This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Time-Conditioned Action Anticipation in One Shot

Qiuhong Ke[1], Mario Fritz[2], Bernt Schiele[1]

[1]Max Planck Institute for Informatics, Saarland Informatics Campus
[2]CISPA Helmholtz Center for Information Security, Saarland Informatics Campus
Saarbrücken, Germany
{qke, schiele}@mpi-inf.mpg.de fritz@cispa.saarland

Abstract

The goal of human action anticipation is to predict future actions. Ideally, in real-world applications such as video surveillance and self-driving systems, future actions should not only be predicted with high accuracy but also at arbitrary and variable time-horizons ranging from short- to long-term predictions. Current work mostly focuses on predicting the next action and thus long-term prediction is achieved by recursive prediction of each next action, which is both inefficient and accumulates errors. In this paper, we propose a novel time-conditioned method for efficient and effective long-term action anticipation. There are two key ingredients to our approach. First, by explicitly conditioning our anticipation network on time allows to efficiently anticipate also long-term actions. And second, we propose an attended temporal feature and a time-conditioned skip connection to extract relevant and useful information from observations for effective anticipation. We conduct extensive experiments on the large-scale Epic-Kitchen and the 50Salads Datasets. Experimental results show that the proposed method is capable of anticipating future actions at both short-term and long-term, and achieves state-of-the-art performance.

1. Introduction

Human action anticipation, which aims to predict future unseen actions, is very important for many real-world applications. For example, in surveillance scenarios, early alert can be produced if abnormal events are anticipated, and in human-robot interaction scenario, the robots can provide timely corresponding interactions if they are able to anticipate human actions [39, 18].

Most current works investigate anticipation of the next action or actions after only one second [28, 29, 39, 4]. In real-world applications such as video surveillance, the system is often expected to be able to anticipate long-term actions (*e.g.*, the action after t seconds of the observation). Long-term anticipations can be achieved by anticipating the following actions one by one in an iterative way, *e.g.*, using RNN models [6]. This indicates that the anticipation at each time step is achieved based on the anticipation results of the previous time steps. This iterative method could be effective for certain scripted activities that contain fixed-order actions. However, in many real-world activities, the actions can be stochastic and not well structured. In this situation, the anticipation may be inaccurate at some steps, and these anticipation errors will accumulate during the iterative anticipation process. This often leads to performance degradation of anticipation, especially when anticipating long-term actions. Besides, if we only want to anticipate an action at the long-term, the iterative method is often time-consuming by producing the intermediate anticipations.

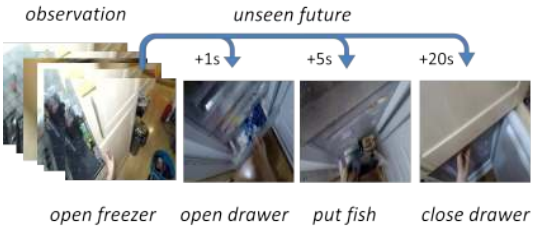

Figure 1. The proposed time-conditioned method for action anticipation. By incorporating the time parameter, the proposed method is capable of anticipating long-term actions efficiently and effectively.

In this paper, we introduce a novel method to achieve accurate and efficient action anticipation. Specifically, our method performs action anticipation by incorporating the time parameter to the information of the observation (see Figure 1). Therefore, it directly anticipates the action at fu-

ture time t in a *one-shot* fashion, and thus avoids anticipating all intermediate actions in the time period before t. Ideally, our method is t times faster than the iterative method for "sparse anticipation" (anticipating future actions at t). When performing "dense anticipation", our method will be less efficient. The advantage of our method in this case is that it is capable of generating more accurate future actions compared to the iterative method, as our method only relies on the observation for anticipation, bypassing accumulated anticipation errors.

The contributions of this paper are summarized as follows: 1) We introduce a new time-conditioned method for action anticipation; 2) We propose an attended temporal feature and a time-conditioned skip connection to extract useful information from the observation; 3) We conduct extensive experiments and analysis, and achieve state-of-the-art performance.

2. Related Work

Early action recognition. Many efforts have been developed for action recognition from RGB and depth videos [36, 8, 5, 40, 2, 25, 7, 31, 30, 14, 32, 38]. Early action recognition attracts an increasing attention in recent years [33, 10, 16, 20, 13, 11, 22, 23, 17, 1, 15, 3, 12, 24]. It is often referred to as action prediction. The goal of early action recognition is to recognize the label of an action from a partial observation of this action. Kong et al. [17] introduced a deep sequential context networks to reconstruct missing information of the partial observation for early action recognition. Liu et al. [23] proposed a new problem of online action recognition from untrimmed 3D skeleton streams and introduced a novel Scale Selection Network, which is capable of effectively and efficiently selecting the correct starting points of observed videos from untrimmed videos, and achieved state-of-the-art performance for early action recognition.

Early action detection. Early action detection aims to detect an action as early as possible before the action ends from untrimmed videos [10]. Ma et al. [27] introduced a new ranking loss to train a model based on LSTM for early action detection. The ranking loss encourages the model to generate non-decreasing detection scores when the model observes more activities. Shou et al. [35] formulated the detection of action start as a classification task of sliding windows and introduced a model based on Generative Adversarial Network to generate hard negative samples to improve the training of the model.

Action anticipation. Several works have investigated anticipation of the immediate future after the observation [28, 29, 39]. Vondrick et al. [39] introduced a regression network to learn the representation of future frames, followed by a classifier to anticipate the actions in one second. Gao et al. [9] introduced a Reinforced Encoder-Decoder Network to anticipate future representations using sequences of visual representations. Mahmud et al. [28] introduced a hybrid Siamese network to anticipate the next action label and the starting time. Qi et al. [29] introduced a spatial-temporal And-Or graph (AOG) to represent events and used a temporal grammar and early parsing algorithm to anticipate the next action. Damen et al. [4] leveraged TSN [40] to anticipate the next action after one second of the observation. The observation is used as input of the TSN and the label of the next action segment is set as the output of the TSN to train the network.

