth labels to enhance the generalization performance in the uncertain future.

(3) The observation 3 motivates us to design a forward-backward mechanism based on a cyclic consistency constraint to train the J-ANN, as shown in Fig. 2(c). Both the forward and backward networks are constructed with the same J-AAN architecture, satisfying a cyclic consistency constraint, in the sense that if the forward J-AAN can accurately anticipate the future from the past, then the backward J-AAN should be able to translate it back from the future to the past.

Our contributions are four-folds:

- We propose a novel Jointly-learnt Action Anticipation Network (J-AAN) that anticipates future actions jointly in both direct and recursive ways, where the recursive

anticipation takes the advantage of direct anticipation for better future action anticipation. To the best of our knowledge, this is the first work that integrates both the direct and recursive anticipation in a unified network.

- We design a Self-Knowledge Distillation (Self-KD) mechanism to train the J-AAN, where the J-AAN gradually distills its own anticipation to soften the hard labels during the training to handle the uncertainty on future action anticipation. To the best of our knowledge, this is the first work that designs Self-KD mechanism in future action anticipation.
- We design a forward and backward mechanism to train the J-AAN, where the backward J-AAN verifies the anticipation of the forward J-AAN based on a cyclic consistency constraint to further improve the performance.
- Our proposed approach outperforms all the latest action anticipation methods on the Breakfast [34], 50Salads [35], and EPIC-Kitchens-55 [36] datasets.

II. Related works

(1) Early action recognition: Human action recognition has been widely studied with significant progress [37], [38], [39], [40], [41], [42], [43], [44], [45]. Last few years, a large body of works [23], [24], [25], [26], [46], [47], [48], [49], [50], [51] focused on early action recognition, which aims to recognize the label of an action from the small early portion of the video of that action. Differently, in this paper, we focus on anticipating the future actions based on the observation of the small early portion of a video, where the future may contain multiple actions that may not appear in observed frames.

(2) Action anticipation. Most of the previous action anticipation algorithms [31], [52], [53], [54], [55], [56] anticipate the near-future action, and are limited to a few seconds in the future. Qi et. al [52] proposed a self-regulated learning framework for egocentric video activity anticipation. Liu et. al [53] introduced a memory augmented recurrent network to anticipate egocentric near future actions. Recently, several works [57], [58], [59] developed neural networks to anticipate the future actions for longer time horizon. Farha et al. [57] introduced an RNN model and a CNN model for future action anticipation. The RNN model conducts the anticipation in an iterative way, while the CNN model outputs a sequence of future actions in a form of a matrix. Ke et. al [58] developed a model to anticipate all the actions directly using temporal convolutions with a time-variable. Sener et al. [59] introduced a multi-granular temporal aggregation framework to anticipate the future actions. More recently, Gong et. al [60] introduced an end-to-end attention model to anticipate all future actions in parallel using fine-grained visual features of past frames. Most of these works anticipate the future actions either recursively or directly. In contrast, we propose a novel Jointly-learnt Action Anticipation Network (J-AAN) to anticipate the future actions and their duration jointly in both direct and recursive ways. Although our J-AAN may seem to be the combination of the direct and recursive anticipation methods, we do not simply train the direct anticipation and recursive anticipation methods, and then fuse their anticipation results. Differently, we jointly train the direct and recursive anticipation modules on top of a shared encoder, hence the encoded feature learning is influenced by both of these modules, eventually encoding the highly discriminative features from the observed past actions. Furthermore, in the joint learning process, the recursive anticipation takes the anticipated actions from the direct anticipation as one of the inputs for better future action anticipation. To the best of our knowledge, this is the first work that integrates both the direct and recursive anticipation in a unified network, and jointly train them to encode the highly discriminative features from the observed past actions for the better future action anticipation.

(3) Knowledge distillation: Distillation was originally proposed to transfer knowledge from a complex network (Teacher network) to a simple network (Student network) to improve the performance of the simple network (Student network) [61], [62], [63]. Recently, some works [23], [26] used the idea of knowledge distillation for early action recognition, where the Teacher network is first trained to recognize actions from full videos, and then the Student network distills the knowledge from the Teacher network and recognizes action from partial videos. More recently, the Self-Knowledge Distillation (Self-KD) becomes popular, in which the Student becomes the Teacher itself, and gradually distills its own knowledge during the training to soften the hard labels. Hence, the targets are adjusted adaptively during each epoch of training by combining the ground-truth (hard labels) and past predictions (soft label) from the model itself. The idea of Self-KD was previously utilized in natural language processing [64] and image classification [65], [66]. Differently, we design a new Self-KD mechanism in the future action anticipation, where our action anticipation model gradually distills its own anticipation during the training to soften the hard labels of the ground-truth future actions to enhance the generalization performance in the uncertain future. To the best of our knowledge, this is the first work that designs Self-KD mechanism in future action anticipation to model the future's uncertainty.

(4) Cycle consistency: Cycle consistency is a concept from machine translation where a phrase translated from English to French should translate it back from French to English. Previously, the cycle consistency was utilized in many computer vision tasks such as image-to-image translation [67], natural language processing [68], [69], visual tracking [70], trajectory prediction [71], depth estimation [72], and dense semantic alignment [73]. Recently, Farha et al. [74] utilized the cycle consistency concept in the future action anticipation to evaluate how well the anticipation network anticipates the future actions by anticipating the past actions given the anticipated future actions. However, this method evaluated the future actions by anticipating the past actions only in a recursive way from the anticipated future actions. Differently, we utilize the cycle consistency in the future action anticipation by designing a forward and backward framework for training the J-AAN, where the backward J-AAN anticipates the past actions in both direct and recursive ways from the anticipated future actions. Meanwhile, the anticipation of the forward J-AAN is evaluated by both of the direct and recursive anticipation modules, eventually leading to better anticipation.

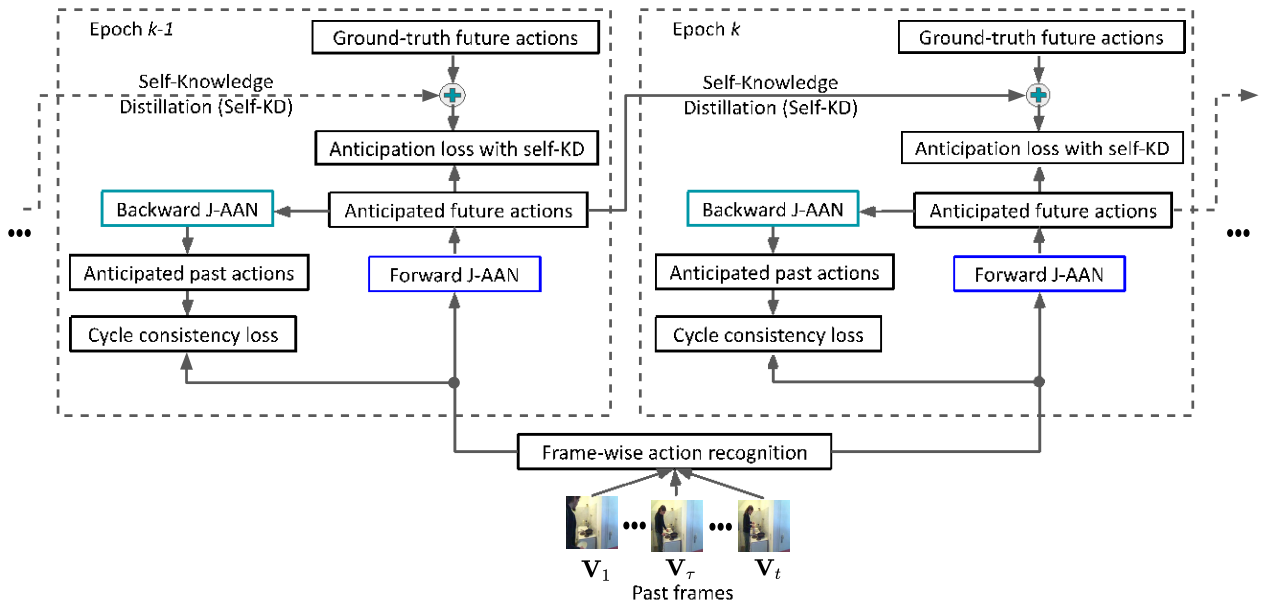


Fig. 3. Overview of our proposed Jointly-learnt Action Anticipation Network (J-AAN) for future action anticipation via Self-Knowledge Distillation (Self-KD) and cycle consistency. During the training, it consists of two networks, a forward network, and a backward network. Both the forward and backward networks are constructed with our proposed J-AAN. The forward J-AAN anticipates the future actions from the observed frame-wise past actions, and gradually distills its own knowledge to soften the hard labels during the training to handle the uncertainty of future actions. The backward J-AAN takes the anticipated future actions from the forward J-AAN as an input to anticipate the past actions, satisfying a cyclic consistency constraint, in the sense that if the forward J-AAN can accurately anticipate the future actions from the past actions, then the backward J-AAN is expected to translate it back from the future actions to the past actions. During the testing, we only use the forward J-AAN to anticipate future actions.

III. Proposed Approach

A. Problem Formulation

Let $V = \{V_i\}_{i=1}^{T}$ be a video with T frames, where V_i is the i-th frame in the video. Given the first few frames of a video, we seek to anticipate the future actions, along with their duration, which will be happening in the remainder of that video. Specifically, given the first t frames of a video $V_{1:t} = \{V_1, \ldots, V_t\}$ corresponding to n actions $A_{1:n} = \{a_1, \ldots, a_n\} \in R^{n \times C}$ (each $a_i \in R^{1 \times C}$), the task is to anticipate future actions $\tilde{A}_{n+1:N} = \{\tilde{a}_{n+1}, \ldots, \tilde{a}_N\} \in R^{(N-n) \times C}$ and their corresponding duration $\tilde{\ell}_{n+1:N} = \{\tilde{\ell}_{n+1}, \ldots, \tilde{\ell}_N\} \in R^{(N-n) \times 1}$ that will be happening in the remainder of that video $V_{t+1:T}$, where N is the total number of actions in that video, and C is the number of action classes.