Recently, Farha et al. [6] introduced two methods for long-term action anticipation. One is based on a RNN model, which outputs the remaining length of the current action, the next action and its length. The prediction is conducted in an iterative way, *i.e.*, combine the prediction with the observation to predict the next action. The limitation of this method is that it is time-consuming and suffers from error accumulation. Another method is based on a CNN model, which outputs a sequence of future actions in a form of a matrix. The limitation of this method is that it introduces many parameters when predicting long sequences of future actions. Besides, it needs to pre-define the scale of the matrix.

Differently, in this paper, we propose a new method that is capable of anticipating a future action at both short-term and long-term in a *one-shot* fashion, which is efficient and effective.

3. Time-conditioned Action Anticipation in One Shot

Most of the existing works on action anticipation focus on anticipating the next action at a short-term time-horizon. The anticipation of a long-term action can be achieved in an iterative way by repeatedly predicting each next action. The limitation of this method is that it is often time-consuming for the anticipation of long-term actions. In addition, combining the anticipation at each time step for further anticipation accumulates the anticipation error and makes long-term anticipation inaccurate. In this section, we introduce our proposed time-conditioned method to mitigate the limitations of previous methods and effectively anticipate the future actions in one shot.

The overall architecture of the proposed method is shown in Figure 2. It mainly consists of two parts, *i.e.*, initial anticipation using an attended temporal feature, and final anticipation by including a time-conditioned skip connection. Below we describe each part in detail.

3.1. Attended Temporal Feature for Initial Anticipation

In order to directly anticipate the action after t seconds of the observation in one shot, we introduce a time parameter

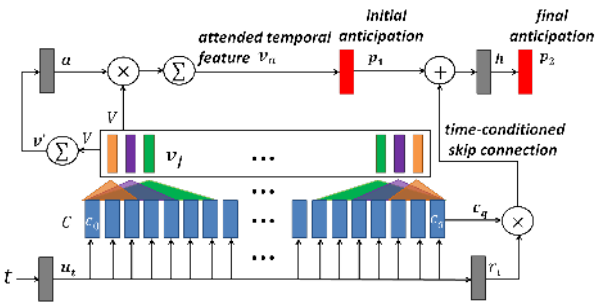

Figure 2. Overall architecture of the proposed method. It consists of two parts, *i.e.*, the attended temporal feature for initial anticipation, and the time-conditioned skip connection for final anticipation. t denotes the anticipation of future actions after t seconds of the observation. The action classes of the observation $C = [\boldsymbol{c_0}, \cdots, \boldsymbol{c_q}]$ and the time representation $\boldsymbol{u_t}$ are first concatenated to form the time-conditioned observation, which is used to extract multi-scale temporal feature V. $\boldsymbol{v'}$ is the sum of V across the temporal dimension. $\boldsymbol{v'}$ is used to generate an attention score $\boldsymbol{a}$, which is used to multiple V to achieve attended temporal feature $\boldsymbol{v_a}$. $\boldsymbol{v_a}$ is used to generate initial anticipation of the future action $\boldsymbol{p_1}$. r_t is the skip-connection weight, which is generated from time representation $\boldsymbol{u_t}$. r_t is used to multiply the last action of the observation $\boldsymbol{c_q}$, which is added to the initial anticipation $\boldsymbol{p_1}$ for final anticipation $\boldsymbol{p_2}$.

t for this task. The time parameter t is fed to a Multi-layer Perception with sigmoid activation layers to produce a time representation u_t. As shown in Figure 2, the action classes of the observed sequence $C = [\boldsymbol{c_0}, \cdots, \boldsymbol{c_q}] \in \mathbb{R}^{d_c \times q}$ and the time representation $u_t \in \mathbb{R}^{d_t}$ are concatenated for further processing. d_c and q denote the number of action classes and the time steps of the observation. d_t denotes the dimension of the time representation. This concatenated representation is referred to as time-conditioned observation. One can also add u_t to the observation to generate the time-conditioned observation. In this case, d_t needs to be set to equal to d_c.

The next step is to learn temporal information from the time-conditioned observation for action anticipation. Considering that the observation generally contains multiple actions, we hypothesize that the observation contains irrelevant information and the temporal information should be modeled from some particular parts of the observation in order to effectively anticipate the future action. To this end, we introduce an attended temporal feature as the representation of the time-conditioned observation for anticipation. Specifically, we design multi-scale temporal convolutions to process the time-conditioned observation, followed by an attention mechanism for selectively feature fusion. Attention has achieved great success in many fields such as caption generation [41], action recognition [34] and re-identification [26]. The temporal convolution of the i^{th} scale is formulated as follows:

$$V^i = f(W_c^i * C' + \boldsymbol{b_c^i}) \tag{1}$$

where $f(\cdot)$ denotes the activation function (here we use the ReLU function). C' denotes the time-conditioned observation. $W_c^i \in \mathbb{R}^{m \times k^i \times d_u}$ and $\boldsymbol{b_c^i} \in \mathbb{R}^m$ are the weight and bias of the temporal convolution of the i^{th} scale. k^i is the kernel size of the temporal convolution of the i^{th} scale. m is the number of convolutional filters of all scale, which is set to the same value for feature fusion. $d_u = d_c + d_t$ denotes the dimension of C' at each time step. $*$ represents the convolution operator. $V^i \in \mathbb{R}^{m \times n^i}$. $n^i = q - k^i + 1$ is the number of time steps of the temporal feature generated from the i^{th} scale temporal convolution. The output temporal features of all scales of temporal convolution are concatenated in the temporal dimension, which results in a multi-scale temporal feature $V \in \mathbb{R}^{m \times n}$. $n = \sum_{i=1}^{scale} n^i$ is the number of time steps of the multi-scale temporal features. As shown in Figure 2, $\boldsymbol{v_j}$ denotes the temporal feature at the j^{th} time step, *i.e.*, $\boldsymbol{v_j}$ corresponds to j^{th} column of V. To generate the attended temporal feature, we first use the sum of $\boldsymbol{v_j}$ to generate an attention score for all time steps of V as follows:

$$\begin{aligned} \boldsymbol{v'} &= \textstyle\sum_{j=1}^{n} \boldsymbol{v_j} \\ \boldsymbol{a} &= \text{softmax}(W_a \boldsymbol{v'} + \boldsymbol{b_a}) \end{aligned} \tag{2}$$

where $W_a \in \mathbb{R}^{n \times m}$ and $\boldsymbol{b_a} \in \mathbb{R}^n$ are the weights and bias of the attention layer. The attended temporal feature is calculated as follows:

$$\boldsymbol{v_a} = \sum_{j=1}^{n} a_j \boldsymbol{v_j} \tag{3}$$

The attended temporal feature is used to conduct initial anticipation of the future action as follows:

$$\boldsymbol{p_1} = \text{softmax}(W_{o1} \boldsymbol{v_a} + \boldsymbol{b_{o1}}) \tag{4}$$

where $W_{o1} \in \mathbb{R}^{d_c \times m}$ and $\boldsymbol{b_{o1}} \in \mathbb{R}^{d_c}$ are the weights and bias. d_c denotes the number of action classes as mentioned above. $\boldsymbol{p_1} \in \mathbb{R}^{d_c}$ is the probability of the future action. The i^{th} element of the prediction $\boldsymbol{p_1^{(i)}} \in [0, 1]$ corresponds to the i^{th} class. We refer to p_1 as initial anticipation.

3.2. Time-conditioned Skip Connection for Final Anticipation

The initial anticipation is generated using the temporal information along the sequence of observation. Human activities generally evolve continuously. The actions within *short temporal distance* are usually relevant to each other. Particularly, the last action of the observation is generally relevant to the future actions. In this section, we introduce a time-conditioned skip connection between the last observed

action and the initial anticipation in order to incorporate this complementary 'short-temporal-distance' information and generate an improved final anticipation. Intuitively, the last observed action is more relevant to short-term future actions than long-term actions. We therefore apply different weights ranging from zero to one to the last observed action before connecting to the initial anticipation. The weights are learned based on t, as shown in Figure 2. We refer to the weight as skip-connection weight. Specifically, given the time representation $\boldsymbol{u_t}$, the skip-connection weight r_t is calculated as follows:

$$r_t = \text{sigmoid}(W_s \boldsymbol{u_t} + b_s) \tag{5}$$

where $W_s \in \mathbb{R}^{1\times d_t}$ and b_s are the weights and bias. We denote the last action of the observation as $\boldsymbol{c_q} \in \mathbb{R}^{d_c}$. The time-conditioned skip connection is formulated as:

$$\boldsymbol{p_s} = r_t \boldsymbol{c_q} + \boldsymbol{p_1} \tag{6}$$

The time-conditioned skip connection is used to generate final anticipation as follows:

$$\begin{aligned} \boldsymbol{h} &= f(W_h \boldsymbol{p_s} + \boldsymbol{b_h}) \\ \boldsymbol{p_2} &= \text{softmax}(W_{o2} \boldsymbol{h} + \boldsymbol{b_{o2}}) \end{aligned} \tag{7}$$

where $W_h \in \mathbb{R}^{d_h \times d_c}$ and $\boldsymbol{b_h} \in \mathbb{R}^{d_h}$ are the weights and bias of the hidden layer before the output layer. $W_{o2} \in \mathbb{R}^{d_c \times d_h}$ and $\boldsymbol{b_{o2}} \in \mathbb{R}^{d_c}$ are the weights and bias of the output layer. $\boldsymbol{p_2} \in \mathbb{R}^{d_c}$ is final anticipation of the future action.

3.3. Objective

In Figure 2, the action classes of the observation C are generated from observed sequences by extracting a local spatial-temporal feature at each time step, and feeding the feature to a hidden layer and an output layer for action recognition. During training, we jointly train the recognition and anticipation network using the sum of all losses, which is formulated as:

$$\ell = \ell_r + \ell_{p_1} + \ell_{p_2} \tag{8}$$

where ℓ_r, ℓ_{p_1} and ℓ_{p_2} are the loss of the recognition, initial anticipation and final anticipation, respectively. Each loss is formulated as follows:

$$\begin{aligned} \ell_r &= -\sum_{j=1}^{q}\sum_{i=1}^{d_c} \boldsymbol{y_j^{(i)}} \log\left(\boldsymbol{c_j^{(i)}}\right) \\ \ell_{p_1} &= -\sum_{i=1}^{d_c} \boldsymbol{y_{t+q}^{(i)}} \log\left(\boldsymbol{p_1^{(i)}}\right) \\ \ell_{p_2} &= -\sum_{i=1}^{d_c} \boldsymbol{y_{t+q}^{(i)}} \log\left(\boldsymbol{p_2^{(i)}}\right) \end{aligned} \tag{9}$$

where q is the number of time steps of the observation as mentioned above. $\boldsymbol{y_j}$ is the ground-truth label of the action at the j^{th} time step of the observation. $\boldsymbol{y_j^{(i)}} = 1$ if the action class is i, and $\boldsymbol{y_j^{(i)}} = 0$ otherwise. $\boldsymbol{y_{t+q}}$ is the ground-truth label of the future action.