For our forward and backward networks, we define the forward and backward frames of different intervals as $V^{f}_{i:j}$ and $V^{b}_{j:i}$, respectively, where the superscripts f and b represent the forward and backward directions, respectively, and the subscript $(i:j)$ represents that the frame starts at time i and ends at time j, and vice versa for $(j:i)$. For example, $V^{f}_{1:t}$ represents the forward past frames, while $V^{b}_{T:t+1}$ represents the backward future frames. Similarly, $\tilde{A}^{f}_{n+1:N}$ and $\tilde{\ell}^{f}_{n+1:N}$ represent the anticipated forward future actions and their duration, respectively, while $\tilde{A}^{b}_{n:1}$ and $\tilde{\ell}^{b}_{n:1}$ represent the anticipated backward past actions and their duration, respectively.

B. Method Overview

As shown in Fig. 3, we propose a novel Jointly-Learnt Action Anticipation Network (J-AAN) via self-knowledge distillation and cycle consistency for future action anticipation. Our three major ideas are as follows:

Jointly-learnt Action Anticipation Network: The Jointly-learnt Action Anticipation Network (J-AAN) anticipates future actions and their duration jointly in both direct and recursive ways from the observed frames. Our J-AAN contains a recurrent encoder, a direct anticipation module, and a recursive anticipation module. The recurrent encoder encodes the actions of the observed frames into a single feature vector, which is then jointly used by the direct and recursive anticipation modules. The direct anticipation module directly anticipates all the future actions and their duration in one single step, while the recursive anticipation module recursively anticipates the future actions and their duration. During the recursive anticipation, the direct anticipation supports the recurrent decoder to improve the performance of the future action anticipation.

Self-Knowledge Distillation: We design a Self-Knowledge Distillation (Self-KD) mechanism to train J-AAN, where the J-AAN gradually distills its own anticipation to soften the hard labels during the training. More specifically, during the training at epoch k, our J-AAN is trained with target labels that are generated from the combination of the hard labels of the ground-truth future actions and the anticipated future actions (soft labels) from the previous epoch $k-1$, to handle the uncertainty of future actions.

Cycle consistency: We design a forward and backward anticipation framework based on a cyclic consistency constraint. Both the forward and backward networks are constructed with J-AAN. The forward J-AAN takes the forward frame-wise past actions as the input to anticipate the forward future actions

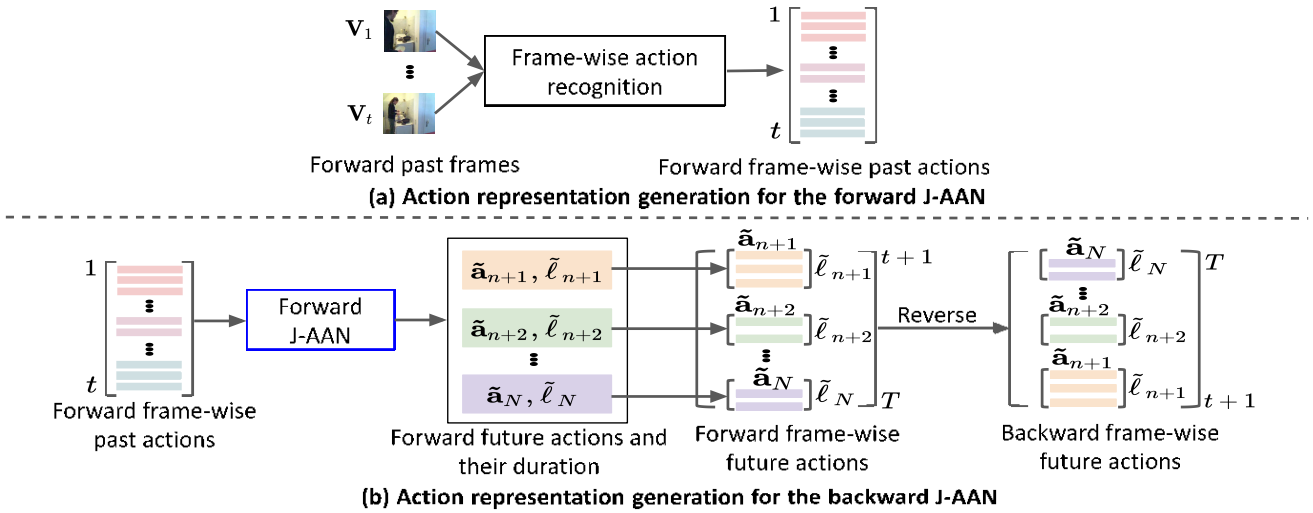


Fig. 4. The action representation generation for the forward J-AAN and the backward J-AAN.

and their duration, while the backward J-AAN evaluates the anticipation of the forward J-AAN by anticipating the backward past actions and their duration from the anticipation of the forward J-AAN. Note that, during the testing, we only use the forward J-AAN to anticipate future actions.

C. Action Representation Generation

Before discussing the architecture of the forward and backward J-AANs, we introduce their inputs - an action representation based on frame-wise action labels, as shown in Fig. 4. When generating the action representation for the forward J-AAN (4(a)), we first separately train a recognition network to infer the action label of each frame (since the anticipation of the future actions significantly depends on the performed actions within the observed part of the videos) of the forward past frames, $V^f_{1:t}$. After that, we convert the inferred action labels to one-hot-vectors and stack them in a matrix with the size of $t \times C$ as the representations of forward frame-wise past actions, $X^f_{1:t}$. The forward J-AAN takes these forward frame-wise past actions as an input to anticipate the forward future actions and their duration.

On the other hand, the action representation for the backward J-AAN (i.e., the backward frame-wise future actions) are generated from the output of the forward J-AAN, as illustrated in Fig. 4(b). Specifically, the anticipated future actions from the forward J-AAN are stacked in a matrix according to their duration to get the forward frame-wise future actions, $X^f_{t+1:T}$, which is then reversed in the temporal domain to get the backward frame-wise future actions, $X^b_{T:t+1}$. The backward J-AAN takes these backward frame-wise future actions as an input to anticipate the backward past actions and their duration.

D. Jointly-learnt Action Anticipation Network (J-AAN)

Both the forward J-AAN and the backward J-AAN share the same network architecture. In this section, we use the forward J-AAN as an example to explain our network design. As shown in Fig. 5, our forward J-AAN consists of three key components: (1) a recurrent encoder; (2) a direct anticipation; and (3) a recursive anticipation.

Recurrent encoder: The recurrent encoder encodes the frame-wise action labels of the observed frames into a single vector that will be used to anticipate future actions. Formally, the recurrent encoder loads the forward frame-wise past actions $X^f_{1:t} \in \mathbb{R}^{t \times C}$ into a Gated Recurrent Unit (GRU) to capture the temporal pattern of the past action sequences and encode them into a single vector, as follows:

$$h^e_\tau = GRU(X^f_\tau, h^e_{\tau-1}), \text{ where } \tau = 1, ..., t \quad (1)$$

where X^f_τ is the input action representation of the forward past frame at time τ, h^e_τ and $h^e_{\tau-1}$ are the hidden states at time τ and $\tau - 1$, respectively. The superscript e denotes the 'encoder'. The hidden state at the last time step, h^e_t, encodes all the forward frame-wise past actions.

Direct anticipation: The direct anticipation module anticipates all the future actions and their corresponding duration in one single step. Formally, the direct anticipation module loads the encoded feature vector h^e_t and anticipates the forward future actions, $(\tilde{A}^{dir}_{n+1:N})^f = [(\tilde{a}^{dir}_{n+1})^f, ..., (\tilde{a}^{dir}_N)^f]$, and their duration, $(\tilde{\ell}^{dir}_{n+1:N})^f = [(\tilde{\ell}^{dir}_{n+1})^f, ..., (\tilde{\ell}^{dir}_N)^f]$. The direct anticipation module consists of two separate branches (each branch is configured with a fully-connected layer) to anticipate the future actions and their duration. Mathematically, the future actions and their duration are directly anticipated as follows:

$$(\tilde{A}^{dir}_{n+1:N})^f = Reshape(W^{dir}_A h^e_t) \quad (2)$$

$$(\tilde{\ell}^{dir}_{n+1:N})^f = Reshape(W^{dir}_L h^e_t) \quad (3)$$

where W^{dir}_A and W^{dir}_L are the trainable parameters.