4. Experiments

The proposed method was evaluated on two datasets, *i.e.*, Epic-Kitchen Dataset [4] and 50Salads Dataset [37]. In this section, we report experimental results and detailed analysis.

4.1. Datasets

Epic-Kitchen Dataset. This dataset is a large first-person video dataset, which is captured by 32 subjects in 32 different kitchens. The videos in this dataset contain daily activities of the subjects, *i.e.*, no scripts are provided to instruct the subjects. This makes this dataset very natural and challenging. There are 272 training videos, which are captured by 28 subjects. Each video contains multiple action segments, which are categorized into 125 classes. Since the annotations of the testing videos are not available, we use the training videos to perform cross-validation for evaluation. Specifically, we randomly split the training videos into 7 splits, each containing videos of 4 subjects. We set the length of the observation to 30s and generate video clips via a temporal sliding window of 30s. The temporal stride of the sliding window is set to 1s. The frames without annotations are removed. This results in about 89600 sequences in total. The average number of testing videos across all splits is about 12800.

50Salads Dataset. This dataset contains 50 videos which are performed by 25 subjects. Each subject is preparing two mixed salads. There are 17 fine-grained action classes. We perform 5-fold cross-validation for evaluation using the splits provided by [21]. As in the Epic-Kitchen Dataset, we set the length of the observation to 30s and generate video segments using a temporal sliding window with a stride of 1s. This results in about 15100 sequences. The average number of the testing sequences across all splits is 3020.

4.2. Implementation Details

The scale of the temporal convolution is set to 4, with kernel sizes of 1,3,7 and 15. The filter sizes of all scales of the temporal convolution are set to 512. The numbers of units of all the hidden fully connected layers are set to 512. The learning rate is set to 0.01 and the batch size is set to 64. For feature representations of videos, we leverage I3D network [2] to extract spatial-temporal features from the videos of the Epic-kitchen Dataset. Specifically, we down-sample the videos to 20 frames/second and feed local video volumes at every second to the network. Each local volume contains 16 frames. For the feature representation of the 50Salads Dataset, we use the features provided by [21] for simplicity. The same feature is used in all methods for fair comparisons.

Methods	Top-1	Top-5	AvgCP	AvgCR
TSN [4, 40]	23.8%	70.2%	2.9%	5.0%
Proposed	**25.6%**	**71.6%**	**6.3%**	**7.3%**

Table 1. Next action anticipation for the Epic-Kitchen Dataset. 'Top-1', 'Top-5', 'AvgCP', 'AvgCR' represent the top-1 accuracy, top-5 accuracy, average class precision and average class recall, respectively.

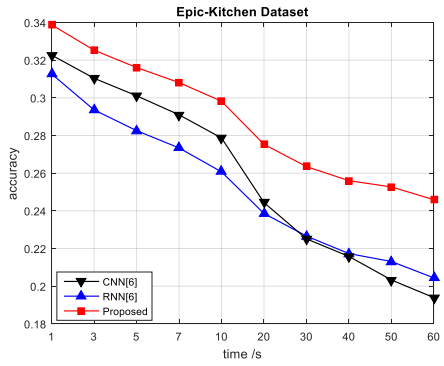

Figure 3. Long-term action anticipation for the Epic-Kitchen Dataset. The time t in the X axis represents the anticipation of the future actions after t seconds of the observation.

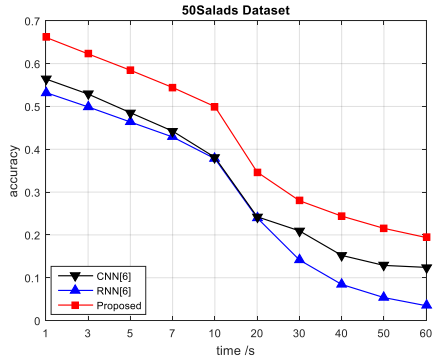

Figure 4. Long-term action anticipation on the 50Salads Dataset. The time t in the X axis represents the anticipation of the future actions after t seconds of the observation.

4.3. Comparison to the State-of-the-art

Epic-Kitchen Dataset. The initial protocol for action anticipation in this dataset is to predict the next action label after 1s of the observation [4]. We first follow this protocol to evaluate the proposed time-conditioned method without skip connection and compare to the Temporal Segment Networks (TSN) [40] method used in the paper [4] for action anticipation. The comparison aims to show that our basic framework can also work for this protocol, although our goal is long-term action anticipation. As the proposed method uses only the RGB frames, we also use RGB frames to train the TSN model. To evaluate anticipation performance using this protocol, we follow [4] and use action boundaries to generate training and testing data. We evaluate the top-1 accuracy, top-5 accuracy, average class precision and average class recall as [4] and show the results in Table 1. The proposed method outperforms TSN in all cases.

The proposed method is compared to the CNN method [6] and the RNN method [6] for long-term action anticipation. We report sparse anticipation results within 60 seconds in Figure 3. The performance of the proposed method is significantly better than the other two methods. The standard deviation among the 7 splits of the proposed method is around 0.03 for all time-steps. From Figure 3 it can also be seen that the improvements of the proposed method are more significant when anticipating longer-term actions, *e.g.*, actions after 60s of the observation. The RNN method anticipates future actions in an iterative way and is incapable to anticipate long-term actions accurately as the anticipation errors accumulate. Although the CNN method anticipates actions directly from the observation, the network tends to minimize the anticipation loss of short-term actions and is unable to anticipate long-term actions accurately. Compared to the RNN and CNN method, the proposed time-conditioned method anticipates future actions in one shot, and achieves the best performance for both short-term and long-term anticipations.