Recursive anticipation: The recursive anticipation aims to recursively anticipate the forward future actions, $(\tilde{A}^{rec}_{n+1:N})^f = [(\tilde{a}^{rec}_{n+1})^f, ..., (\tilde{a}^{rec}_N)^f]$, and their duration, $(\tilde{\ell}^{rec}_{n+1:N})^f = [(\tilde{\ell}^{rec}_{n+1})^f, ..., (\tilde{\ell}^{rec}_N)^f]$. The recursive anticipation contains two components: (1) a recursive initialization module

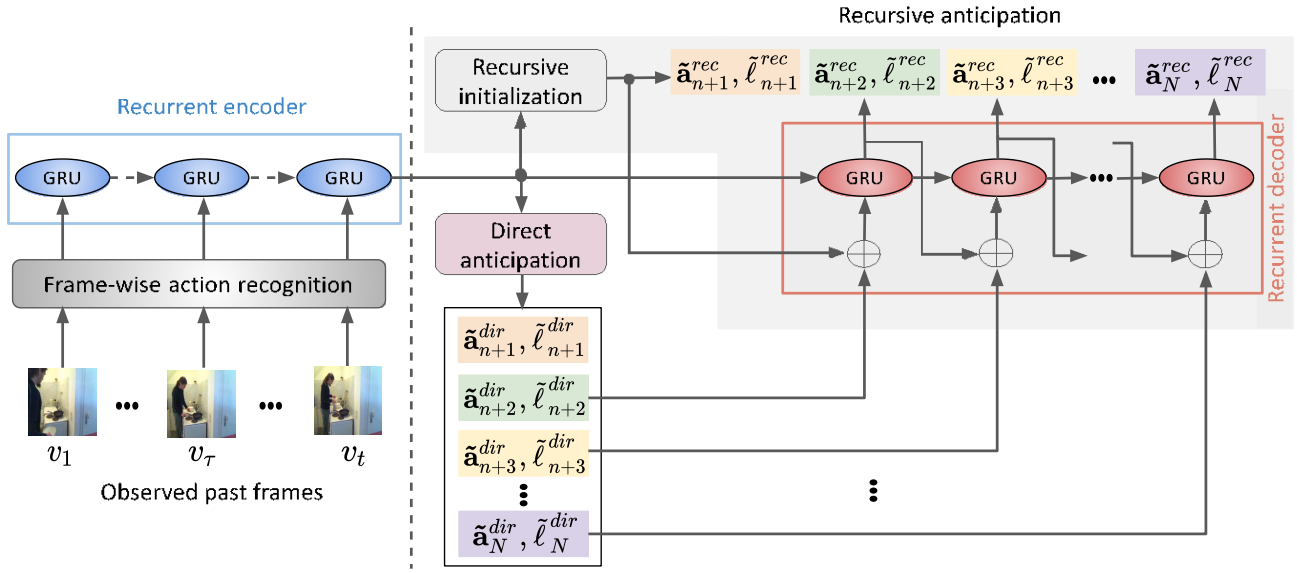


Fig. 5. Illustration of our proposed Jointly-learnt Action Anticipation Network (J-AAN). The J-AAN contains a recurrent encoder, a direct anticipation module, and a recursive anticipation module. The recurrent encoder encodes the frame-wise actions over different time steps of the observed frames. The direct anticipation module takes the encoded features and directly anticipates all the actions and their corresponding duration, while the recursive anticipation contains a recursive initialization and a recurrent decoder to recursively anticipate the future actions and their duration. During the recursive anticipation, the recurrent decoder takes the anticipated actions from the direct anticipation as one of the inputs for better future action anticipation.

that anticipates the immediate future, i.e., the (n+1)-th forward future action $(\tilde{a}^{rec}_{n+1})^f$ and its duration $(\tilde{\ell}^{rec}_{n+1})^f$ to initialize the recurrent decoder; and (2) a recurrent decoder that recursively anticipates the remaining forward future actions $(\tilde{A}^{rec}_{n+2:N})^f$ from $n+2$ to N, and their corresponding duration $(\tilde{\ell}^{rec}_{n+2:N})^f$.

(a) Recursive initialization: The recursive initialization module loads the encoded feature h^e_t and anticipates the immediate future action and its duration. The recursive initialization module consists of two separate branches, where each branch is configured with a fully-connected layer. One branch with a fully-connected layer anticipates the immediate (i.e. the $n+1$-th) future action $(\tilde{a}^{rec}_{n+1})^f$, while the other branch with another fully-connected layer anticipates the duration $(\tilde{\ell}^{rec}_{n+1})^f$ of that immediate future action.

Please note that we use two fully-connected layers in the recursive initialization module to respectively anticipate the immediate (i.e., the $n+1$-th) future action and its duration, while we use two fully-connected layers in the direct initialization module to respectively anticipate all the unseen (i.e., from $n+1$ to N) future actions and their duration

(b) Recurrent decoder: Given the outputs of the encoder and the direct anticipation module, the recurrent decoder recursively anticipates the future actions and their duration. The recurrent decoder consists of GRU and the hidden state at each time step is updated as follows:

$$h^d_m = \mathrm{GRU}([(\tilde{a}^{rec}_{m-1})^f, (\tilde{a}^{dir}_m)^f], h^d_{m-1}), m = n+2, ..., N \quad (4)$$

where the input of the recurrent decoder at time step m is the concatenation of the anticipated actions $(\tilde{a}^{rec}_{m-1})^f$ and $(\tilde{a}^{dir}_m)^f$ i.e., the anticipated action at the previous step by recursive anticipation and the anticipated action at the current step by direct anticipation, respectively. The hidden states h^d_m and h^d_{m-1} are the current and previous hidden states of the decoder, respectively. The superscript d denotes the 'decoder'. During the first time step of the recurrent decoder, the input is initialized with the concatenation of the $(n+1)$-th forward future action anticipated by the recursive initialization module and the $(n+2)$-th forward future action anticipated by the direct anticipation module, and the output of the recurrent encoder h^e_t is used as the previous hidden state.

Given the hidden state h^d_m at each time step, the future action and its duration are anticipated as follows:

$$(\tilde{a}^{rec}_m)^f = W^{rec}_A h^d_m \quad (5)$$

$$(\tilde{\ell}^{rec}_m)^f = W^{rec}_L h^d_m \quad (6)$$

where $(\tilde{a}^{rec}_m)^f$ and $(\tilde{\ell}^{rec}_m)^f$ are the anticipated forward future action and its duration from the recurrent decoder at time step m. W^{rec}_A and W^{rec}_L are the trainable parameters.

E. Self-Knowledge Distillation

Since multiple action sequences may follow the same action, training an action anticipation model with hard labels that assign zero probabilities to incorrect yet semantically similar actions may not handle the future's uncertainty. Therefore, we design a Self-Knowledge Distillation (Self-KD) mechanism in the J-AAN, where our J-AAN gradually utilizes its previous anticipations to have more informative supervision during training to handle the future's uncertainty.

Formally, let the forward J-AAN directly and recursively anticipates the forward future actions and their duration from the forward frame-wise past actions at epoch k. Mathematically, we get the following two functions from the forward J-AAN at epoch k:

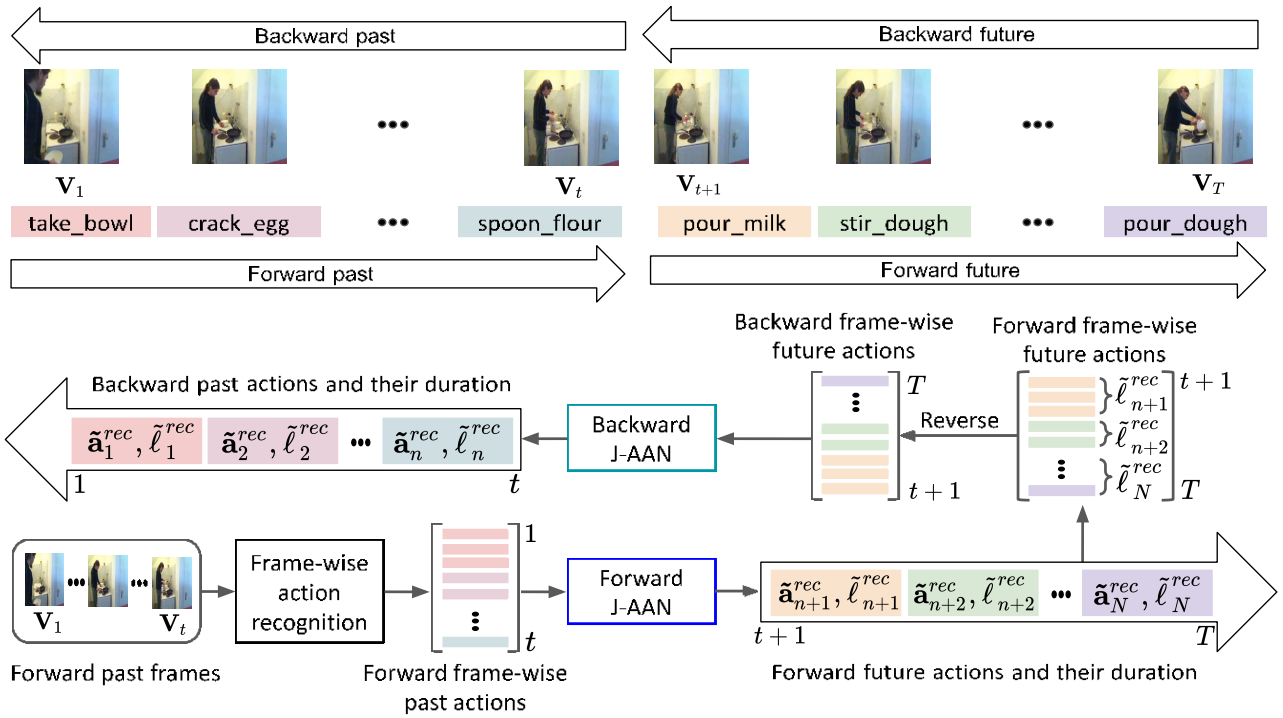


Fig. 6. Illustration of our proposed forward and backward J-AANs. The forward J-AAN anticipates the forward future actions and their duration from the forward frame-wise past actions. The anticipated future actions are stacked in a matrix according to their duration to get the forward frame-wise future actions, which is then reversed in temporal domain to get the backward frame-wise future actions. The backward J-AAN loads the backward frame-wise future actions and anticipates the backward past actions and their corresponding duration, which satisfies a cycle consistency. Note that, both the forward J-AAN and the backward J-AAN perform their corresponding anticipation jointly in both direct and recursive ways. For the simplicity, we show the output of the recursive anticipation for each J-AAN.

Forward direct anti.: $X^{f}_{1:t} \rightarrow ((\tilde{A}^{dir}_{n+1:N})^{f}_{k}, (\tilde{\ell}^{dir}_{n+1:N})^{f}_{k})$

Forward recursive anti.: $X^{f}_{1:t} \rightarrow ((\tilde{A}^{rec}_{n+1:N})^{f}_{k}, (\tilde{\ell}^{rec}_{n+1:N})^{f}_{k})$

Then our forward loss for the direct anticipation at k-th epoch can be written as:

$$(L^{dir}_{f})_k = ||(\alpha A^{f}_{n+1:N} + (1-\alpha)(\tilde{A}^{dir}_{n+1:N})^{f}_{k-1}) - (\tilde{A}^{dir}_{n+1:N})^{f}_{k}||_2 + ||\ell_{n+1:N} - (\tilde{\ell}^{dir}_{n+1:N})^{f}_{k}||_2 \quad (7)$$

where $A^{f}_{n+1:N}$ is the hard labels of the ground-truth forward future actions and $\ell^{f}_{n+1:N}$ is their corresponding ground-truth duration, $(\alpha A^{f}_{n+1:N} + (1-\alpha)(\tilde{A}^{dir}_{n+1:N})^{f}_{k-1})$ is the soft labels of the forward future actions that are generated from the combination of the hard labels of the ground-truth forward future actions and the directly anticipated forward future actions of the previous epoch, and α is a hyper-parameter. Similarly, our forward loss for the recursive anticipation at epoch k can be written as:

$$(L^{rec}_{f})_k = ||(\alpha A^{f}_{n+1:N} + (1-\alpha)(\tilde{A}^{rec}_{n+1:N})^{f}_{k-1}) - (\tilde{A}^{rec}_{n+1:N})^{f}_{k}||_2 + ||\ell_{n+1:N} - (\tilde{\ell}^{rec}_{n+1:N})^{f}_{k}||_2 \quad (8)$$

where $(\alpha A^{f}_{n+1:N} + (1-\alpha)(\tilde{A}^{rec}_{n+1:N})^{f}_{k-1})$ is the soft labels of the forward future actions for the recursive anticipation. Note that, we use the anticipated action probabilities over C action classes from epoch $(k-1)$ for softening the hard labels to handle the uncertainty on future action anticipation, not the duration since we only anticipate a single value for the duration at each time step. Finally, the anticipation loss with Self-KD for the forward J-AAN is computed as:

$$(L^{S-KD}_{Anti})_k = (L^{dir}_{f})_k + (L^{rec}_{f})_k \quad (9)$$

F. Cycle Consistency

Since human actions are probably anticipable in both forward and backward directions, we design a forward and backward anticipation framework with our proposed J-AAN based on a cyclic consistency constraint, as shown in Fig. 6. Given the predicted future actions and their corresponding duration from the forward J-AAN, we then anticipate the past actions and their duration from the backward J-ANN. The goal is to enforce the output of the backward J-ANN to be consistent with the input of the forward J-ANN.

Specifically, the forward J-AAN directly $((\tilde{A}^{dir}_{n+1:N})^{f}, (\tilde{\ell}^{dir}_{n+1:N})^{f})$ and recursively $((\tilde{A}^{rec}_{n+1:N})^{f}, (\tilde{\ell}^{rec}_{n+1:N})^{f})$ anticipates the forward future actions and their corresponding duration from the forward frame-wise past actions. Since the actions anticipated by the direct anticipation module are already fed into the recursive anticipation module as one of its input, we only use the recursively anticipated future actions and their duration to generate the input to the backward J-AAN. First, the recursively anticipated future actions are stacked in a matrix according to their duration to get the forward frame-wise future actions $X^{f}_{t+1:T}$, which is then reversed

TABLE I
COMPARISON WITH OTHER STATE-OF-THE-ART METHODS ON BREAKFAST DATASET FOR FUTURE ACTION ANTICIPATION FROM THE INFERRED ACTION LABELS OF THE OBSERVED FRAMES, USING MEAN-OVER-CLASSES (MOC) ACCURACY.

Observation first →	20%				30%			
Prediction following →	10%	20%	30%	50%	10%	20%	30%	50%
Grammar (reported from [57])	16.6	15.0	13.5	13.4	21.1	18.2	17.5	16.3
Nearest-Neighbor (reported from [57])	16.4	15.0	14.5	13.3	19.9	18.6	18.0	16.6
RNN model [57], CVPR2018	18.1	17.2	15.9	15.8	21.6	20.0	19.7	19.2
CNN model [57], CVPR2018	17.9	16.4	15.4	14.5	22.4	20.1	19.7	18.8
Time-Conditioned [58], CVPR2019	18.4	17.2	16.4	15.8	22.8	20.4	19.6	19.8
TAB [59], ECCV2020	37.4	31.2	30.0	26.1	39.5	34.1	31.0	27.9
Proposed approach	39.9	36.1	35.7	33.8	44.9	41.5	39.8	37.5

in the temporal domain to get the backward frame-wise future actions $X^b_{T:t+1}$. The backward J-AAN loads these backward frame-wise future actions and anticipates the backward past actions and their corresponding duration in both direct and recursive ways. Mathematically, we obtain the following two functions from the backward J-AAN at epoch k:

Backward direct anti.: $X^b_{T:t+1} \rightarrow ((\tilde{A}^{dir}_{n:1})^b_k, (\tilde{\ell}^{dir}_{n:1})^b_k)$

Backward recursive anti.: $X^b_{T:t+1} \rightarrow ((\tilde{A}^{rec}_{n:1})^b_k, (\tilde{\ell}^{rec}_{n:1})^b_k)$

The backward loss at epoch k for the direct anticipation can be written as:

$$(L^{dir}_b)_k = ||A^b_{n:1} - (\tilde{A}^{dir}_{n:1})^b_k||_2 + ||\ell^b_{n:1} - (\tilde{\ell}^{dir}_{n:1})^b_k||_2 \quad (10)$$

While the backward loss for the recursive anticipation at k-th epoch can be written as:

$$(L^{rec}_b)_k = ||A^b_{n:1} - (\tilde{A}^{rec}_{n:1})^b_k||_2 + ||\ell^b_{n:1} - (\tilde{\ell}^{rec}_{n:1})^b_k||_2 \quad (11)$$

where $A^b_{n:1}$ and $\ell^b_{n:1}$ are the ground-truth backward future actions and their duration, respectively. Finally, the cycle consistency loss for the backward J-AAN is computed from the combination of the backward loss for the direct anticipation and the backward loss for the recursive anticipation:

$$(L_{cyc})_k = (L^{dir}_b)_k + (L^{rec}_b)_k \quad (12)$$

G. Training and Inference

To train our forward and backward J-AANs, we sum up all the related losses:

$$\begin{aligned} L_k &= (L^{S-KD}_{anti})_k + (L_{cyc})_k \\ &= ((L^{dir}_f)_k + (L^{rec}_f)_k) + ((L^{dir}_b)_k + (L^{rec}_b)_k) \end{aligned} \quad (13)$$

At the inference, we only use the forward J-AAN for our future action anticipation.

IV. EXPERIMENTS

A. Datasets

Breakfast Dataset [34]: This dataset contains 1712 videos of 52 different actors preparing breakfast meals. Overall, there are 48 fine-grained action classes. The videos are recorded in 18 different kitchens. The dataset is provided with four different train/test splits. For evaluation, we quantify the performance with the average scores over the four splits.

50Salads Dataset [35]: This dataset contains 50 videos with 17 fine-grained action classes. All the videos correspond to salad preparation activities and the actions are performed by 25 subjects. We perform five-fold cross-validation for evaluation using the splits provided by [35] and report the average scores.

Epic-Kitchens-55 Dataset [36]: This dataset is a large-scale and fine-grained cooking video dataset, which involves 2513 unique actions. The actions are performed by 32 participants in diverse kitchen environments. Following the literature [53], [75], [76], [77], we follow the same experimental setting in [33], where the 28,472 activity segments in the public training set are further split into 23,493 segments for training and 4,979 segments for validation.

B. Evaluation Metric

Following the literature [57], [58], [59], [60], [74], we use the Mean over Classes (MoC) as the quantitative evaluation metric to evaluate the performance of the future action anticipation on both the Breakfast and 50Salads datasets. We also follow the standard training and testing protocol: for each video in the training set, 4 training examples are generated by using the first 10%, 20%, 30%, and 50% of the video, respectively, as observation and the following 50% of the video as ground-truth for the anticipation; for the testing, we observe the first 20% or 30% of the video and anticipate the following 10%, 20%, 30%, and 50% of that video. On the other hand, following the literature [33], [52], [53], [77], we use the Top-K accuracy, i.e., we assume a prediction correct if the ground truth action falls in the Top-K predictions, to evaluate the performance of the future action anticipation on the EPIC-Kitchens-55 dataset.

C. Implementation Details

Our forward and backward Jointly-learnt Action Anticipation Networks (J-AANs) are constructed with a recurrent encoder, a direct anticipation module, and a recursive anticipation module. The recurrent encoder is configured with GRUs and hidden state's dimension of the GRU is set to 512. The output of the recurrent encoder is fed into the direct anticipation and recursive anticipation modules. The direct anticipation module learns the parameters W^{dir}_A and W^{dir}_L to directly anticipate all the actions and their corresponding duration, while the recurrent decoder in recursive anticipation module is configured with GRUs (hidden state is set to 512) to recursively anticipate

TABLE II
COMPARISON WITH OTHER STATE-OF-THE-ART METHODS ON 50SALADS DATASET FOR FUTURE ACTION ANTICIPATION FROM THE INFERRED ACTION LABELS OF THE OBSERVED FRAMES, USING MEAN-OVER-CLASSES (MOC) ACCURACY.