50Salads Dataset. The anticipation performance of the 50Salads Dataset is shown in Figure 4. The proposed method significantly outperforms the CNN method [6] and the RNN method [6] in all anticipation cases. Specifically, when anticipating future actions after 10s of the observation, the performance of the proposed method is 50.0%, which is 11.9% and 12.2% better than the CNN method (38.1%) and RNN (37.8%) method, respectively. The average anticipation accuracy of the proposed method is 32.5%. Compared to the average accuracy of the CNN method (23.8%) and the RNN method (18.5%), the improvements of the proposed method are 8.7% and 14%. We also follow the protocol in [6] to generate the training and testing data for dense anticipation. In this protocol, the input is set to the labels of a particular percentage (*e.g.*, 20%) of each video, and the goal is to anticipate a following sub-sequence with a percentage (*e.g.*, 10%) of the video. We follow [6] to anticipate future action segments and the duration of each action for fair comparison with their iterative method. The time t in this case represents the t^{th} action segment in unseen videos. The duration is generated in a form of vector that includes the duration ratios of all unseen segments using an additional softmax layer. The results are shown in Table 2 and Table 3. We also follow [6] to evaluate dense anticipation on the Breakfast Dataset [19]. The results are shown in Table 4 and Table 5. We also achieve better performance than the RNN method and the CNN method overall.

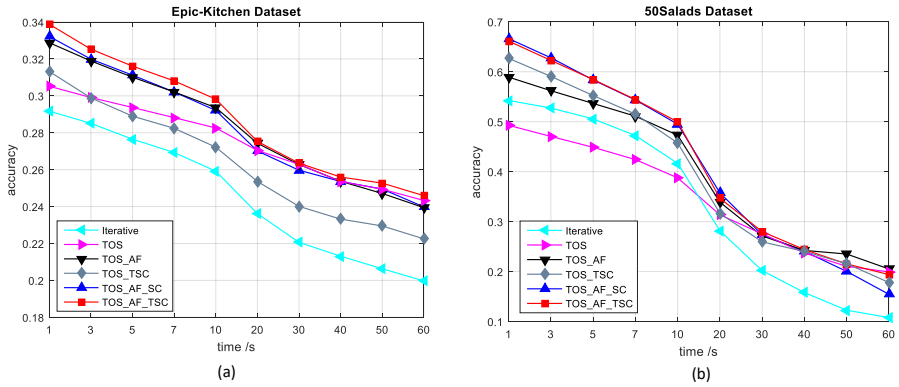

Figure 5. Comparison between different baselines and the proposed TOS_AF_TSC on the Epic-Kitchen Dataset and the 50Salads Dataset.

4.4. Benefit of Time-conditioned One-shot Anticipation

In this paper, we incorporate the time parameter to anticipate actions of any future time in one shot. We conduct the following baselines to demonstrate the benefit of this method for long-term action anticipation. 1) Time-conditioned One-shot Anticipation (**TOS**). In this baseline, we simply average the temporal features and incorporate the time parameter for future action anticipation. This baseline does not contain the attended temporal feature or the time-conditioned skip connection in order to show the benefit of the time-conditioned method for long-term action anticipation. 2) Iterative Anticipation (**Iterative**). In this baseline, we do not incorporate the time parameter for future action anticipation. Instead, we use the same feature as the TOS baseline to anticipate future actions in next time step. This baseline is similar to the RNN method [6]. Particularly, the anticipation of first time step is combined with the observation to predict the next time step. This process is repeated t times in an iterative way to anticipate the actions in the t^{th} time step. As the anticipation is combined with the observation for anticipation of the next time step, the length of the anticipating time step is set to the same frame rate of the observation. In our case, the length of each time step is 1s. The results of these two baselines are shown in Figure 5. The one-shot baseline significantly outperforms the iterative baseline for long-term action anticipation on both datasets. The iterative baseline anticipates long-term future actions by repeatedly combining the prediction of next step with the observation. This process accumulates the predicting error of each step and results in worse performance for long-term action anticipation. From Figure 5(b) it can be seen that the iterative baseline outperforms the one-shot baseline on the 50Salad dataset for the anticipation of future actions within 10s. This could be due to that this dataset contains scripted actions, making it easy to anticipate actions in short term. In this case, there is less error for the anticipation of the short-term actions. The TOS method outperforms iterative method when anticipating actions after 20s of the observation. It clearly shows the advantage of conditioning on time for long-term action anticipation.

4.5. Benefit of Attended Temporal Feature

In this work, we use an attended temporal feature for action anticipation. In order to demonstrate the benefit of this method, we further conduct the following baseline: Time-conditioned One-shot Anticipation using Attended Temporal Feature (**TOS_AF**). This baseline is used to compare to the TOS baseline. In this baseline, we also incorporate time parameter to anticipate future actions. Instead of equally using the observation by averaging the temporal features, we use the attended temporal feature for action anticipation. The results on the Epic-Kitchen Dataset and the 50Salads Dataset are shown in Figure 5. The TOS_AF baseline improves the TOS baseline in both datasets, especially for shot-term action anticipation. When anticipating long-term actions, the improvement of the TOS_AF baseline compared to the TOS baseline is not that significant. For anticipating long-term actions, it is better to use all the observed actions to obtain a high-level concept of the future activities. In this case, the averaged temporal feature provides useful information, which makes the TOS baseline achieve a similar performance to the TOS_AF baseline.