Observation first →	20%				30%			
Prediction following →	10%	20%	30%	50%	10%	20%	30%	50%
Grammar (reported from [57])	24.7	22.3	19.8	12.7	29.7	19.2	15.2	13.1
Nearest-Neighbor (reported from [57])	19.0	16.1	14.1	10.4	21.6	15.5	13.5	13.9
RNN model [57], CVPR2018	30.1	25.4	18.7	13.5	30.8	17.2	14.8	09.8
CNN model [57], CVPR2018	21.2	19.0	16.0	09.9	29.1	20.1	17.5	10.9
Time-Conditioned [58], CVPR2019	32.5	27.6	21.3	16.0	35.1	27.1	22.1	15.6
TAB [59], ECCV2020	34.7	25.9	23.7	15.7	34.5	26.1	19.0	15.5
Proposed approach	36.9	29.8	24.9	17.7	37.5	30.1	25.6	18.1

TABLE III
COMPARISON WITH OTHER STATE-OF-THE-ART METHODS ON BREAKFAST AND 50SALADS DATASETS FOR FUTURE ACTION ANTICIPATION FROM THE GROUND-TRUTH ACTION LABELS OF THE OBSERVATION, USING MEAN-OVER-CLASSES (MOC) ACCURACY.

	Breakfast				50Salads			
Observation first →	30%				30%			
Prediction following →	10%	20%	30%	50%	10%	20%	30%	50%
Grammar (reported in [57])	52.3	42.2	38.4	33.1	26.7	14.6	11.7	09.3
Nearest-Neighbor (reported in [57])	44.2	37.7	35.7	30.2	22.1	17.2	18.4	14.7
RNN model [57], CVPR2018	61.5	50.3	44.9	41.8	44.2	29.5	20.0	10.4
CNN model [57], CVPR2018	60.3	50.1	45.2	40.5	37.4	24.8	20.8	14.1
Time-Conditioned [58], CVPR2019	66.0	55.9	49.1	44.2	46.4	34.8	25.2	13.8
TAB [59], ECCV2020	67.4	56.1	47.4	41.5	44.8	32.7	23.5	15.3
Proposed approach	71.1	59.7	54.2	50.6	48.2	37.1	28.4	18.5

TABLE IV
COMPARISON WITH OTHER STATE-OF-THE-ART METHODS ON BREAKFAST AND 50SALADS DATASETS FOR FUTURE ACTION ANTICIPATION DIRECTLY FROM THE FRAME-WISE FEATURES OF THE OBSERVATION, USING MEAN-OVER-CLASSES (MOC) ACCURACY.

	Breakfast				50Salads			
Observation first →	30%				30%			
Prediction following →	10%	20%	30%	50%	10%	20%	30%	50%
CNN model [57], CVPR2018	17.7	16.9	15.5	14.1	-	-	-	-
Sequence-to-Sequence [74], GCPR2020	29.7	27.4	25.6	25.2	34.4	23.7	18.9	15.9
TAB [59], ECCV2020	30.4	26.3	23.8	21.2	30.6	22.5	19.1	11.2
FUTR [60], CVPR2022	32.3	29.9	27.5	25.9	35.2	24.9	24.2	15.3
Proposed approach	33.6	31.0	28.4	26.5	34.9	25.8	24.4	16.1

the future actions and their corresponding duration. During the training, we train the anticipation module with ground truth action labels, whereas the labels generated by the action representation module are used during the inference. Please note that, at the inference, we only use the forward J-AAN for our future action anticipation from the observation of the forward frame-wise past actions.

D. Comparison with State-of-the-arts

Anticipation on Breakfast and 50Sladas: Table I and Table II show the comparison results of our proposed approach with other state-of-the-art methods on Breakfast and 50Salads datasets for the long-term future action anticipation, where the future actions are anticipated based on the inferred action (obtain from an action recognition network) labels of the observed frames. We use the I3D features [3] and Temporal Convolutional Networks (TCN) [81] as our action recognition module on the observed frames. As shown in Table I and Table II, our approach gives a significant boost in performance, outperforming all the latest methods and establishing a new state-of-the-art on Breakfast and 50Salads.

We also compare the performance of our approach with state-of-the-art methods on Breakfast and 50Salads datasets using the ground-truth action labels on the observed frames. Since every method to be compared has the same perfect labels (ground truth) on the observed frames, this comparison can show the effectiveness of each action anticipation method clearly. As shown in Table III, our method outperforms all the latest state-of-the-arts again when using the ground-truth action labels of the observed frames.

Finally, to show the effectiveness of our proposed approach, we further compare the anticipation performance with other methods, where we directly apply our method on the observed frame-wise features to anticipate the future actions. For this experiment, as same as the latest works of future action anticipation [60], [74] , we use the I3D features [3] as the input observed features. The results are shown in Table IV, where we achieve the superior performance compared to the latest works of future action anticipation directly from the features, for both the Breakfast and 50Salads datasets.

Anticipation on EPIC-Kitchens-55: Table V shows the comparison results of our proposed method with other state-

TABLE V
COMPARING OUR PROPOSED APPROACH WITH OTHER STATE-OF-THE-ART METHODS ON EPIC-KITCHENS-55 DATASET, WHERE THE TASK IS TO ANTICIPATE THE FUTURE ACTION ONE SECOND BEFORE IT STARTS.

Method	Top-1 acc.	Top-5 acc.
RL [78], CVPR2016	-	29.61
VN-CE [79], ECCV2018	5.79	17.31
SVM-TOP3 [80], ECCVW2018	11.09	25.42
RU-LSTM [33], ICCV2019	-	35.32
FIA [75], ACMMM2020	14.07	33.37
AVHN [76], ICCVW2019	19.29	35.91
LS [77], ICPR2021	-	35.90
SRL [52], PAMI2021	-	35.52
HRO [53], CVPR2022	-	37.42
Proposed approach	20.31	38.94

TABLE VI
ABLATION STUDY ON DIFFERENT MODULES OF THE PROPOSED APPROACH ON BREAKFAST DATASET FOR BOTH GROUND TRUTH (GT) AND WITHOUT GROUND TRUTH OBSERVATIONS, WHERE THE OBSERVATION IS THE FIRST 30% OF THE VIDEOS AND THE ANTICIPATION IS THE FOLLOWING 50%.

Methods	MoC (w/GT obs.)	MoC (w/o GT obs.)
(i) Baseline (RNN model [57])	41.8	18.8
(ii) Direct anticipation	41.3	25.1
(iii) Recursive anticipation	43.5	27.3
(iv) Jointly-learnt Action Anticipation Network (J-AAN) (Direct + Recursive anticipation)	47.1	32.5
(v) J-AAN + Self-knowledge distillation	49.8	35.8
(vi) J-AAN + Self-knowledge distillation + Cycle consistency	50.6	37.5

of-the-art methods on the EPIC-Kitchens-55 dataset, where the task is to anticipate the future action one second before it starts. For the fair comparison as same as the latest works [52], [53], [76], [77], we directly use the appearance, motion, and object features extracted from each time-step provided by [33]. Similar to the existing methods, we first train our model separately for each feature modality, and then the final anticipation results are obtained by a late fusion of predictions from the different modalities. As shown in Table V, our method achieves superior performance compared to other state-of-the-art methods on EPIC-Kitchens-55 dataset, on both Top-1 and Top-5 accuracy.

E. Ablation Studies

To systematically evaluate our method and study the contribution of each module, we perform several ablation studies on the Breakfast dataset:

(i) Baseline: Since the RNN model [57] leverages recursive prediction at the inference, we use this method as a baseline.

(ii) Direct anticipation: We apply a recurrent encoder and a direct anticipation module to directly anticipate all the future actions and their corresponding duration.

(iii) Recursive anticipation: We apply a recurrent encoder and a recursive anticipation module to recursively anticipate the future actions and their corresponding duration.

(iv) Jointly-learnt Action Anticipation Network (J-AAN) (Direct + Recursive anticipation): We jointly train both the direction and recursive anticipation modules on top a recurrent encoder for anticipating the future actions and their duration.

(v) J-AAN + Self-knowledge distillation: In addition to (iv), we apply self-knowledge distillation mechanism, where the J-AAN graudally distills its own knowledge to soften the hard labels during the training.

(vi) J-AAN + Self-knowledge distillation + Cycle consistency: in addition to (v), we apply the cycle consistency for which we design forward-backward J-AANs. The forward J-AAN anticipates the future actions from the observed past actions, and the backward J-AAN anticipates the past actions from the anticipated future actions, satisfying a cyclic consistency constraint. This is our proposed approach.

In Table VI, by comparing (iv) with (ii) and (iii), we observe that the joint anticipation improves the anticipation performance compared to the individual direct and recursive baseline, respectively, which verifies the observation 1 described in the introduction. Comparing (iv) and (v), we can see that the self-knowledge distillation can improve the anticipation performance by handling the uncertainty on future action anticipation, which justifies the observation 2. Finally, comparing (v) and (vi), we observe that the cycle consistency further helps to improve the anticipation performance, which verifies the observation 3.

F. Qualitative analysis

We present some qualitative results on different modules of the proposed approach on the test set of Breakfast and 50Salads datasets, as shown in Fig. 7-9, where our model observes the first 30% of the video and anticipates the frame-wise actions of the following 50% of that video.