4.6. Benefit of Time-conditioned Skip Connection

The proposed method (**TOS_AF_TSC**) contains a time-conditioned skip connection to provide useful 'short-temporal-distance' information of the last observed action for action anticipation. We compare the proposed method with the TOS_AF baseline. In order to demonstrate the benefit of conditioning on time for skip connection, we further conduct the following baseline: Time-conditioned One-shot Anticipation using Attended Temporal Feature and Skip Connection (**TOS_AF_SC**). This baseline is used

Observation	20%				30%			
Prediction	10%	20%	30%	50%	10%	20%	30%	50%
RNN [6]	0.3006	0.2543	0.1874	0.1349	0.3077	0.1719	0.1479	0.0977
CNN [6]	0.2124	0.1903	0.1598	0.0987	0.2914	0.2014	0.1746	0.1086
Proposed	**0.3251**	**0.2761**	**0.2126**	**0.1599**	**0.3512**	**0.2705**	**0.2205**	**0.1559**

Table 2. Dense anticipation mean over classes accuracy on the 50Salads Dataset (without ground-truth observation).

Observation	20%				30%			
Prediction	10%	20%	30%	50%	10%	20%	30%	50%
RNN [6]	0.4230	0.3119	0.2522	0.1682	0.4419	0.2951	0.1996	0.1038
CNN [6]	0.3608	0.2762	0.2143	0.1548	0.3736	0.2478	0.2078	**0.1405**
Proposed	**0.4512**	**0.3323**	**0.2759**	**0.1727**	**0.4640**	**0.3480**	**0.2524**	0.1384

Table 3. Dense anticipation mean over classes accuracy on the 50Salads Dataset (with ground-truth observation).

Observation	20%				30%			
Prediction	10%	20%	30%	50%	10%	20%	30%	50%
RNN [6]	0.1811	0.1720	0.1594	0.1581	0.2164	0.2002	**0.1973**	0.1921
CNN [6]	0.1790	0.1635	0.1537	0.1454	0.2244	0.2012	0.1969	0.1876
Proposed	**0.1841**	**0.1721**	**0.1642**	**0.1584**	**0.2275**	**0.2044**	0.1964	**0.1975**

Table 4. Dense anticipation mean over classes accuracy on the Breakfast Dataset (without ground-truth observation).

Observation	20%				30%			
Prediction	10%	20%	30%	50%	10%	20%	30%	50%
RNN [6]	0.6035	0.5044	0.4528	0.4042	0.6145	0.5025	0.4490	0.4175
CNN [6]	0.5797	0.4912	0.4403	0.3926	0.6032	0.5014	0.4518	0.4051
Proposed	**0.6446**	**0.5627**	**0.5015**	**0.4399**	**0.6595**	**0.5594**	**0.4914**	**0.4423**

Table 5. Dense anticipation mean over classes accuracy on the Breakfast Dataset (with ground-truth observation).

to compared to the TOS_AF baseline and the TOS_AF_TSC method. In this baseline, besides using the attended feature for future action anticipation, we also incorporate skip connection to anticipate future actions. Compared to the TOS_AF_TSC method, this baseline does not use the time parameter to generate the skip-connection weight. The results of the TOS_AF_SC and the TOS_AF_TSC methods are shown in Figure 5. The skip connection improves the performance of the TOS_AF baseline, especially for the short-term action anticipation. However, when predicting future actions after 60s of the observation, the accuracy of the TOS_AF_SC baseline is 15.5%, which is 5% worse than the TOS_AF baseline (20.5%). The TOS_AF_SC baseline directly adds the last observed action to the initial anticipation for the final anticipation of the actions in any future time. Intuitively, the information of the last observed action is more beneficial for anticipating short-term actions as actions generally change continuously and the neighbour actions are usually relevant. For long-term action anticipation, the information of the last observed action is less important. In this case, directly adding the last observed action makes the performance worse. The proposed TOS_AF_TSC, on the other hand, uses the time parameter to generate a weight for skip connection, which improves the performances of both short-term and long-term action anticipation.

	Future time				
Number	1s	5s	10s	30s	50s
1	33.9%	31.6%	29.8%	26.4%	25.3%
2	33.5%	31.1%	29.1%	25.5%	24.1%
3	33.4%	31.1%	29.2%	25.7%	24.2%
4	33.5%	31.1%	29.4%	25.7%	24.5%
5	33.8%	31.4%	29.6%	25.8%	24.7%

Table 6. Anticipation accuracy on the Epic-kitchen Dataset. 'Number' represents the numbers of observations used for skip connection in the proposed method.

4.7. Comparison of Attended Temporal Feature and Time-conditioned Skip Connection

The attended temporal feature aims to select relative temporal information from the whole observation for initial anticipation, while the time-conditioned skip connection incorporates the last observed action to the initial anticipation for final anticipation. We have shown that the time-conditioned skip connection improves the initial anticipation of the attended temporal feature. In order to show that the attended temporal feature is indispensable for anticipation, we also conduct the following baseline: using only the last observed action for anticipation, *i.e*., the time representation is concatenated with the last observed action to generate initial anticipation. The time-conditioned skip connection is also incorporated for final anticipation. Compared to the proposed TOS_AF_TSC, there is no attended temporal feature. We refer to this baseline as **TOS_TSC**. The results are shown in Figure 5. It can be seen that the performance of TOS_TSC is worse than the proposed method. When anticipating future actions after 60s of the observation, the performance of the TOS_TSC baseline is 22.3%, which is 2.3% worse than the proposed method with attended temporal feature. From Figure 5 it can also be seen that the TOS_AF baseline outperforms the TOS_TSC baseline, especially for long-term action anticipation. The TOS_TSC baseline uses only the last observed action to anticipate future actions. It does not contain the temporal information of the observation. While the last action could be useful for short-term action anticipation, the temporal information is more useful to anticipate long-term actions.

4.8. Analysis on the Number of Observations for Skip Connection

In the proposed method, we use only the last observation in the skip connection for anticipation. We further conduct the experiments of including more observations to the last one in the skip connection. The results on the Epic-kitchen Dataset are shown in Table 6. There is no much difference among the performance of the methods that use different numbers of observations. One possible reason is that the last few observations might belong to the same action class and including more observations does not add more information.

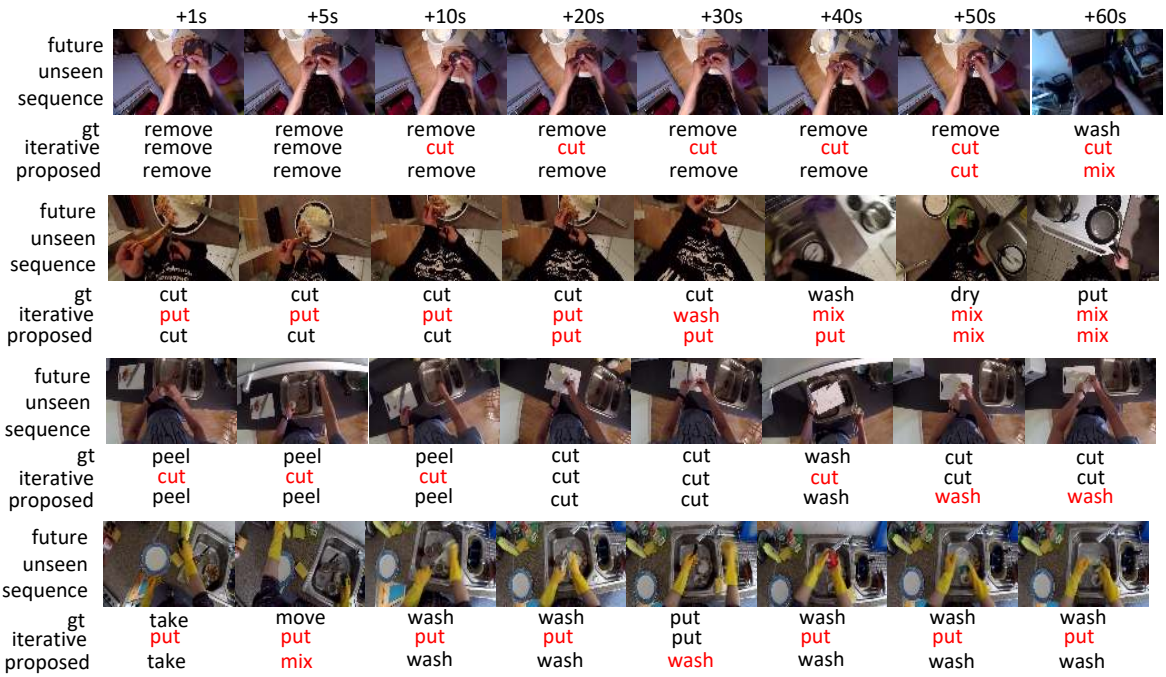

Figure 6. Visualization of future action anticipation on the Epic-Kitchen Dataset. We show the results of the proposed method and the iterative method for anticipating future actions after t seconds of the observation. Each column corresponds to one t value, which is indicated in the first row. Incorrect anticipations are shown in red.

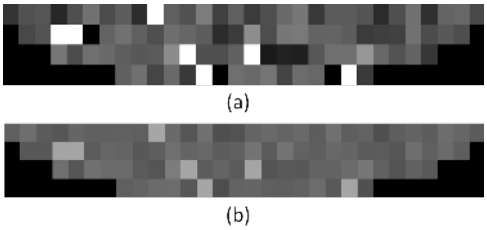

Figure 7. Average attention maps of anticipating actions at (a) future 1s and (b) future 30s on the Epic-kitchen Dataset. Each row corresponds to the feature of one scale of temporal convolution. The bottom row corresponds to the feature of the largest scale of temporal convolution, which contains less time steps than the features of other smaller scales due to the larger kernel size of temporal convolution.

4.9. Visualization of Future Action Anticipation

Figure 6 shows some examples of future action anticipation on the Epic-Kitchen Dataset. It can be seen that the proposed method generates more diverse anticipations of future actions, while the iterative method tends to generate the same anticipation for different future time steps. Besides, when the anticipations of short-term actions are incorrect, the proposed method can still generate correct anticipations of long-term actions, as shown in the last example in Figure 6. This is because the proposed method does not rely on the anticipations of previous time steps to anticipate actions.

4.10. Visualization of Attention Map

Figure 7 shows average attention maps (a in Figure 2) of anticipating actions at future 1s and future 30s on the Epic-kitchen dataset. The brighter color denotes attention of a larger weight. The attention of anticipating 1s is more selective, and focuses on different time steps of the multi-scale temporal features, while the attention of anticipating 30s is less selective and includes all time steps of the features.

5. Conclusions

In this paper, we have introduced a novel method for action anticipation. The proposed method explicitly conditions the anticipation on time, which is more efficient and effective for long-term action anticipation. Moreover, we have introduced an attended temporal feature to extract useful temporal information of the observation. We have also introduced a time-conditioned skip connection to incorporate the information of the last observed action to enhance the anticipation. We have conduct extensive experiments and have shown the advantages of the proposed method for anticipating future actions at both short-term and long-term.

Acknowledgment

This research was supported in part by the German Research Foundation (DFG CRC 1223).

References

[1] M. S. Aliakbarian, F. Saleh, M. Salzmann, B. Fernando, L. Petersson, and L. Andersson. Encouraging lstms to anticipate actions very early. In *IEEE International Conference on Computer Vision*, 2017.

[2] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4724–4733, 2017.

[3] L. Chen, J. Lu, Z. Song, and J. Zhou. Part-activated deep reinforcement learning for action prediction. In *European Conference on Computer Vision*, pages 421–436, 2018.