Importance of Jointly-learnt Action Anticipation Network (J-AAN): Fig. 7 shows the importance of our J-AAN compared to the individual direct and recursive anticipation. The J-AAN anticipates the future actions jointly in both direct and recursive ways. In joint learning, the fusion of these

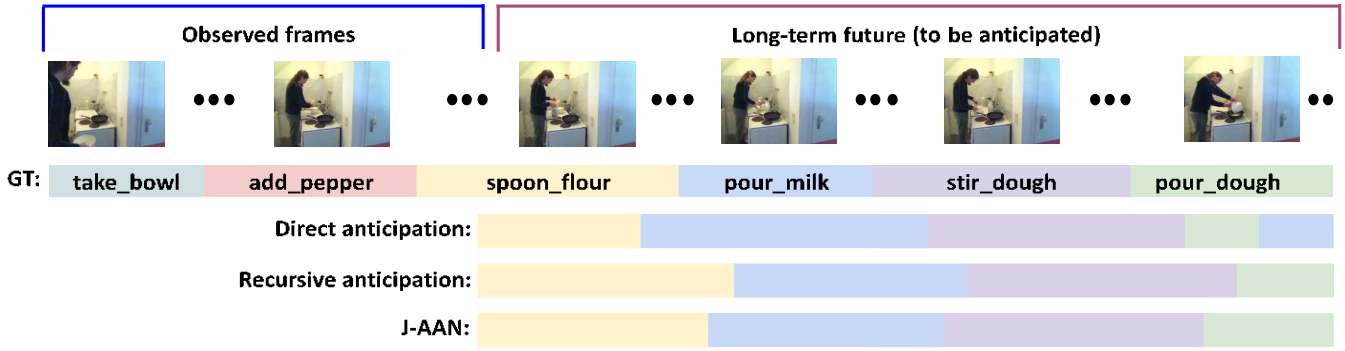


Fig. 7. Importance of Jointly-learnt Action Anticipation Network (J-AAN). The J-AAN anticipates the future action jointly in both direct and recursive ways. The joint learning of these two anticipation approaches performs better action anticipation compared to the individual ones. The model observes the first 30% of the video and anticipates the frame-wise actions of following 50% of that video. (GT: ground truth).

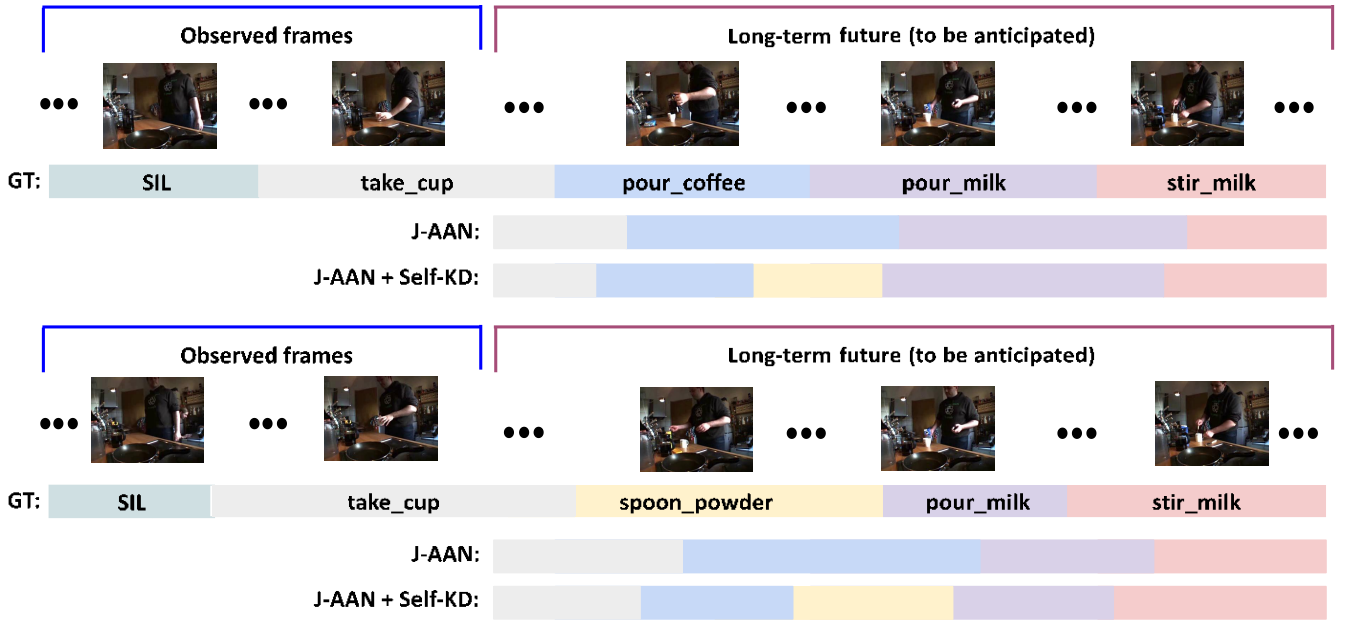


Fig. 8. Impact of the Self-Knowledge Distillation (Self-KD) mechanism. Both actions "pour_coffee" and "spoon_powder" follow the same observed actions. The J-AAN trained without Self-KD mechanism anticipates the same action for both cases. The Self-KD mechanism can handle this uncertainty on the future action anticipation. The model observes the first 30% of the video and anticipates the frame-wise actions of following 50% of that video. (GT: ground truth).

two anticipation approaches performs better action anticipation compared to the individual direct or recursive anticipation, as shown in Fig. 7.

Impact of Self-Knowledge Distillation (Self-KD) mechanism: Fig. 8 shows the impact of the Self-KD mechanism in our proposed J-AAN. In Fig. 8, we see the uncertainty on future action anticipation, where both actions "pour_coffee" and "spoon_powder" follow the same observed actions. As shown in Fig. 8, the J-AAN trained without the Self-KD mechanism anticipates the same action for both cases, while the J-AAN trained with the Self-KD mechanism is capable of handling the uncertainty on the future action anticipation.

Impact of cycle consistency loss: Fig. 9 shows the impact of cycle consistency loss in our proposed J-AAN. Since the cycle consistency anticipates the past actions from the anticipated future actions, it can verify whether all the required future actions are anticipated or not. As shown in Fig. 9, without the cycle consistency loss, the J-AAN did not anticipate the "add oil" action, which is an intermediate step to prepare salads. The cycle consistency resolves this issue and the J-AAN can anticipate the complete set of the future actions.

G. Discussions and future works

In this paper, we propose a Jointly-learnt Action Anticipation Network (J-AAN) via self-knowledge distillation and cycle consistency, to anticipate actions in a longer time horizon from the observed past actions. Our proposed J-AAN anticipates the future actions jointly in both direct and recursive ways. The performance improvement from our J-AAN compared to the sole direct and recursive anticipation indicates that the joint learning process can encode the highly informative information from the observed past actions.

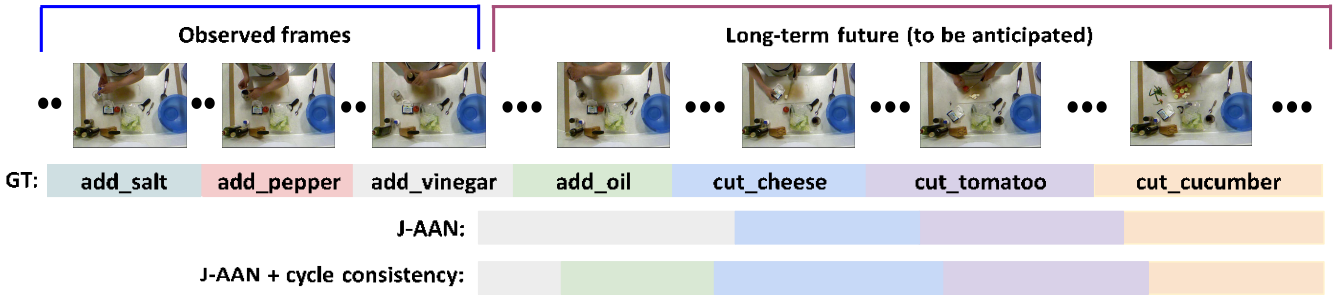


Fig. 9. Impact of the cycle consistency loss. By anticipating the past actions from the anticipated future actions, the cycle consistency verifies whether the complete set of the future actions are anticipated or not. Without the cycle consistency loss, the J-AAN missed to anticipate the "add_oil" action. The cycle consistency resolves this issue. The model observes the first 30% of the video and anticipates the frame-wise actions of following 50% of that video.

TABLE VII
ANTICIPATION ON BREAKFAST, GIVEN DIFFERENT PERCENTAGES OF BOTH GROUND TRUTH (GT) AND INFERRED ACTION LABELS OBSERVATIONS.

Observation first →	10%	20%	30%	50%
Prediction following →	50%	50%	50%	50%
A	From GT labels			
Our approach	35.7	48.8	50.6	53.5
B	From inferred action labels			
Our approach	20.9	33.8	37.5	40.3

TABLE VIII
DIFFERENT PERCENTAGES OF ANTICIPATION ON BREAKFAST FOR BOTH GROUND TRUTH (GT) AND INFERRED ACTION LABELS OBSERVATIONS.