[4] 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 *European Conference on Computer Vision*, 2018.

[5] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015.

[6] Y. A. Farha, A. Richard, and J. Gall. When will you do what?-anticipating temporal occurrences of activities. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[7] C. Feichtenhofer, A. Pinz, and A. Zisserman. Convolutional two-stream network fusion for video action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1933–1941, 2016.

[8] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5378–5387, 2015.

[9] J. Gao, Z. Yang, and R. Nevatia. Red: Reinforced encoder-decoder networks for action anticipation. In *British Machine Vision Conference*, 2017.

[10] M. Hoai and F. De la Torre. Max-margin early event detectors. *International Journal of Computer Vision*, 107(2):191–202, 2014.

[11] J.-F. Hu, W.-S. Zheng, L. Ma, G. Wang, and J. Lai. Real-time RGB-D activity prediction by soft regression. In *European Conference on Computer Vision*, pages 280–296, 2016.

[12] J.-F. Hu, W.-S. Zheng, L. Ma, G. Wang, J.-H. Lai, and J. Zhang. Early action prediction by soft regression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[13] Q. Ke, M. Bennamoun, S. An, F. Boussaid, and F. Sohel. Human interaction prediction using deep temporal features. In *European Conference on Computer Vision*, pages 403–414, 2016.

[14] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid. Learning clip representations for skeleton-based 3d action recognition. *IEEE Transactions on Image Processing*, 27(6):2842–2855, 2018.

[15] Q. Ke, M. Bennamoun, S. An, F. Sohel, and F. Boussaid. Leveraging structural context models and ranking score fusion for human interaction prediction. *IEEE Transactions on Multimedia*, 20(7):1712–1723, 2018.

[16] Y. Kong, D. Kit, and Y. Fu. A discriminative model with multiple temporal scales for action prediction. In *European Conference on Computer Vision*, pages 596–611. 2014.

[17] Y. Kong, Z. Tao, and Y. Fu. Deep sequential context networks for action prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1473–1481, 2017.

[18] H. S. Koppula and A. Saxena. Anticipating human activities using object affordances for reactive robotic response. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(1):14–29, 2016.

[19] H. Kuehne, A. Arslan, and T. Serre. The language of actions: Recovering the syntax and semantics of goal-directed human activities. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 780–787, 2014.

[20] T. Lan, T.-C. Chen, and S. Savarese. A hierarchical representation for future action prediction. In *European Conference on Computer Vision*, pages 689–704, 2014.

[21] C. Lea, M. D. Flynn, R. Vidal, A. Reiter, and G. D. Hager. Temporal convolutional networks for action segmentation and detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 156–165, 2017.

[22] W. Li and M. Fritz. Recognition of ongoing complex activities by sequence prediction over a hierarchical label space. In *IEEE Winter Conference on Applications of Computer Vision*, pages 1–9, 2016.

[23] J. Liu, A. Shahroudy, G. Wang, L.-Y. Duan, and A. C. Kot. Ssnet: Scale selection network for online 3d action prediction. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 8349–8358, 2018.

[24] J. Liu, A. Shahroudy, G. Wang, L.-Y. Duan, and A. C. Kot. Skeleton-based online action prediction using scale selection network. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[25] J. Liu, A. Shahroudy, D. Xu, A. C. Kot, and G. Wang. Skeleton-based action recognition using spatio-temporal lstm network with trust gates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):3007–3021, 2018.

[26] Y. Lou, Y. Bai, J. Liu, and L.-Y. Duan. Veri-wild: A large dataset and a new method for vehicle re-identification in the wild. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[27] S. Ma, L. Sigal, and S. Sclaroff. Learning activity progression in lstms for activity detection and early detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1942–1950, 2016.

[28] T. Mahmud, M. Hasan, and A. K. Roy-Chowdhury. Joint prediction of activity labels and starting times in untrimmed videos. In *IEEE International Conference on Computer Vision*, pages 5784–5793, 2017.

[29] S. Qi, S. Huang, P. Wei, and S.-C. Zhu. Predicting human activities using stochastic grammar. In *IEEE International Conference on Computer Vision*, 2017.

[30] H. Rahmani and M. Bennamoun. Learning action recognition model from depth and skeleton videos. In *IEEE International Conference on Computer Vision*, pages 5832–5841, 2017.

[31] H. Rahmani, A. Mahmood, D. Q. Huynh, and A. Mian. Hopc: Histogram of oriented principal components of 3d pointclouds for action recognition. In *European Conference on Computer Vision*, pages 742–757, 2014.

[32] H. Rahmani, A. Mian, and M. Shah. Learning a deep model for human action recognition from novel viewpoints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(3):667–681, 2018.

[33] M. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *IEEE International Conference on Computer Vision*, pages 1036–1043, 2011.

[34] S. Sharma, R. Kiros, and R. Salakhutdinov. Action recognition using visual attention. In *International Conference on Learning Representations Workshop*, 2016.

[35] Z. Shou, J. Pan, J. Chan, K. Miyazawa, H. Mansour, A. Vetro, X. Giro-i Nieto, and S.-F. Chang. Online detection of action start in untrimmed, streaming videos. In *European Conference on Computer Vision*, pages 534–551, 2018.

[36] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *Advances in Neural Information Processing Systems*, pages 568–576, 2014.

[37] S. Stein and S. J. McKenna. Combining embedded accelerometers with computer vision for recognizing food preparation activities. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 729–738, 2013.

[38] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6):1510–1517, 2018.

[39] C. Vondrick, H. Pirsiavash, and A. Torralba. Anticipating visual representations from unlabeled video. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 98–106, 2016.

[40] 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 *European Conference on Computer Vision*, pages 20–36, 2016.

[41] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*, pages 2048–2057, 2015.