Observation first →	50%	50%	50%	50%
Prediction following →	10%	20%	30%	50%
A	From GT labels			
Our approach	74.7	63.1	57.9	53.5
B	From inferred action labels			
Our approach	47.1	44.8	42.6	40.3

Intuitively, training an action anticipation model with hard labels that assigns zero probabilities to semantically similar actions may not handle the future's uncertainty. Therefore, we design a Self-Knowledge Distillation (Self-KD) mechanism to train our J-AAN, where the J-AAN gradually distills its own knowledge to soften the hard labels during the training to handle the uncertainty on future action anticipation. The significant improvement in action anticipation after using the Self-KD mechanism to train the J-AAN indicates that J-AAN with Self-KD mechanism can handle the future's uncertainty. In addition to Self-KD, we design a forward and backward mechanism to train the J-AAN, where the backward J-AAN verifies the anticipation of the forward J-AAN based on a cyclic consistency constraint. The further improvement in action anticipation after using the cycle consistency constraint indicates that the backward J-AAN helps the forward J-AAN to learn better for future action anticipation.

Although we achieve superior performance compared to the state-of-the-arts, as shown in Tables I - V, the future action anticipation is still challenging, particularly anticipating the far-future actions. The challenge to anticipate the far-future actions from the observation of the small early portion of the video is obvious, since the network does not get enough information from the small early portion of the observed video and the future becomes more uncertain with the increasing anticipation time. As shown in Table VII, we perform the anticipation experiments on different percentages of observations. We observe the first 10%, 20%, 30%, or 50% of the video to anticipate the following 50% of that video. As the observation percentage increases, the network observes different actions and provides better anticipation. We also perform the experiment on different percentages of anticipation, as shown in Table VIII. We observe the first 50% of the video to anticipate the following 10%, 20%, 30%, and 50% of that video. Since the future becomes more uncertain with the increasing anticipation time, anticipating the far-future actions becomes more challenging. In future, we will focus on relational knowledge distillation to progressively distill relation between the far-future and the observed actions for better anticipation of the far-future.

V. CONCLUSION

In this paper, we developed a new approach, Jointly-learnt Action Anticipation Network (J-AAN) via Self-Knowledge Distillation (Self-KD) and cycle consistency, for human action anticipation. The extensive experiments show the effectiveness of our three major contributions: (1) In contrast to anticipate the future actions either directly or recursively, the joint learning of these two approaches leads to a better future action anticipation; (2) The self-knowledge distillation mechanism to train the J-AAN can handle the uncertainty on future action anticipation; and (3) The forward and backward mechanism based on cycle consistency constraint to train the J-AAN further improves the future action anticipation performance. Our proposed approach outperforms other state-of-the-art methods on the Breakfast [34], 50Salads [35], and EPIC-Kiteches-55 [36] datasets.

ACKNOWLEDGMENT

This project was supported by National Science Foundation via Cyber-Physical Systems project CMMI-1646162, National Robotics Initiative project CMMI-1954548, and Human-Technology Frontier project ECCS-2025929.

REFERENCES

[1] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in NeurIPS, 2014, pp. 568–576.

[2] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in CVPR, 2014, pp. 1725–1732.

[3] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in CVPR, 2017, pp. 6299–6308.

[4] Z. Yang, Y. Li, J. Yang, and J. Luo, "Action recognition with spatio-temporal visual attention on skeleton image sequences," IEEE-TCSVT, vol. 29, no. 8, pp. 2405–2415, 2018.

[5] X. Jiang, K. Xu, and T. Sun, "Action recognition scheme based on skeleton representation with ds-lstm network," IEEE-TCSVT, vol. 30, no. 7, pp. 2129–2140, 2019.

[6] Z. Shao, Y. Li, Y. Guo, X. Zhou, and S. Chen, "A hierarchical model for human action recognition from body-parts," IEEE-TCSVT, vol. 29, no. 10, pp. 2986–3000, 2018.

[7] Y. Ji, Y. Yang, F. Shen, H. T. Shen, and W.-S. Zheng, "Arbitrary-view human action recognition: A varying-view rgb-d action dataset," IEEE-TCSVT, vol. 31, no. 1, pp. 289–300, 2020.

[8] M. Moniruzzaman, Z. Yin, Z. H. He, R. Qin, and M. Leu, "Human action recognition by discriminative feature pooling and video segmentation attention model," IEEE-TMM, 2021.

[9] R. Yan, L. Xie, J. Tang, X. Shu, and Q. Tian, "Social adaptive module for weakly-supervised group activity recognition," in European Conference on Computer Vision, 2020, pp. 208–224.

[10] J. Tang, X. Shu, R. Yan, and L. Zhang, "Coherence constrained graph lstm for group activity recognition," IEEE-PAMI, 2019.

[11] R. Yan, J. Tang, X. Shu, Z. Li, and Q. Tian, "Participation-contributed temporal dynamic model for group activity recognition," in ACM-MM, 2018, pp. 1292–1300.

[12] R. Yan, L. Xie, J. Tang, X. Shu, and Q. Tian, "Higcin: Hierarchical graph-based cross inference network for group activity recognition," IEEE-PAMI, 2020.

[13] S. Paul, S. Roy, and A. K. Roy-Chowdhury, "W-talc: Weakly-supervised temporal activity localization and classification," in ECCV, 2018, pp. 563–579.

[14] P. Nguyen, T. Liu, G. Prasad, and B. Han, "Weakly supervised action localization by sparse temporal pooling network," in CVPR, 2018, pp. 6752–6761.

[15] Z. Shou, H. Gao, L. Zhang, K. Miyazawa, and S.-F. Chang, "Autoloc: Weakly-supervised temporal action localization in untrimmed videos," in ECCV, 2018, pp. 154–171.

[16] P. Lee, Y. Uh, and H. Byun, "Background suppression network for weakly-supervised temporal action localization," in AAAI, vol. 34, no. 07, 2020, pp. 11 320–11 327.

[17] M. Moniruzzaman, Z. Yin, Z. He, R. Qin, and M. C. Leu, "Action completeness modeling with background aware networks for weakly-supervised temporal action localization," in ACM-MM, 2020, pp. 2166–2174.

[18] B. Wang, X. Zhang, and Y. Zhao, "Exploring sub-action granularity for weakly supervised temporal action localization," IEEE-TCSVT, vol. 32, no. 4, pp. 2186–2198, 2021.

[19] A. Stoian, M. Ferecatu, J. Benois-Pineau, and M. Crucianu, "Fast action localization in large-scale video archives," IEEE-TCSVT, vol. 26, no. 10, pp. 1917–1930, 2015.

[20] X. Peng and C. Schmid, "Multi-region two-stream r-cnn for action detection," in ECCV, 2016, pp. 744–759.

[21] V. Kalogeiton, P. Weinzaepfel, V. Ferrari, and C. Schmid, "Action tubelet detector for spatio-temporal action localization," in ICCV, 2017, pp. 4405–4413.

[22] J. Huang, N. Li, T. Li, S. Liu, and G. Li, "Spatial–temporal context-aware online action detection and prediction," IEEE-TCSVT, vol. 30, no. 8, pp. 2650–2662, 2019.

[23] S. Bhardwaj, M. Srinivasan, and M. M. Khapra, "Efficient video classification using fewer frames," in CVPR, 2019, pp. 354–363.

[24] M. S. Ryoo, "Human activity prediction: Early recognition of ongoing activities from streaming videos," in ICCV, 2011, pp. 1036–1043.

[25] M. Sadegh Aliakbarian, F. Sadat Saleh, M. Salzmann, B. Fernando, L. Petersson, and L. Andersson, "Encouraging lstms to anticipate actions very early," in ICCV, 2017, pp. 280–289.

[26] X. Wang, J.-F. Hu, J.-H. Lai, J. Zhang, and W.-S. Zheng, "Progressive teacher-student learning for early action prediction," in CVPR, 2019, pp. 3556–3565.

[27] L. Chen, J. Lu, Z. Song, and J. Zhou, "Recurrent semantic preserving generation for action prediction," IEEE-TCSVT, vol. 31, no. 1, pp. 231–245, 2020.

[28] J. Weng, X. Jiang, W.-L. Zheng, and J. Yuan, "Early action recognition with category exclusion using policy-based reinforcement learning," IEEE-TCSVT, vol. 30, no. 12, pp. 4626–4638, 2020.

[29] J. Gao, Z. Yang, and R. Nevatia, "Red: Reinforced encoder-decoder networks for action anticipation," arXiv preprint arXiv:1707.04818, 2017.

[30] T. Lan, T.-C. Chen, and S. Savarese, "A hierarchical representation for future action prediction," in ECCV, 2014, pp. 689–704.

[31] C. Vondrick, H. Pirsiavash, and A. Torralba, "Anticipating visual representations from unlabeled video," in CVPR, 2016, pp. 98–106.

[32] Y. Wu, L. Zhu, X. Wang, Y. Yang, and F. Wu, "Learning to anticipate egocentric actions by imagination," IEEE-TIP, pp. 1143–1152, 2020.

[33] A. Furnari and G. M. Farinella, "What would you expect? anticipating egocentric actions with rolling-unrolling lstms and modality attention," in ICCV, 2019.

[34] H. Kuehne, A. Arslan, and T. Serre, "The language of actions: Recovering the syntax and semantics of goal-directed human activities," in CVPR, 2014, pp. 780–787.

[35] S. Stein and S. J. McKenna, "Combining embedded accelerometers with computer vision for recognizing food preparation activities," in ACM-UbiComp, 2013, pp. 729–738.

[36] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price et al., "The epic-kitchens dataset: Collection, challenges and baselines," IEEE-PAMI, vol. 43, no. 11, pp. 4125–4141, 2020.

[37] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," in CVPR, 2015, pp. 2625–2634.

[38] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in ICCV, 2019, pp. 6202–6211.

[39] R. Girdhar and D. Ramanan, "Attentional pooling for action recognition," in NeurIPS, 2017, pp. 34–45.

[40] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," IEEE-PAMI, vol. 35, no. 1, pp. 221–231, 2012.

[41] Z. Qiu, T. Yao, C.-W. Ngo, X. Tian, and T. Mei, "Learning spatio-temporal representation with local and global diffusion," in ICCV, 2019, pp. 12 056–12 065.

[42] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "An end-to-end spatio-temporal attention model for human action recognition from skeleton data," arXiv preprint arXiv:1611.06067, 2016.

[43] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in ICCV, 2015, pp. 4489–4497.

[44] L. Wang, Y. Xiong, D. Lin, and L. Van Gool, "Untrimmednets for weakly supervised action recognition and detection," in CVPR, 2017, pp. 4325–4334.

[45] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in ECCV, 2016, pp. 20–36.

[46] J.-F. Hu, W.-S. Zheng, L. Ma, G. Wang, and J. Lai, "Real-time rgb-d activity prediction by soft regression," in ECCV, 2016, pp. 280–296.

[47] Q. Ke, M. Bennamoun, S. An, F. Boussaid, and F. Sohel, "Human interaction prediction using deep temporal features," in ECCV, 2016, pp. 403–414.

[48] Y. Kong, D. Kit, and Y. Fu, "A discriminative model with multiple temporal scales for action prediction," in ECCV, 2014, pp. 596–611.

[49] Y. Kong, Z. Tao, and Y. Fu, "Deep sequential context networks for action prediction," in CVPR, 2017, pp. 1473–1481.

[50] J. Liu, A. Shahroudy, G. Wang, L.-Y. Duan, and A. C. Kot, "Ssnet: scale selection network for online 3d action prediction," in CVPR, 2018, pp. 8349–8358.

[51] L. Chen, J. Lu, Z. Song, and J. Zhou, "Ambiguousness-aware state evolution for action prediction," IEEE-TCSVT, 2022.

[52] Z. Qi, S. Wang, C. Su, L. Su, Q. Huang, and Q. Tian, "Self-regulated learning for egocentric video activity anticipation," IEEE-PAMI, 2021.

[53] T. Liu and K.-M. Lam, "A hybrid egocentric activity anticipation framework via memory-augmented recurrent and one-shot representation forecasting," in CVPR, 2022, pp. 13 904–13 913.

[54] D. Damen, H. Doughty, G. Maria Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price et al., "Scaling egocentric vision: The epic-kitchens dataset," in ECCV, 2018, pp. 720–736.

[55] T. Mahmud, M. Hasan, and A. K. Roy-Chowdhury, "Joint prediction of activity labels and starting times in untrimmed videos," in ICCV, 2017, pp. 5773–5782.

[56] Y. Shi, B. Fernando, and R. Hartley, "Action anticipation with rbf kernelized feature mapping rnn," in ECCV, 2018, pp. 301–317.

[57] Y. Abu Farha, A. Richard, and J. Gall, "When will you do what?-anticipating temporal occurrences of activities," in CVPR, 2018, pp. 5343–5352.

[58] Q. Ke, M. Fritz, and B. Schiele, "Time-conditioned action anticipation in one shot," in CVPR, 2019, pp. 9925–9934.

[59] F. Sener, D. Singhania, and A. Yao, "Temporal aggregate representations for long-range video understanding," in ECCV, 2020, pp. 154–171.

[60] D. Gong, J. Lee, M. Kim, S. J. Ha, and M. Cho, "Future transformer for long-term action anticipation," in CVPR, 2022, pp. 3052–3061.

[61] Y. Li, N. Wang, J. Liu, and X. Hou, "Demystifying neural style transfer," arXiv preprint arXiv:1701.01036, 2017.

[62] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," arXiv preprint arXiv:1412.6550, 2014.

[63] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in CVPR, 2017, pp. 4133–4141.

[64] S. Hahn and H. Choi, "Self-knowledge distillation in natural language processing," arXiv preprint arXiv:1908.01851, 2019.

[65] S. Yun, J. Park, K. Lee, and J. Shin, "Regularizing class-wise predictions via self-knowledge distillation," in CVPR, 2020, pp. 13 876–13 885.

[66] K. Kim, B. Ji, D. Yoon, and S. Hwang, "Self-knowledge distillation: A simple way for better generalization," arXiv preprint arXiv:2006.12000, 2020.

[67] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223–2232.

[68] R. W. Brislin, "Back-translation for cross-cultural research," IACCP, vol. 1, no. 3, pp. 185–216, 1970.

[69] D. He, Y. Xia, T. Qin, L. Wang, N. Yu, T.-Y. Liu, and W.-Y. Ma, "Dual learning for machine translation," NeurIPS, vol. 29, pp. 820–828, 2016.

[70] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: Automatic detection of tracking failures," in ICPR. IEEE, 2010, pp. 2756–2759.

[71] H. Sun, Z. Zhao, Z. Yin, and Z. He, "Reciprocal twin networks for pedestrian motion learning and future path prediction," IEEE-TCSVT, vol. 32, no. 3, pp. 1483–1497, 2021.

[72] C. Godard, O. Mac Aodha, and G. J. Brostow, "Unsupervised monocular depth estimation with left-right consistency," in CVPR, 2017, pp. 270–279.

[73] T. Zhou, P. Krahenbuhl, M. Aubry, Q. Huang, and A. A. Efros, "Learning dense correspondence via 3d-guided cycle consistency," in CVPR, 2016, pp. 117–126.

[74] Y. A. Farha, Q. Ke, B. Schiele, and J. Gall, "Long-term anticipation of activities with cycle consistency," arXiv preprint arXiv:2009.01142, 2020.

[75] T. Zhang, W. Min, Y. Zhu, Y. Rui, and S. Jiang, "An egocentric action anticipation framework via fusing intuition and analysis," in ACMMM, 2020, pp. 402–410.

[76] G. Kapidis, R. Poppe, E. van Dam, L. Noldus, and R. Veltkamp, "Multitask learning to improve egocentric action recognition," in ICCVW, 2019.

[77] G. Camporese, P. Coscia, A. Furnari, G. M. Farinella, and L. Ballan, "Knowledge distillation for action anticipation via label smoothing," in ICPR, 2021, pp. 3312–3319.

[78] S. Ma, L. Sigal, and S. Sclaroff, "Learning activity progression in lstms for activity detection and early detection," in CVPR, 2016, pp. 1942–1950.

[79] D. Damen, H. Doughty, G. M. Farinella, S. Fidler, A. Furnari, E. Kazakos, D. Moltisanti, J. Munro, T. Perrett, W. Price et al., "Scaling egocentric vision: The epic-kitchens dataset," in ECCV, 2018, pp. 720–736.

[80] A. Furnari, S. Battiato, and G. Maria Farinella, "Leveraging uncertainty to rethink loss functions and evaluation measures for egocentric action anticipation," in ECCVW, 2018.

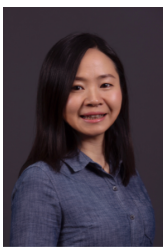
[81] Y. A. Farha and J. Gall, "Ms-tcn: Multi-stage temporal convolutional network for action segmentation," in CVPR, 2019, pp. 3575–3584.

Md Moniruzzaman received the B.S. degree from the Department of Electronics and Communication Engineering, Khulna University of Engineering and Technology, Bangladesh. He is currently working toward his Ph.D. degree at the Department of Computer Science, Stony Brook University, NY, USA. His current research interests include human action anticipation, human action recognition, temporal action localization, and human pose estimation.

Zhaozheng Yin is a SUNY Empire Innovation Associate Professor at Stony Brook University. He is affiliated with the AI Institute, Department of Biomedical Informatics, and Department of Computer Science. His group has been working on Biomedical Image Analysis, Computer Vision, and Machine Learning. Zhaozheng is an IEEE senior member and he has served as Area Chair for CVPR, ECCV, MICCAI and WACV, and an Associate Editor for IEEE-TCSVT and JVCI.

Zhihai He is a Robert Lee Tatum Distinguished Professor at University of Missouri. He is affiliated with the Department of Electrical and Computer Engineering. His current research interests include multimedia networking, wireless sensor networks, computer vision, and machine learning. Zhihai is an IEEE fellow and he has served as an Associate Editor for IEEE-TCSVT, IEEE-TMM, and JVCI. He was also the Guest Editor for the IEEE-TCSVT Special Issue on Video Surveillance.

Ming C Leu is the Keith and Pat Bailey Distinguished Professor in the Department of Mechanical and Aerospace Engineering, Missouri University of Science and Technology. He founded Missouri S&T's Center for Aerospace Manufacturing Technologies. His research interests are in the area of design and manufacturing automation including additive manufacturing, rapid prototyping, virtual prototyping, CAD/CAM, robotics, and machine dynamics and control. Dr. Leu has contributed to the engineering community through professional societies such as ASME, SME, and CIRP. Currently he is a member of the editorial board for the Journal of Virtual and Physical Prototyping and the Journal of Manufacturing Science and Technology. He co-organized and co-chaired the NSF CAREER Proposal Writing Workshop, NSF Workshop on Frontiers of Additive Manufacturing Research and Education, and NSF-ONR Roadmap for Additive Manufacturing Workshop.



Ruwen Qin is an Associate Professor at Stony Brook University. She is affiliated with the Department of Civil Engineering. Her research falls in the areas of data analytics, machine learning, and systems engineering. Using data captured by various sensors and other methods, she creates analytics tools and artificial intelligence (AI) models for analyzing, understanding, characterizing, and modeling people, systems, and processes. She seamlessly integrates the developed analytics tools and AI components into existing engineered systems to effectively turn them into cyber-physical systems, intelligent automation systems, and smart connected systems.